EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION

EAMT2018

Proceedings of the 21st Annual Conference of the European Association for Machine Translation

28–30 May 2018
Universitat d'Alacant
Alacant, Spain

Edited by

Juan Antonio Pérez-Ortiz
Felipe Sánchez-Martínez
Miquel Esplà-Gomis
Maja Popović
Celia Rico
André Martins
Joachim Van den Bogaert
Mikel L. Forcada

Organised by

Universitat d'Alacant
Universidad de Alicante

transducens
research group

ISBN: 978-84-09-01901-4

Contents

Foreword from the General Chair

As president of the European Association for Machine Translation (EAMT), it is a great pleasure for me to write the foreword to the Proceedings of the 21th annual conference of the EAMT.

The EAMT started organizing annual workshops in 1996; later, these workshops became annual conferences, and were hosted all around Europe. Years ago, the venue was steadily moving from west to east: from Barcelona (2009) to Saint-Raphaël (2010) to Leuven (2011) to Trento (2012) to Dubrovnik (2014) —after skipping one year to host the successful world-wide MT Summit 2013 in Nice— , but recently turned around to go west again at Antalya (2015), to go to Riga (2016), then Prague (2017) and now Alacant (2018). There will be no EAMT 2019, as it is the Association's turn to organize the Machine Translation Summit, which will take place in Dublin, but EAMT 2020 will inevitably take place west from Alacant: it will be soon announced.

By the way, if you have not done so yet, and live in Europe, North Africa, or the Middle East, please consider joining the EAMT. Our membership rates are low, particularly for students and people not based in Europe. You will benefit from discounts when attending not only our conferences, but also the conferences held by our partner associations the Asia-Pacific Association for Machine Translation (AAMT) and the Association for Machine Translation in the Americas (AMTA). You will also have an exclusive chance to benefit from funding for your activities related to machine translation. And perhaps you can get even more involved and participate in serving the European machine translation community by becoming a member of the Executive Committee of the EAMT.

But let me go back to EAMT 2018. As in previous conferences, I am so happy to see the strong programme put together by our programme chairs: Maja Popović, research track chair, André Martins and Joachim van Bogaert, user track co-chairs, and Celia Rico, who will chair the new translators' track, aiming at bringing machine translator researchers, developers, and vendors closer to the actual individuals using them. To accommodate this new track, EAMT 2018 will for the first time be a full three-day conference.

As in previous editions, there will also be a projects and products session showcasing the advance of machine translation in Europe. And, last but not least, I also feel very fortunate to have Sharon O'Brien from Dublin City University as our invited speaker.

EAMT 2018 would have never been possible without the generous offer to host and the hard work subsequently done by the local organizing committee at the Transducens research group of the Universitat d'Alacant, headed by Juan Antonio Pérez-Ortiz. I warmly thank my local colleagues (especially Juan Antonio, Felipe Sánchez-Martínez, and Miquel Esplà-Gomis) for putting EAMT 2018 together!

It is also with great pleasure that I thank our sponsors: Pangeanic (gold sponsor), Star Group and text&form (silver sponsors), Vicomtech (bronze sponsor), and Prompsit, Apertium, Linguaserve, and Unbabel (supporting sponsors), and ample support from the Universitat d'Alacant. Finally, I would like to thank EAMT 2018 attendees for coming to Alacant. I hope the conference leads to new friendships and fruitful collaboration.

Mikel L. Forcada
EAMT President

Message from the Organising Committee Chair

I want to take this opportunity to give you a big thank you for joining us in the 21st Annual Conference of the European Association for Machine Translation, EAMT 2018. On behalf of the organising committee, it is my pleasure to welcome you to Alacant. This year the Transducens research group and the Universitat d'Alacant proudly assume hosting the conference from the 28th to the 30th of May 2018. We decided to set the conference venue in downtown, but the main campus of our university is only six kilometres away. Regarded as one of the most beautiful European campuses, I encourage you to visit us there someday.

The city has held many different names: the Carthaginians called it Akra Leuka (white mountain), then the Romans changed its name to Lucentum, and the Moors—who ruled the region for a few centuries and started the building of the Santa Bàrbara castle on top of Mount Benacantil—called it Medina Laqant or al-Laqant. The Moorish name later resulted in the Catalan toponym Alacant and the Spanish Alicante. If you stand by the Postiguet beach and look up to the Mount Benacantil you will notice a rock formation that clearly resembles a man's face, a face that we have chosen as the main motif in the EAMT 2018 logo, where it is accompanied by some blue waves from the Mediterranean Sea. Legends tell us that the face is that of a Moorish king who was doomed to eternal damnation when her heartbroken daughter threw herself off the castle on to the rocks of Mount Benacantil after the king had disapproved her marriage with the one she truly loved. The king was condemned this way to watch all the lovers in the city and remember for all the eternity what he sadly forbade.

According to our predictions, this is going to be one of the most crowded editions ever of the EAMT conference. The unexpectedly high number of attendees have forced us to make some unavoidable last-minute changes that I hope will not negatively affect your enjoyment of the scientific and social activities of the conference.

We look forward to your active participation during the three days of the conference. Do not hesitate to ask questions when the session chairs invite you to do so. Please, contribute to make this edition of the conference a fruitful forum where a multidisciplinary group of researchers, developers, practitioners, leaders, vendors, users, and translators all share experiences and motivating ideas.

Finally, I would like to express my sincere appreciation to the persons and organisations that have made this conference possible: the European Association for Machine Translation, our gold sponsor (Pangeanic), silver sponsors (text&form and Star Group), bronze sponsor (Vicomtech), supporters (Apertium, Linguaserve, Prompsit, Unbabel), media sponsor (Multilingual), institutional partners (Universitat d'Alacant, Institut Universitari d'Investigació Informàtica), programme chairs (Maja Popović, Celia Rico, André Martins, Joachim Van den Bogaert), keynote speaker (Sharon O'Brien), and, finally but so importantly, my colleagues Miquel Esplà-Gomis, Mikel L. Forcada and Felipe Sánchez-Martínez who have worked extraordinarily hard to make your stay as pleasant and inspiring as possible.

Juan Antonio Pérez-Ortiz
Universitat d'Alacant

Preface by the Programme Chairs

It is our pleasure to welcome you to the 21st annual conference of the European Association for Machine Translation (EAMT) to be held in Alicante, Spain. We have really enjoyed serving as programme chairs for this edition of the conference. The EAMT conference has become the most importan event in Europe in the area of machine translation for researchers, users and professional translators. This year, there are four different tracks: research, user and project/product track, as in previous editions, and for the first time, translators' – individual translators are invited to share their insights in the use of MT.

The research track concerns novel and significant research results in any aspect of machine translation and related areas while the user track reports users' experiences with machine translation in industry, government, NGOs, as well as innovative uses of MT. The project/product track offers project and products the opportunity to be presented to the wide audience of the conference. Finally, the machine translation community needs to hear the translators' voice in a fresh and unfiltered way and learn from their insights through the translators' track.

This year we have received 46 submissions to the research track, 16 submissions to the user track, 22 descriptions of projects and products and 10 submissions to the translators' track. Each submission to the research, user and translator tracks was peer reviewed by three independent members of the Programme Committee. In the research track, 27 papers (58.7%) were accepted for publication, whereas 7 papers (44%) were accepted for the user track and 8 submissions (80%) for the translators track. Aside from regular papers from the four tracks, the programme includes an invited talk by Sharon O'Brien from Dublin City University on "Human-centred translation technology". We will also have a presentation by Barry Haddow on "The WMT Shared Tasks", and a presentation by the winner of the EAMT Best Thesis Award.

We would like to thank the Programme Committee members whose names are listed below for their high quality reviews and recommendation which have been very useful for the Programme Chairs to make decisions. We would also like to thank all the authors for trying their best to incorporate the reviewers' suggestions when preparing the final versions of their papers. For the papers which were not accepted, we hope that the reviewers' comments will be useful for improving them. Special thanks to Mikel Forcada for all his help and advices.

Maja Popović
Humboldt-Universität zu Berlin

Celia Rico
Universidad Europea

André Martins
Unbabel

Joachim Van den Bogaert
CrossLang

EAMT 2018 Committees

General Chair

Mikel L. Forcada, EAMT President, Universitat d'Alacant

Programme Chairs

Research track

Maja Popović, Humboldt-Universität zu Berlin

User track

André Martins, Unbabel
Joachim Van den Bogaert, CrossLang

Translators' track

Celia Rico, Universidad Europea

Organising committee

Juan Antonio Pérez-Ortiz, Universitat d'Alacant (chair)
Miquel Esplà-Gomis, Universitat d'Alacant
Felipe Sánchez-Martínez, Universitat d'Alacant

Programme Committee

Research track

Aleš Tamchyna, Memsource a.s.
Anabela Barreiro, INESC-ID
Andreas Eisele, European Commission, DGT
Andreas Guta, RWTH Aachen University
Andrei Popescu-Belis, IDIAP
Annette Rios Gonzales, University of Zurich
Antonio Toral, University of Groningen
Arianna Bisazza, University of Leiden
Bogdan Babych, University of Leeds
Carla Parra Escartín, Dublin City University
Carolina Scarton, The University of Sheffield

Christian Dugast, tech2biz
Christian Federmann, Microsoft
Christian Hardmeier , Uppsala University
Clare Voss, ARL
Constantin Orasan, University of Wolverhampton
Cristina España i Bonet, UdS and DFKI
Daniel Ortiz-Martínez, Universitat Politècnica de Valencia
David Vilar Torres, Amazon
Dimitar Shterionov, Dublin City University
Ekaterina Lapshinova-Koltunski, Saarland University
Eleftherios Avramidis, DFKI
Eva Vanmassenhove, Dublin City University
Evgeny Matusov, eBay
Federico Gaspari, Dublin City University
Felipe Sánchez-Martínez, Universitat d'Alacant
Francisco Casacuberta, Universitat Politècnica de València
Francisco Javier Guzman, Facebook
Francis M. Tyers, Higher School of Economics
Franck Burlot, LIMSI-CNRS
François Yvon, LIMSI/CNRS et Université Paris-Sud
George Foster, NRC
Helena Caseli, Federal University of São Carlos (UFSCar)
Houda Bouamor, Carnegie Mellon University
Iacer Calixto, Dublin City University
Irina Temnikova, University of Sofia
Jan Niehues, Karlsruhe Institute of Technology
Jerneja Žganec Gros, Alpineon R&D
Joachim Daiber, Apple
Joke Daems, Ghent University
Jörg Tiedemann, University of Helsinki
José G. C. de Souza, eBay Inc.
Joss Moorkens, Dublin City University
Juan Antonio Pérez-Ortiz, Universitat d'Alacant
Laura Jehl, Universität Heidelberg
Lieve Macken, Ghent University
Luisa Bentivogli, FBK-irst
Mārcis Pinnis, Tilde
Marco Turchi, Fondazione Bruno Kessler
Maria Nadejde, The University of Edinburgh
Marianna Apidianaki, LIMSI-CNRS
Marija Brkić, University of Rijeka
Marion Weller-Di Marco, University of Amsterdam
Mark Fishel, University of Tartu
Markus Freitag, IBM
Marta R. Costa-Jussà, Universitat Politècnica de Catalunya
Martin Volk, University of Zurich
Matteo Negri, Fondazione Bruno Kessler (FBK-irst)
Matthias Huck, Ludwig Maximilian University of Munich
Michel Simard, National Research Council Canada (NRC)

Mihaela Vela, Saarland University
Miloš Stanojević, The University of Edinburgh
Miquel Esplà-Gomis, Universitat d'Alacant
Mireia Farrús, Universitat Pompeu Fabra
Mirjam Sepesy Maučec, University of Maribor
Nicola Ueffing, eBay
Nizar Habash, New York University Abu Dhabi
Núria Bel, Universitat Pompeu Fabra
Parnia Bahar, RWTH Aachen University
Philipp Koehn, Johns Hopkins University
Philip Williams, The University of Edinburgh
Preslav Nakov, Qatar Computing Research Institute
Qun Liu, Dublin City University
Rudolf Rosa, Charles University Prague
Samuel Läubli, University of Zurich
Sanja Štajner, University of Mannheim
Sara Stymne, Uppsala University
Sharon O'Brien, Dublin City University
Sheila Castilho, Dublin City University
Špela Vintar, University of Ljubljana
Stephan Peitz, Apple
Tamer Alkhouli, RWTH Aachen University
Teresa Herrmann, Fujitsu
Víctor M. Sánchez-Cartagena, Universitat d'Alacant
Vincent Vandeghinste, Katholieke Universiteit Leuven
Violeta Seretan, University of Geneva
Yvette Graham, Dublin City University

User track

Aljoscha Burchardt, DFKI
Andrzej Zydroń, XTM Internation Ltd.
Andy Way, ADAPT Centre — Dublin City University
Arda Tezcan, Ghent University
Arle Lommel, CSA Research
Bianka Buschbeck, SAP
Bruno Pouliquen, World Intellectual Property Organization
Charlotte Tesselaar, LexisNexis
Christine Bruckner, Freelance
Fábio Kepler, Unbabel
Gema Ramírez-Sánchez, Prompsit Language Engineering, S.L.
Heidi Depraetere, CrossLang
Heidi Van Hiel, Yamagata Europe
Helena Moniz, Unbabel
Joao Almeida Graca, Unbabel
Jost Zetzsche, IWG
Julie Beliao, Unbabel
Kim Harris, text & form
Lena Marg, Welocalize
Matiss Rikters, University of Latvia

Matthias Heyn, SDL
Maxim Khalilov, Booking.com
Miriam Kaeshammer, SAP
Olga Beregovaya, Welocalize
Ramon Astudillo, Unbabel / INESc-ID
Samuel Läubli, University of Zürich
Sara Szoc, CrossLang
Tatjana Gornostaja, Tilde
Teresa Herrmann, Fujitsu, Luxembourg
Tony O'Dowd, Xcelerator Machine Translations Ltd.

Translators' track

Ana González, freelance translator
Ana Guerberof, Adapt Centre, Dublin
Enrique Torrejón, Deloitte, European Union Intellectual Property Office
Gabriel Cabrera, freelance translator
Ignacio Garcia, Western Sydney University
Javier Mallo, freelance translator
Javier Sánchez, Donnelley Language Solutions
Julia Aymerich, PanAmerican Health Organization
María Azqueta, Seprotec Multilingual Solutions
Livia Florensa, CPSL
Lucía Morado, Université de Genève
Luis González, DGT – European Commission
Manuel Mata, freelance translator
Olga Blasco, Consultant, Business strategy
Olga Torres-Hostench, Universitat Autònoma de Barcelona
Pilar Sánchez-Gijón, Universitat Autònoma de Barcelona
Rubén Rodríguez de la Fuente, PayPal
Uwe Muegge, Anthrex
Vanessa Enríquez Raído, The University of Auckland
Vicenta Ten Soriano, SDL Trados Studio
Willem Stoeller, Localization Institute

Sponsors

Gold sponsor

pangeanic

Silver sponsors

STAR

text & form

Bronze sponsors

vicomtech
visual interaction & communication technologies

Supporters

Apertium

linguaserve

prompsit

Unbabel

Media sponsor

MultiLingual
language | tech | business

multilingual.com

Institutional partners

Universitat d'Alacant
Universidad de Alicante

UNIVERSIDAD DE ALICANTE
UNIVERSITAT D'ALACANT
I.U. INVESTIGACIÓN INFORMÁTICA
I.U. D'INVESTIGACIÓ INFORMÀTICA

transducens
research group

Invited Speech

Human-centered translation technology

Sharon O'Brien, Dublin City University, Ireland

As AI drives advances in technology one recurring question is: what is the role of the human now and in the (near) future? This question is relevant for many disciplines, including medicine, law, accounting and, not least, translation. Translation is not a stranger to technology disruption and the modern translation pipeline is already highly technologised, at least in some sectors. However, there are benefits to focusing on users even in this high tech production pipeline. In my Keynote, I will suggest that we need to move from "computer-aided translation" and "human-in-the loop" to a human-centered translation technology (HCTT) paradigm. By focusing on three cohorts - professional translators, ad hoc translators, and end users - I will demonstrate how attention is shifting to HCTT and I will propose some research and development challenges for translation technology to embrace in order to position the human firmly in the centre of the design and use ecosystem.

EAMT 2018 Best Thesis Award — Anthony C Clarke Award

Twelve PhD theses defended in 2017 were received as candidates for the 2018 edition of the EAMT Best Thesis Award, Anthony C Clarke Award, and all twelve were eligible. A panel of 41 reviewers was recruited to examine and score the theses, considering how challenging the problem tackled in each thesis was, how relevant the results are for machine translation as a field, and what the strength of its impact in terms of scientific publications was. It became very clear that 2017 was a very good year for PhD theses in machine translation. The scores of the best theses were very close, which made it very hard to select a winner. A panel of three EAMT Executive Committee members (Barry Haddow, Juan Antonio Pérez-Ortiz, and Mikel L. Forcada) was assembled to process the reviews and select a winner.

The panel has decided to grant the 2018 edition of the EAMT Best Thesis Award, Anthony C Clarke Award, to **Daniel Emilio Beck** for his thesis “Gaussian Processes for Text Regression”, University of Sheffield, supervised by Lucia Specia and Trevor Cohn.

Gaussian Processes for Text Regression

Daniel Beck[1]
School of Computing and Information Systems
The University of Melbourne, Australia
d.beck@unimelb.edu.au

This thesis deals with the general problem of predicting numerical indicators from textual data. This task, which we call Text Regression, arises in a range of different applications in Natural Language Processing (NLP). For instance, in Quality Estimation (QE) (Blatz et al., 2004; Specia et al., 2009), sentences generated from Machine Translation (MT) systems are evaluated according to a task-based metric such as post-editing effort or time. In Emotion Analysis (EA) (Strapparava and Mihalcea, 2007), natural language sentences are assigned with numerical scores mapping the strength of a particular emotion (or a set of emotions).

Standard approaches for Text Regression rely on architectures similar to the ones used in classification tasks. These use engineered features and/or simple text representations such as bag-of-words (BOW), and make predictions in the form of single point estimates. These simplifying assumptions ignore important aspects of the data. Representations such as BOW ignore structural aspects of sentences and fails to capture structural linguistic phenomena such as word order. Point estimate predictions lack uncertainty information on the predicted variable, which can help subsequent decision making and is particularly important when annotations are noisy (such as post-editing time in QE).

The goal of this thesis is to advance the state-of-the-art in Text Regression by improving these two aspects: improved text representations and better uncertainty modelling in the response variables. In order to achieve that goal we propose to use Gaussian Processes (GPs) (Rasmussen and Williams, 2006) as the regression model. GPs are a Bayesian kernelised framework which is considered the state-of-the-art in regression (Hensman et al., 2013). Perhaps surprisingly, GPs were not widely investigated in the context of NLP applications.[2] Therefore a secondary goal of this thesis is to disseminate GPs in the NLP community, in particular for regression tasks.

The theory behind Gaussian Processes regression makes it ideal to solve the two problems mentioned above. Since it models response variables as well-calibrated distributions, it naturally provides a measure of uncertainty over the predictions. Furthermore, by employing kernels as the underlying learning component, we can incorporate complex text representations through what we named *structural kernels*. Combining with the efficient model selection procedures provided by GPs, we show in this thesis how to essentially learn representations by enabling richer kernel parameterisations. In this thesis, we focus on string kernels (Lodhi et al., 2002; Cancedda et al., 2003) and tree kernels (Collins and Duffy, 2001; Moschitti, 2006) but the theory can easily be extended to other kinds of structures such as graph kernels (Vishwanathan et al., 2010).

We benchmark our approach in two Text Regression applications. The first one is Emotion Analysis, where we use a GP model with a soft string kernel using word embeddings for similarity calculation between words. We show that this proposed model can obtain better results compared to simpler baselines. For this task, we also propose a multi-task model which leverages multiple emotional labels and show how we can inspect GP

[1] This thesis was written while the author was a Ph.D. student at The University of Sheffield, United Kingdom.

[2] Notable exceptions are Polajnar et al. (2011) and Cohn and Specia (2013).

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 5–6
Alacant, Spain, May 2018.

hyperparameters to cluster similar emotions.

The second benchmark is Machine Translation Quality Estimation. In this task, we show that can obtain better results compared to baselines while also providing uncertainty estimates for predictions. More important, we show how to employ the predictive distributions in an asymmetric risk scenario, where over and underestimates of post-editing time have different costs. This is an example application where propagating full uncertainty information can be beneficial for further decision making in a translation pipeline. As another application example, we also show how to use uncertainty estimates to annotate QE datasets via active learning.

Finally, as mentioned before, this thesis also has the goal of disseminating Gaussian Processes among the NLP community. By providing the theoretical grounds and showcasing its application in two benchmarks, we hope that it will serve as a starting point for other NLP problems in the future.

Access to the full thesis is open and available at the White Rose eTheses repository (`etheses.whiterose.ac.uk/17619`).

Acknowledgements

Daniel Beck was supported by funding from CNPq/Brazil (No. 237999/2012-9). The author would also like to thank his Ph.D. supervisors, Lucia Specia and Trevor Cohn, and his thesis examiners, Andreas Vlachos and Xavier Carreras.

References

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th Conference on Computational Linguistics*, pages 315–321.

Cancedda, Nicola, Eric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-Sequence Kernels. *The Journal of Machine Learning Research*, 3:1059–1082.

Cohn, Trevor and Lucia Specia. 2013. Modelling Annotator Bias with Multi-task Gaussian Processes: An Application to Machine Translation Quality Estimation. In *Proceedings of ACL*, pages 32–42.

Collins, Michael and Nigel Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of NIPS*, pages 625–632.

Hensman, James, Nicolò Fusi, and Neil D. Lawrence. 2013. Gaussian Processes for Big Data. In *Proceedings of UAI*, pages 282–290.

Lodhi, Huma, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text Classification using String Kernels. *The Journal of Machine Learning Research*, 2:419–444.

Moschitti, Alessandro. 2006. Making Tree Kernels practical for Natural Language Learning. In *EACL*, pages 113–120.

Polajnar, Tamara, Simon Rogers, and Mark Girolami. 2011. Protein interaction detection in sentences via Gaussian Processes: a preliminary evaluation. *International Journal of Data Mining and Bioinformatics*, 5(1):52–72, jan.

Rasmussen, Carl Edward and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*, volume 1. MIT Press Cambridge.

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of EAMT*, pages 28–35.

Strapparava, Carlo and Rada Mihalcea. 2007. SemEval-2007 Task 14 : Affective Text. In *Proceedings of SemEval*, pages 70–74.

Vishwanathan, S. V. N., Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. 2010. Graph Kernels. *Journal of Machine Learning Research*, 11:1201–1242.

Special Feature

The WMT Shared Tasks

Barry Haddow
School of Informatics
University of Edinburgh
Edinburgh
Scotland
bhaddow@staffmail.ed.ac.uk

Abstract

The annual WMT Conference in Machine Translation has been running shared tasks since 2006. It started with a translation task based on Europarl, and has grown to include tasks on all aspects of MT corpus preparation, training and evaluation, including the flagship task on news translation. I will review the history of the task, lessons learnt, and plans for future tasks.

1 Introduction

We began organising shared tasks in machine translation at the Workshop in Machine Translation (WMT) in 2006, initially with a translation task based on Europarl. In later years, funding from the EU projects EuroMatrixPlus and MosesCore (FP7) and QT21 and Cracker (H2020), plus commercial sponsorship, enabled us to increase the number of tasks and to produce professionally translated, unseen test sets drawn from news texts for the translation task. In 2016 WMT became a conference (retaining the acronym) and in the last three years the number of shared tasks has varied between 7 and 10.

The shared tasks have covered translation (mainly news, but also other domains such as IT and biomedical and also more specialized tasks such as pronoun and multimodal), training (both tuning of SMT and training of NMT), reference-based evaluation, quality estimation, corpus preparation (document alignment and corpus cleaning) as well as automatic post-editing. The quality estimation task has included different subtasks on estimating the quality of MT output at word, sentence and document level, as well as trying to predict the post-editing effort required for a given MT output. The data from all the WMT tasks, including the training data, test data and task submissions is made available for future research and has been heavily used in academic publications.

In the news translation task we have tried to innovate in MT evaluation, whilst still providing for comparison with previous years. After several years using a *relative ranking* approach, where evaluators compare output from different systems, we switched to *direct assessment* (DA) in 2017. In DA, evaluators provide an assessment of adequacy on a scale from 0 to 100, which we find offers a reliable system ranking and a more interpretable and comparable final score. The news task covers a variety of languages, mainly European, with English–German and English–Czech as our "core" languages. We have included both low-resource (e.g. Estonian–English and Hindi–English) and high-resource (e.g. French–English) pairs, and we release our own parallel and monolingual data sets, as well as using standard sets like Europarl.

In this talk I will review the history of the tasks, the lessons learnt and plans for future tasks, focusing on the news translation task. I will explain how this task provides a common benchmark for comparing different MT systems, which helps to drive MT research. I will also show how running the task reveals difficulties and pitfalls in comparative evaluation of MT systems.

2 Website

The URL for the latest conference/task is `www.statmt.org/wmt18`, where you will find links to all previous conferences/workshops, tasks and papers.

Research papers

Contextual Handling in Neural Machine Translation: Look Behind, Ahead and on Both Sides

Ruchit Agrawal(1,2), **Marco Turchi**(1), **Matteo Negri**(1)
(1)Fondazione Bruno Kessler, Italy
(2)University of Trento, Italy
{ragrawal, turchi, negri}@fbk.eu

Abstract

A salient feature of Neural Machine Translation (NMT) is the end-to-end nature of training employed, eschewing the need of separate components to model different linguistic phenomena. Rather, an NMT model learns to translate individual sentences from the labeled data itself. However, traditional NMT methods trained on large parallel corpora with a one-to-one sentence mapping make an implicit assumption of sentence independence. This makes it challenging for current NMT systems to model inter-sentential discourse phenomena. While recent research in this direction mainly leverages a single previous source sentence to model discourse, this paper proposes the incorporation of a context window spanning previous as well as next sentences as source-side context and previously generated output as target-side context, using an effective non-recurrent architecture based on self-attention. Experiments show improvement over non-contextual models as well as contextual methods using only previous context.

1 Introduction

Neural Machine Translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014; Cho et al., 2014) has consistently outperformed other MT paradigms across a range of domains, applications and training settings (Bentivogli et al., 2016; Castilho et al., 2017; Toral and Sánchez-Cartagena, 2017), thereby emerging as the *de facto* standard in Machine Translation. NMT models are typically trained at the sentence level (Cho et al., 2014), whereby the probability of an output sentence given an input sentence is maximized, implicitly making an assumption of sentence independence across the dataset. This works well for the translation of stand-alone sentences or datasets containing shuffled sentences, which are not connected with each other in terms of discursive dependencies. However, in real life situations, written text generally follows a sequential order featuring a number of cross-sentential phenomena. Additionally, speech-like texts (Bawden, 2017) exhibit the trait of contextual dependency and sequentiality as well, often containing a greater number of references that require a common knowledge ground and discourse understanding for correct interpretation. Figure 1 shows an example of such inter-sentential dependencies. These dependencies are not fully leveraged by the majority of contemporary NMT models, owing to the treatment of sentences as independent units for translation.

In order to perform well on sequential texts, NMT models need access to extra information, which could serve as the disambiguating context for better translation. Recent work in this direction (Zoph and Knight, 2016; Jean et al., 2017; Tiedemann and Scherrer, 2017; Bawden et al., 2017; Wang et al., 2017) has primarily focused on previous source-side context for disambiguation. Since all of these approaches utilize recurrent architectures, adding context comprising of more than a single previous sentence can be challenging due to either (i) the increased number of estimated parameters and training time, in case of the multi-encoder approach (Jean et al., 2017), or (ii)

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 11–20
Alacant, Spain, May 2018.

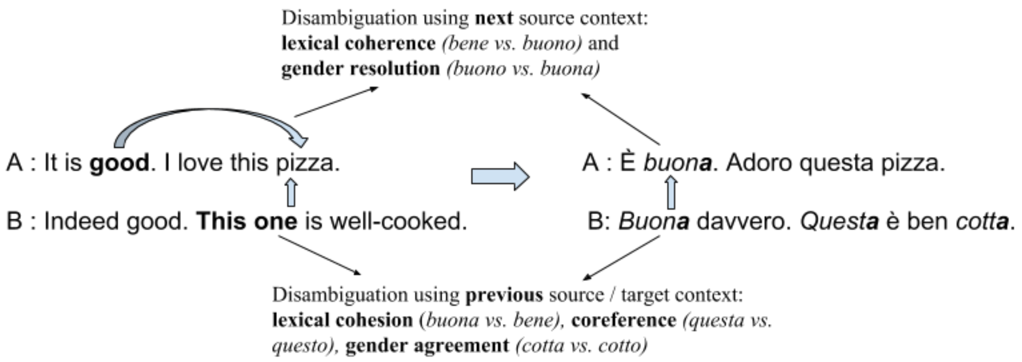

Disambiguation using **next** source context:
lexical coherence *(bene vs. buono)* and
gender resolution *(buono vs. buona)*
A : It is **good**. I love this pizza.
B : Indeed good. **This one** is well-cooked.
A : È *buona*. Adoro questa pizza.
B: *Buona* davvero. *Questa* è ben *cotta*.
Disambiguation using **previous** source / target context:
lexical cohesion (*buona vs. bene*), **coreference** *(questa vs. questo)*, **gender agreement** *(cotta vs. cotto)*

Figure 1: Inter-sentential dependencies requiring previous (source and target) and next (source) context

performance drop due to very long inputs (Koehn and Knowles, 2017), in case of extended translation units (Tiedemann and Scherrer, 2017). Hence, the impact of utilizing a large-sized context window on the source as well as the target side remains unclear. Additionally, the impact of incorporating the next sentences as context in the source side also needs to be examined, owing to discourse phenomena like cataphora and gender agreement, illustrated in Figure 1.

We address this gap and investigate the contribution of a context window looking behind as well as ahead on the source-side, combined with previous target-side context, in an efficient non-recurrent "Transformer" architecture with self-attention (*hereafter Transformer*), recently proposed by Vaswani et al. (2017). We choose this architecture due to its effective handling of long-range dependencies and easily achievable computational parallelization. These characteristics are due to the fact that the Transformer is based entirely on self-attention, as opposed to LSTMs or GRUs. The non-recurrent architecture enables effective parallelization, which is not possible with RNNs due to their sequentiality, thereby reducing the computational complexity considerably. We perform experiments using differently sized context windows on the source and target side. This is the first effort towards contextual NMT with Transformer to the best of our knowledge. On the English-Italian data from the IWSLT 2017 shared task (Cettolo et al., 2017), the best of our models achieves a 2.3% increase in BLEU score over a baseline Transformer model trained without any inter-sentential context and a 2.6% increase in BLEU score over a multi-source BiLSTM model trained using a previous source sentence as additional context.

The major contributions of this paper are summarized below:

- We demonstrate that looking ahead at the following text in addition to looking behind at the preceding text on the source-side improves performance.
- We demonstrate that both source-side context as well as target-side context help to improve translation quality, the latter however is more prone to error propagation.
- We demonstrate that looking further beyond a single previous sentence on the source-side results in better performance, especially in absence of target-side context.
- We show that a simple method like concatenation of the multiple inputs, when used with the Transformer, generates efficient translations, whilst being trained more than three times faster than an RNN based architecture.

The rest of the paper is organized as follows: We describe an outline of the related work in Section 2, and provide a theoretical background in Section 3. Section 4.1 briefly describes the discourse phenomena which we would like to capture using our contextual NMT models. Our approach to model discourse and the experiments conducted are described in Section 4. Section Section 5 presents the results obtained by our models, along with a linguistic analysis of the implications therein. We present the conclusions of the present research and highlight possible directions for future work in Section 6.

2 Related Work

Discourse modeling has been explored to a significant extent for Statistical Machine Translation (Hardmeier, 2012), using methods like discriminative learning (Giménez and Màrquez, 2007; Tamchyna et al., 2016), context features (Gimpel and Smith, 2008; Costa-Jussà et al., 2014; Sánchez-Martínez et al., 2008; Vintar et al., 2003), bilingual language models (Niehues et al., 2011), document-wide decoding (Hardmeier et al., 2012; Hardmeier et al., 2013) and factored models (Meyer et al., 2012). The majority of these works, however, look mainly at intra-sentential discourse phenomena, owing to the limited capability of SMT models to exploit extra-sentential context. The neural MT paradigm, on the other hand, offers a larger number of avenues for looking beyond the current sentence during translation.

Recent work on incorporating contextual information into NMT models has delved primarily into multi-encoder models (Zoph and Knight, 2016; Jean et al., 2017; Bawden et al., 2017), hierarchy of RNNs (Wang et al., 2017) and extended translation units containing the previous sentence (Tiedemann and Scherrer, 2017). These approaches build upon the multi-task learning method proposed by Luong et al. (2015), adapting it specifically for translation. Zoph and Knight (2016) propose a multi-source training method, which employs multiple encoders to represent inputs coming from different languages. Their method utilizes the sources available in two languages in order to produce better translations for a third language. Jean et al. (2017) use the multi-encoder framework, with one set of encoder and attention each for the previous and the current source sentence as an attempt to model context. However, this method would be computationally expensive with an increase in the number of contextual sentences owing to the increase in estimated parameters.

Wang et al. (2017) employ a hierarchy of RNNs to summarize source-side context (previous three sentences). This method addresses the computational complexity to an extent, however it does not incorporate target-side context, which has been shown to be useful by (Bawden et al., 2017). Bawden et al. (2017) present an in-depth analysis of the evaluation of discourse phenomena in NMT and the challenges faced thereof. They provide a hand-crafted test set specifically aimed at capturing discursive dependencies. However, this set is created with the assumption that the disambiguating context lies in the previous sentence, which is not always the case (Scarton et al., 2015).

Our work is most similar to (Tiedemann and Scherrer, 2017), who employ the standard NMT architecture without multiple encoders, but using larger blocks containing the previous and the current sentence as input for the encoder, as an attempt to better model discourse phenomena. The primary limitation of this method is the inability to add larger context due to the ineffective handling of long-range dependencies by RNNs (Koehn and Knowles, 2017). Additionally, this method does not look at the following source-text, due to which phenomena like cataphora and lexical cohesion are not captured well.

While the above-mentioned works employ the previous source text, we propose employing a context window spanning previous as well as next source sentences in order to model maximal discourse phenomena. On the target-side, we decode the previous and current sentence while looking at the source-window, thereby employing target-side context as well. Additionally, we employ the Transformer for our contextual models, as opposed to the above-mentioned works using RNNs, due to the enhanced long-range performance and computational parallelization.

3 Background

3.1 NMT with RNNs and Transformer

Neural MT employs a single neural network trained jointly to provide end-to-end translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014). NMT models typically consist of two components - an encoder and a decoder. The components are generally composed of Stacked RNNs (Recurrent Neural Networks), using either Long Short Term Memory (LSTM) (Sundermeyer et al., 2012) or Gated Recurrent Units (GRU) (Chung et al., 2015). The encoder transforms the source sentence into a vector from which the decoder extracts the probable targets. Specifically, NMT aims to model the conditional probability $p(y|x)$ of translating a source sentence x = $x_1, x_2...x_u$ to a target sentence y = $y_1, y_2, ...y_v$. Let s be the representation of the source sentence as computed by the encoder. Based on the source representation, the decoder produces a translation, one target word at a time and decomposes the conditional probability as:

$$\log p(y|x) = \sum_{j=1}^{v} \log p(y_j|y_{<j}, s) \qquad (1)$$

The entire model is jointly trained to maximize the (conditional) log-likelihood of the parallel training corpus:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(y^{(n)}|x^{(n)}, \theta) \qquad (2)$$

where $(y^{(n)}, x^{(n)})$ represents the n^{th} sentence in parallel corpus of size N and θ denotes the set of all tunable parameters.

Research in NMT recently witnessed a major breakthrough in the Transformer architecture proposed by Vaswani et al. (2017). This architecture eschews the recurrent as well as convolution layers, both of which are integral to the majority of contemporary neural network architectures. Instead, it uses stacked multi-head attention as well as positional encodings to model the complete sequential information encoded by the input sentences. The decoder comprises of a similar architecture, using masked multi-head attention followed by softmax normalization to generate the output probabilities over the target vocabulary. The positional encodings are added to the input as well as output embeddings, enabling the model to capture the sequentiality of the input sentence without having recurrence. The encodings are computed from the position (pos) and the dimension (i) as follows:

$$PE_{(pos,2i)} = sin(pos/10000^{(2i/d_{model})}) \qquad (3)$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{(2i/d_{model})}) \qquad (4)$$

where PE stands for positional encodings and d_{model} is the dimensionality of the vectors resulting from the embeddings learned from the input and output tokens. Thus, each dimension of the encoding (i) corresponds to a sinusoid.

3.2 Inter-sentential discourse phenomena

Coherence in a text is implicitly established using a variety of discourse relations. Contextual information can help in handling a variety of discourse phenomena, mainly involving lexical choice, linguistic agreement, coreference - anaphora (Hardmeier and Federico, 2010) as well as cataphora, and lexical coherence. Spoken language especially contains a large number of such dependencies, due to the presence of an environment facilitating direct communication between the parties (Pierrehumbert and Hirschberg, 1990), where gestures and a common ground/theme are often used the disambiguating context, thereby rendering the need for explicit mentions in the text less important. A reasonable amount of noun phrases are established deictically, and the theme persists until it's taken over by another theme.

The deictic references are challenging to resolve for NMT models using only the current sentence-pair in consideration, and possible errors involving gender usage as well as linguistic agreement can be introduced in the translation. For instance, for English → Italian translation, establishing the linguistic features of the noun under consideration is crucial for translation. The co-ordination with the adjective (*buona* vs *buono*), pronominal references (*lui* vs *lei*), past participle verb form (*sei andato* vs *sei andata*) as well as articles (*il* vs *la*) depends on the noun.

Establishing the noun under consideration could improve MT quality significantly, an example of which is shown in (Babych and Hartley, 2003), wherein Named Entity Recognition benefit translation. This would eventually lead to less post-editing effort, which is significant for correcting coreference related errors (Daems et al., 2015). Other inter-sentential phenomena we would like to capture include temporality (precedence, succession), causality (reason, result), condition (hypothetical, general, unreal, factual), implicit assertion, contrast (juxtaposition, opposition) and expansion (conjunction, instantiation, restatement, alternative).

4 Experiments

4.1 Context integration

We model discourse using context windows on the source as well as the target side. For the source, we use one, two and three previous sentences and one next sentence as additional context. For the target, we use one and two previous sentences as additional context.[1] We choose the Transformer for our experiments. The non-recurrent architecture enables it to better handle longer sequences, without an additional computational cost. This

[1] Increasing beyond this caused a drop in performance in our preliminary experiments.

is made possible by using a multi-headed self-attention mechanism. The attention is a mapping from (query, key, value) tuples to an output vector. For the self-attention, the query, key and value come from the previous encoder layer, and the attention is computed as:

$$SA(Q, K, V) = softmax(QK^T/\sqrt{d_k})V \quad (5)$$

where Q is the query matrix, K is the key matrix and V is the value matrix, d_k is the dimensionality of the queries and keys, and SA is the computed self-attention. This formulation ensures that the net path length between any two tokens irrespective of their position in the sequence is O(1).

The multi-head attention makes it possible for the Transformer to model information coming in from different positions simultaneously. It employs multiple attention layers in parallel, with each head using different linear transformations and thereby learning different relationships, to compute the net attention:

$$MH(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (6)$$

where MH is the multi-head attention, h is the number of attention layers (also called "heads"), $head_i$ is the self-attention computed over the i^{th} attention layer and W^O is the parameter matrix of dimension hd_v*d_{model}. In this case, queries come from the previous decoder layer, and the key-value pairs come from encoder output.

For training the contextual models, we investigate the usage of all the possible combinations from the following configurations for modeling context on both sides:

- Source side configuration:
  - Previous sentence, previous two sentences, previous three sentences, previous and next sentence, previous two and next sentence.
- Target side configuration:
  - Previous sentence, previous two sentences.

For our experiments using the Transformer model, we concatenate the contextual information in our training and validation sets using a *BREAK* token, inspired by (Tiedemann and Scherrer, 2017). Since the Transformer has positional encodings, it encodes position information inherently and using just a single *BREAK* token worked better than appending a feature for each token specifying the sentence it belongs to. The models are referred to by the following label subsequently:

$$Prev_m + Curr + Next_n \rightarrow Prev_p + Curr$$

where *m* is the number of previous sentences used as source-side context, *n* is the number of next sentences used as source-side context, and p is the number of previous sentences used as target-side context. $Curr$ refers to the current sentence on both sides.

For comparison with RNN based techniques, we trained baseline as well as contextual models using a BiLSTM architecture. We employed the previous sentence as source-side context for the contextual models, integrated using the methods of concatenation and multi-encoder RNN proposed by Tiedemann and Scherrer (2017) and Jean et al. (2017) respectively. These are denoted by the labels *concat* and $Multi-Source$. For the concatenation, the $BREAK$ token was used, similar to the Transformer experiments. We also compared the performance using target-side context (Tiedemann and Scherrer, 2017; Bawden et al., 2017). The contextual models using only source-context are labeled "2 to 1", while those using the previous target sentence as context are labeled "2 to 2".

4.2 Dataset

For our experiments, we employ the IWSLT 2017 (Cettolo et al., 2012) dataset, for the language direction English → Italian (en → it). The dataset contains parallel transcripts of around 1000 TED talks, spanning various genres like Technology, Entertainment, Business, Design and Global issues.[2] We use the "train" set for training, the "tst2010" set for validation, and the "tst2017" set for testing. The statistics for the training, validation and test splits are as given in Table 1. For training the models, the sentences are first tokenized, following by segmentation of the tokens into subword units (Sennrich et al., 2015) using Byte Pair Encoding (BPE). The number of BPE operations is set to 32,000 and the frequency threshold for the vocabulary filter is set to 35.

[2] This dataset is publicly available at https://wit3.fbk.eu/

Phase	Training	Validation	Test
#Sentences	221,688	1,501	1,147
#Tokens-en	4,073,526	27,191	21,507
#Tokens-it	3,799,385	25,131	20,238

Table 1: Statistics for the IWSLT dataset

4.3 Model Settings

We employ OpenNMT-tf (Klein et al., 2017) for all our experiments.[3] For training the Transformer models, we use the Lazy Adam optimizer, with a learning rate of 2.0 , model dimension of 512, label smoothing of 0.1, beam width of 4, batch size of 3,072 tokens, bucket width of 1 and stopping criteria at 250,000 steps or plateau in BLEU, in case of the larger context models, since we observed some instability in the convergence behavior of the Transformer, especially for the contextual models. The maximum source length is set to be 70 for the baseline model, increasing linearly with more context. The maximum target length is set to be 10% more than the source length.[4] For training the RNN models, we employ the stochastic gradient descent optimizer, with a learning rate of 1.0, decay rate 0.7 with an exponential decay, beam width of 5, batch size 64, bucket width 1 and stopping criteria 250,000 steps or plateau in BLEU, whichever occurs earlier.

4.4 Evaluation

The evaluation of discourse phenomena in MT is a challenging task (Hovy et al., 2002; Carpuat and Simard, 2012), requiring specialized test sets to quantitatively measure the performance of the models for specific linguistic phenomena. One such test set was created by (Bawden et al., 2017) to measure performance on coreference, cohesion and coherence respectively. However, the test set was created with the assumption that the disambiguating context always lies in the previous sentence, which is not necessarily the case. Traditional automatic evaluation metrics do not capture discourse phenomena completely (Scarton et al., 2015), and using information about the discourse structure of a text improves the quality of MT evaluation (Guzmán et al., 2014). Hence, alternate methods for evaluation have been proposed (Mitkov et al., 2000; Fomicheva and Bel, 2016) However, these methods do not look at the document as a whole, but mainly model intra-sentential discourse. Developing an evaluation metric that considers document-level discourse remains an open problem. Hence, we perform a preliminary qualitative analysis in addition to the automatic evaluation of our outputs.

Configuration	BLEU	TER
(i) BiLSTM, no context	28.2	52.9
(ii) BiLSTM, Concat, 2 to 1	26.3	53.7
(iii) BiLSTM, Multi-Source, 2 to 1	28.9	52.6
(iv) BiLSTM, Concat, 2 to 2	25.4	53.4
(v) BiLSTM, Multi-Source, 2 to 2	28.9	52.5

Table 2: Performance using RNN based approaches

Model Configuration	BLEU	TER
(i) $Curr \rightarrow Curr$	29.2	52.8
(ii) $Prev_1 + Curr \rightarrow Curr$	29.4	52.5
(iii) $Prev_2 + Curr \rightarrow Curr$	29.8	51.9
(iv) $Prev_3 + Curr \rightarrow Curr$	29.2	52.8
(v) $Curr + Next_1 \rightarrow Curr$	29.7	51.9
(vi) $Prev_1 + Curr + Next_1 \rightarrow Curr$	30.6	51.1
(vii) $Prev_2 + Curr + Next_1 \rightarrow Curr$	29.8	51.4

Table 3: Results of our models using only source-side context, on en → it, IWSLT 2017

For automatic evaluation, we measure the performance of our models using two standard metrics: BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). For comparison with the test set, we extract the current sentence separated by the *BREAK* tokens from the output generated by the contextual models. We also measure the percentage of sentences for which the contextual models improve over the baseline model. This is done by computing the sentence-level TER for each generated output sentence, and comparing it with the corresponding one in the test set.

5 Results and Discussion

5.1 Performance on automatic evaluation metrics

Tables 3 and 4 show the results obtained by the different configurations of our models using the Transformer architecture. For comparison with previous approaches, we also train four contextual configurations using RNN-based models, and report the results in Table 2.

The RNN results confirm that:

- Adding contextual information is useful for RNN models, provided that it is incorporated using a multi-encoder architecture ($\approx$ 28.9

[3] The code is publicly available at https://github.com/OpenNMT/OpenNMT-tf

[4] This is done to ensure no loss in target-side information, a known sensitivity of the Transformer architecture.

Model Configuration	BLEU	TER
(i) $Prev_1 + Curr \rightarrow Prev_1 + Curr$	29.5	52.1
(ii) $Prev_2 + Curr \rightarrow Prev_1 + Curr$	29.8	51.9
(iii) $Prev_2 + Curr \rightarrow Prev_2 + Curr$	29.7	52.1
(iv) $Prev_3 + Curr \rightarrow Prev_1 + Curr$	29.2	52.2
(v) $Prev_3 + Curr \rightarrow Prev_2 + Curr$	28.9	52.9
(vi) $Prev_1 + Curr + Next_1 \rightarrow Prev_1 + Curr$	31.5	49.7
(vii) $Prev_2 + Curr + Next_1 \rightarrow Prev_1 + Curr$	31.1	50.5
(viii) $Prev_2 + Curr + Next_1 \rightarrow Prev_2 + Curr$	30.2	51.2

Table 4: Results of our models using source as well as target side context, on en → it, IWSLT 2017

Model Configuration	% sentences
$Prev_1 + Curr \rightarrow Curr$	62.8
$Curr + Next_1 \rightarrow Curr$	61.3
$Prev_1 + Curr + Next_1 \rightarrow Curr$	67.2

Table 5: Percentage of sentences for which TER score is less than or equal to the baseline model, depending upon the source-context used

BLEU score with multi-source, ≈ 0.8 more than the baseline BLEU score of 28.18).

- RNNs are sensitive to the length of the sentence, both on the source and target side (Table 2, (ii) and (iv)). This can be attributed to a vanishing signal between very long-range dependencies, despite the gating techniques employed.

- The RNN models need more sophisticated techniques than concatenation, like multi-source training, to leverage the information from the previous sentence (Table 2, (iii), (v)). This can be attributed to the drop in performance on very long sequences (Cho et al., 2014; Koehn and Knowles, 2017)[5], owing to concatenation.

For the Transformer architecture, the contextual models achieve an increase of 1-2% in BLEU score over a baseline model trained without any inter-sentential context (Tables 3 and 4).

The results suggest that:

- Looking further ahead at the next sentence can help in disambiguation, evident from the improved performance of the configurations involving both previous as well as next sentences on the source side than those looking only at previous context (Table 3, (v) - (vii)).

- Target-side context also helps to improve performance (Table 4, (i)-(v) vs. Table 3. (ii)-(iv)). as also suggested by (Bawden et al., 2017). However, a larger context window on the source side and a window with one previous sentence on the target side generally works better. Our intuition is that going beyond one previous sentence on the target side increases the risk of error propagation (Table 4, (viii)).

- The Transformer performs significantly better than RNN's for very long inputs (Table 2, (iv) vs. Table 4, (i)). This can be attributed to the multi-head self-attention, which captures long-range dependencies better.

- Contextual information does not necessarily come from the previous one sentence. Incorporating more context, especially on source-side, helps on TED data (Table 4, (vi), (vii)), and can be effectively handled with Transformer.

- The self-attention mechanism of the Transformer architecture enables a simple strategy like concatenation of a context window to work better than multi-encoder RNN based approaches.

Additionally, the training time for the Transformer models was significantly shorter than the RNN based ones (≈ 30 hours and ≈ 100 hours respectively). This can be attributed to the fact that the positional encodings capture the sequentiality in the absence of recurrence, and the multi-head attention makes it easily parallelizable. In addition to the corpus level scores, we also compute sentence level TER scores, in order to estimate the percentage of sentences which are better translated using cross sentential source-side context. These are given in Table 5.

5.2 Qualitative analysis

In addition to the performance evaluation using the automatic evaluation metrics, we also analyzed a random sample of outputs generated by our models, in order to have a better insight as to which linguistic phenomena are handled better by our contextual NMT models. Tables 6 and 7 compare the outputs of our best-performing contextual models (Table 4, (vi)) with the baseline model. The contextual models in general make better morphosyntactic choices generating more coherent translations than the baseline model. For instance, in the output of the contextual model (Table 6, (iii)), the

[5] On manual inspection, we observed frequent short, incomplete predictions in this case.

Source	I went there with **my friend**. She was amazed to see that it had multiple floors. **Each one** had a number of shops.
(i) Baseline Transformer	Arrivai li con **il mio amico**. Rimaneva meravigliato di vedere che aveva una cosa piu incredibile. **Ognuna** aveva tanti negozi.
(ii) Contextual Transformer (Prev)	Arrivai la con **il mio amico**. Era sorpresa vedere che aveva diversi piani. **Ognuno** aveva un certo numero di negozi.
(iii) Contextual Transformer (Prev + Next)	Sono andato con **la mia amica**. Fu sorpresa nel vedere che aveva piu piani. **Ognuno** aveva tanti negozi.
Reference	Sono andato la' con la mia amica. E' rimasta meraviglia nel vedere che aveva piu' piani. Ognuno aveva tanti negozi.

Table 6: Qualitative analysis - Improvement for cataphora, anaphora and gender agreement

Source	OK, I need you to take out your phones. Now that you have your phone out, I'd like you to unlock your phone.
(i) Baseline Transformer	Ok, devo **tirare** fuori i vostri cellulari. Ora che avete il vostro telefono, vorrei che bloccaste il vostro telefono.
(ii) Contextual Transformer (Prev)	OK, dovete tirare i vostri **cellulari**. Ora che avete il vostro telefono, vorrei che faceste sbloccare il vostro telefono.
(iii) Contextual Transformer (Prev + Next)	Ok, ho bisogno che **tiriate** fuori i vostri **telefoni**. Ora che avete il vostro telefono, vorrei che sbloccaste il vostro telefono.
Reference	Ok, ho bisogno che tiriate fuori i vostri telefoni. Ora che avete il vostro telefono davanti vorrei che lo sbloccaste.

Table 7: Qualitative analysis - Improvement for lexical cohesion and verbal inflections

phrase *sono andato* employs the *passato prossimo* ("near past") verb form *andato*, which is more appropriate than the *passato remoto* ("remote past") form *arrivai*, since the latter refers to events occurred far in the past, while the former refers to more recent ones. Additionally, the cataphor *my friend* is successfully disambiguated to refer to the postcedent *she*, apparent from the correctly predicted gender of the translated phrase *la mia amica* (feminine) as opposed to *il mio amico* (masculine). Similarly, the anaphora *Each one* is resolved (*ognuna* as opposed to *ognuno*). In the second example from Table 7, improved lexical choice -*che tiriate* (second person plural subjunctive), *bisogno* ("I need") as opposed to *devo* ("I must") and lexical cohesion *cellulari* ("mobile phones") *vs. telefoni* ("phones") can be observed.

While our models are able to incorporate contextual information from the surrounding text, they cannot leverage the disambiguating context which lies very far away from the current sentence being translated. In such cases, concatenating the sentences would be non-optimal, since there is a high possibility of irrelevant information overpowering disambiguating context. This is also evident from our experiments using n $>$ 2 previous sentences as additional context using concatenation (Table 3, (iv)).

6 Conclusion

Neural MT methods, being typically trained at sentence level, fail to completely capture implicit discourse relations established at the inter-sentential level in the text. In this paper, we demonstrated that looking behind as well as peeking ahead in the source text during translation leads to better performance than translating sentences in isolation. Additionally, jointly decoding the previous as well as current text on the target-side helps to incorporate target-side context, which also shows improvement in translation quality to a certain extent, albeit being more prone to error propagation with increase in the size of the context window. Moreover we showed that using the Transformer architecture, a simple strategy like concatenation of the context yields better performance on spoken texts than non-contextual models, whilst being trained significantly faster than recurrent architectures. Contextual handling using self-attention is hence a promising direction to explore in the future, possibly with multi-source techniques in conjugation with the Transformer architecture. In the future, we would like to perform a fine-grained analysis on the improvement observed for specific linguistic phenomena using our extended context models.

References

Babych, Bogdan and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, pages 1–8. Association for Computational Linguistics.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bawden, Rachel, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2017. Evaluating discourse phenomena in neural machine translation. *arXiv preprint arXiv:1711.00513*.

Bawden, Rachel. 2017. Machine translation of speech-like texts: Strategies for the inclusion of context. In *19es REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL 2017)*.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267.

Carpuat, Marine and Michel Simard. 2012. The trouble with smt consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449. Association for Computational Linguistics.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Panayota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. 2017. A comparative quality evaluation of pbsmt and nmt using professional translators.

Cettolo, Mauro, Girardi Christian, and Federico Marcello. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268.

Cettolo, Mauro, Federico Marcello, Bentivogli Luisa, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Chung, Junyoung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075.

Costa-Jussà, Marta R, Parth Gupta, Paolo Rosso, and Rafael E Banchs. 2014. English-to-hindi system description for wmt 2014: deep source-context features for moses. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 79–83.

Daems, Joke, Sonia Vandepitte, Robert Hartsuiker, and Lieve Macken. 2015. The impact of machine translation error types on post-editing effort indicators. In *4th Workshop on Post-Editing Technology and Practice (WPTP4)*, pages 31–45. Association for Machine Translation in the Americas.

Fomicheva, Marina and Núria Bel. 2016. Using contextual information for machine translation evaluation.

Giménez, Jesús and Lluís Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166. Association for Computational Linguistics.

Gimpel, Kevin and Noah A Smith. 2008. Rich source-side context for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17. Association for Computational Linguistics.

Guzmán, Francisco, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 687–698.

Hardmeier, Christian and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010.*, pages 283–289.

Hardmeier, Christian, Joakim Nivre, and Jörg Tiedemann. 2012. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190. Association for Computational Linguistics.

Hardmeier, Christian, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A document-level decoder for phrase-based statistical machine translation. In *ACL 2013 (51st Annual Meeting of the Association for Computational Linguistics); 4-9 August 2013; Sofia, Bulgaria*, pages 193–198. Association for Computational Linguistics.

Hardmeier, Christian. 2012. Discourse in statistical machine translation. a survey and a case study. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, (11).

Hovy, Eduard, Margaret King, and Andrei Popescu-Belis. 2002. Principles of context-based machine translation evaluation. *Machine Translation*, 17(1):43–75.

Jean, Sebastien, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Kalchbrenner, Nal and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Meyer, Thomas, Andrei Popescu-Belis, Najeh Hajlaoui, and Andrea Gesmundo. 2012. Machine translation of labeled discourse connectives. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, number EPFL-CONF-192524.

Mitkov, Ruslan, Richard Evans, Constantin Orasan, Catalina Barbu, Lisa Jones, and Violeta Sotirova. 2000. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pages 49–58. Citeseer.

Niehues, Jan, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider context by using bilingual language models in machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206. Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pierrehumbert, Janet and Julia Bell Hirschberg. 1990. The meaning of intonational contours in the interpretation of discourse. *Intentions in communication*, pages 271–311.

Sánchez-Martínez, Felipe, Juan Antonio Pérez-Ortiz, and Mikel L Forcada. 2008. Using target-language information to train part-of-speech taggers for machine translation. *Machine Translation*, 22(1-2):29–66.

Scarton, Carolina, Marcos Zampieri, Mihaela Vela, Josef van Genabith, and Lucia Specia. 2015. Searching for context: a study on document-level labels for translation quality estimation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tamchyna, Aleš, Alexander Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. 2016. Target-side context for discriminative models in statistical machine translation. *arXiv preprint arXiv:1607.01149*.

Tiedemann, Jörg and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92.

Toral, Antonio and Víctor M Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. *arXiv preprint arXiv:1701.02901*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Vintar, Špela, Ljupčo Todorovski, Daniel Sonntag, and Paul Buitelaar. 2003. Evaluating context features for medical relation mining. *Data Mining and Text Mining for Bioinformatics*, page 64.

Wang, Longyue, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. *arXiv preprint arXiv:1704.04347*.

Zoph, Barret and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

Towards a post-editing recommendation system for Spanish–Basque machine translation

Nora Aranberri
IXA research group
University of the Basque Country UPV/EHU
nora.aranberri@ehu.eus

Jose A. Pascual
School of Computer Science
The University of Manchester
jose.pascual@manchester.ac.uk

Abstract

The overall machine translation quality available for professional translators working with the Spanish–Basque pair is rather poor, which is a deterrent for its adoption. This work investigates the plausibility of building a comprehensive recommendation system to speed up decision time between post-editing or translation from scratch using the very limited training data available. First, we build a set of regression models that predict the post-editing effort in terms of overall quality, time and edits. Secondly, we build classification models that recommend the most efficient editing approach using post-editing effort features on top of linguistic features. Results show high correlations between the predictions of the regression models and the expected HTER, time and edit number values. Similarly, the results for the classifiers show that they are able to predict with high accuracy whether it is more efficient to translate or to post-edit a new segment.

1 Introduction

Although machine translation (MT) quality is getting better every day, neither regular users nor professional translators can blindly trust the correctness of a translation. Therefore, providing them with information about the quality together with the actual translation seems sensible. We could argue that this is relevant for regular users, who might not necessarily have a native-like command of the source and target languages. But it is of no lesser importance for professional translators who, being able to assess the quality themselves, might be able to speed up this process.

In this paper, we specifically focus on the case of Spanish–to–Basque professional translators. Note that MT quality for this language pair can be considered relatively poor (Aranberri et al., 2014; Aranberri et al., 2017) - at least that provided by freely accessible systems such as *itzultzailea*[1] or *Google Translate*[2]- and as a result, MT in the professional domain is very rarely used (Garmendia et al., 2017). In this context, we investigate whether we could build estimation models that may prove informative for translators and help with the integration of MT technology in this sector.

To that end, we use a small set of data collected in a post-editing workshop, where post-editing seems to benefit productivity at times. We first aim at providing professional translators with indicators of estimated work to guide their decision whether to post-edit or translate from scratch. For this, we build a set of regression models to estimate indicators of post-editing effort (overall MT quality, time and edits) which we obtain from data solely consisting of post-editing work. Results show high correlations over 0.70 between real and estimated indicators.

Nevertheless, it is undeniable that a recommendation model that suggests the most efficient editing approach would be a more direct way to help in such process. The recommendation could be used either to opt for the most efficient approach during editing or to filter out MT output before the editing phase starts. Thus, we build classification

[1] http://www.itzultzailea.euskadi.eus
[2] https://translate.google.com

models using linguistic features to recommend the editing approach that increases the productivity the most. However, given the low accuracy of the classifiers, we try to improve them by adding specific post-editing effort features. As this information is only available once the editing is completed, we estimate it using the above-mentioned regression models. Results show a large increase in the capacity of the classifiers to provide the correct editing approach even considering the loss of accuracy introduced by the regression models.

The remaining of the paper is structured as follows. A short overview of related work is presented in Section 2. In Section 3 we describe the data sets and features used to train the models while the experimental set-up is outlined in Section 4. Section 5 and Section 6 present the results for the regression and classification models, respectively. Finally, Section 7 summarizes the main conclusions and possible lines of future work.

2 Background

In this section we present an overview of the quality and post-editing effort indicators studied in the literature. In 2004, Blatz et al. (2004) brought confidence estimation techniques, mainly used in speech recognition until then, to the area of MT as they considered that these could help in filtering translations for post-editing, among other tasks. They built models for sentence-level annotation by training regressors and classifiers to predict NIST and WER values. Whereas the tasks themselves proved interesting, experiments revealed that estimated automatic metrics did not match human annotations of quality or post-editing effort.

Similar results were reported by Specia et al. (2009), who used a number of MT system-independent and MT system-dependent features to train a regression algorithm to estimate both NIST and human scores. The models performed well for human annotations, but once again, correlations with automatic metrics were not as successful. From then on, Specia and Farzindar (2010) tested the use of TER and HTER (Snover et al., 2006), which supposedly consider the actual post-editing work translators perform more closely, to build the estimation models. This time, the models correlated well with human annotations of post-editing effort. For that reason, HTER was established as the global quality indicator in quality estimation (QE) tasks and remains so today, despite attempts at looking for alternative ways of measuring quality (Specia et al., 2011).

Since then, a number of authors have worked on building models to provide translators with useful information. Some have tried to describe post-editing time (Specia, 2011) whereas others have focused on selecting the best MT output from a pool of candidates (Avramidis et al., 2011), or on recommending whether a source segment should be tackled using a MT candidate or a translation memory candidate (He et al., 2010). However, it could be argued that the main bulk of research in quality estimation has been shaped by the yearly QE Shared Task, in place since 2012. In its first year, participants focused on correlating estimation models with manual annotations of quality defined as 5 levels of post-editing effort (Moreau and Vogel, 2012; Hardmeier et al., 2012). In 2013 and 2014, the goals were broadened and tasks involved predicting HTER and post-editing time and ranking MT candidates (Beck et al., 2014; Bicici and Way, 2014). Since then, however, efforts have mainly addressed HTER and even if submissions for other indicators such as post-editing time and keystrokes have been welcome, no results have been published on these aspects.

In order to provide professionals with a wider set of pointers that guides the translation task, in this paper we expand the post-editing effort indicators. Specifically, we propose to create a recommendation system that (1) estimates the quality of the MT output as defined by HTER, (2) predicts post-editing effort according to time, and the type and number of edits, and (3) recommends the editing approach for a particular segment by classifying it for either post-editing or translation from scratch.

3 Data Collection and Processing

Unlike for other mainstream language pairs, no readily-available data exists to train quality estimation models for the Spanish–Basque pair. For this purpose, therefore, we adapted post-editing data collected in a workshop for professional translators run in 2015. In this section we describe the data and the linguistic features that we used to build the estimation models.

3.1 Data Sets

In the above-mentioned workshop, translators worked on a series of post-editing tasks and pro-

Task Number	Task Type	Translators	Text	MT System	Sentences	Source Words
1–4	post-editing	10	1	itzultzailea	60	1,467
5	productivity	10	1	itzultzailea	21	495
6–9	post-editing	10	1	itzultzailea	81	1,958
10	productivity	9	1	itzultzailea	16	506
11–14	post-editing	8	1	itzultzailea	82	2,043
15	productivity	8	1	itzultzailea	22	366
16–19	post-editing	8	1	itzultzailea	80	1,964
20	productivity	8	1	itzultzailea	29	516
21–24	post-editing	8	1	itzultzailea	138	2,045
25	productivity	8	1	itzultzailea	26	515
26–29	post-editing	6	1	Google Translate	121	2,082
30	productivity	6	1	Google Translate	24	508
31–34	post-editing	5	2	itzultzailea	187	2,012
35	productivity	5	2	itzultzailea	60	486

Table 1: List of total tasks performed by professional translators.

ductivity tests over a period of seven weeks (See Table 1). For the productivity tests, translators alternately post-edited and translated source sentences. We divided translators in two groups who performed the opposing editing approach for each segment. Throughout the workshop they translated a report by the Basque Institute of Women about Sexism in toys advertising (Text 1) and two short user guides for a mobile phone and a washing machine (Text 2). The original Spanish texts were translated using *itzultzailea*, the MT system made publicly available by the Basque Government and powered by Lucy. The overall MT output was of relative low quality (~ 50.7 HTER) and translators introduced a significant number of edits to turn the segments into acceptable translations.

Task	Avg. PE time/word	Avg. TR time/word
5	4576.73	**4353.46**
10	3058.86	3882.97
15	2920.31	4400.37
20	3454.05	4224.66
25	3174.79	3520.80
30	3523.23	**2974.36**
35	3054.51	291.58

Table 2: Average post-editing and translation time (ms) per word for each productivity task performed by translators.

For our experiments (see Section 4.1), we divided the data collected in the tasks into two sets, namely, the post-editing (PE) set and the productivity (PR) set. The former includes all the segments from the post-editing tasks whereas the latter includes those from the productivity tests. We discarded all tasks performed during the first week (tasks 1–5) as this was the first contact translators had with post-editing and therefore their work was deemed unreliable. Also, we decided to discard tasks 26–30, as they were performed using a different MT system, with which the translating time appears to be lower than the post-editing time (See Table 2). As a result, we collected work for 568 source segments (10,022 words) from the post-editing task and 153 segments (2,389 words) from the productivity test. Note that because all translators were asked to perform the same tasks, our sets include information about several final translations for each of the source segments.

Finally, we added the information required to train the models which is not present in the original data to both sets. Firstly, in order to build models to predict the post-editing indicators, we added HTER scores, the number of each edit-type and the total number of edits to the PE set. Editing times were already present. Secondly, for the classification models, we added to the PR set labels referring to the editing approach that benefits each source segment the most. As opposed to the method used in the 2012 QE Task where manual annotation of perceived post-editing effort was performed by professional translators according to a 5-level scale, our strategy to assign the labels mainly relied on the time gain introduced by the fastest approach. To this end, we used the productivity ratio (translation time/post-editing time). In our case, we calculated the ratio for each source segment with the averaged editing times of the different translators to account for translator variability. Scores above 1 indicate that post-editing is more productive whereas scores below 1 indicate the extent to which translation is faster.

We used three sets of labels, L2, L3 and L5 which involve two, three and five labels, respectively. L2 directly assigns a *post-edit* label to all ratios above 1 and a *translate* label to all ratios below 1. L3 considers that, given the editing variability among translators, scores close to 1 may not reliably predict the most effective approach nor indicate much time difference between them. Therefore, ratios ranging between 0.90–1.10 are assigned the *any approach* label. Finally, L5 adds two extra labels to the L3 set which identify those segments that are clearly more efficient to either post-edit (above 1.30) or translate (below 0.70).

3.2 Features

We extracted the same set of 17 baseline features provided by the WMT12-17 QE Tasks using *Quest++* (Specia et al., 2015). They are black-box features, that is, shallow MT system-independent features. Most of them rely on the comparison of the sentences against a large training corpus, e.g. language model probabilities, n-gram frequencies and translation options per word.

The monolingual Spanish and Basque corpora we used to this end consist of 38 and 44 million segments, respectively. The Spanish corpus includes data released for the WMT tasks (Europarl corpus, UN corpus, News Commentary corpus, etc.). The Basque one comprises texts from different sources such as the Basque newspaper *egunkaria* and radio–television *EITB*, the Elhuyar Web Corpus and administrative translation memories. The bilingual corpus used to train GIZA++ is a considerably smaller set of 7.8 million segments. Overall, the corpora, and specially the monolingual sections, are of a good size to model the relevant languages. However, the domain of our data sets is not represented in them, which could significantly harm the accuracy of the features.

For this reason, we tried to overcome this drawback by adding linguistic information directly extracted from the segments in the data sets. We want to remark, however, that it is not the aim of this work to do feature ingeneering as in Specia and Felice (2012) and Avramidis (2012). For Spanish, we processed the text using *ixa-pipes* tools (Agerri et al., 2014) and for Basque, we used *ixaKat* (Otegi et al., 2016). We collected POS frequencies, tags for morphological features and dependency relations for both source and target segments. Therefore, we added a feature for each POS, morphological feature and dependency relation, whose value was the number of times it appeared in the segment (10, 185 and 42 features for Spanish, respectively, and 10, 316 and 28 for Basque). However, preliminary tests showed that no improvement was coming from the morphological features so we decided to discard them.

For the experiments, we therefore use four different data sets. PE-17 and PR-17 include the baseline features only and PE-107 and PR-107 also use the additional linguistic features.

4 Experimental Set-up

In this section we explain the experiments carried out to predict the MT quality and the post-editing effort required to transform the MT output into the desired quality standard.

4.1 Experiments

We divided the experiments into three distinct parts. In the first part we evaluate the ability of five regression algorithms to learn a number of models to predict indicators of post-editing effort. The indicators are as follows:

- **HTER:** This metric is used as a global quality measure for the professional translator. It is an edit-distance metric that considers the number of edits to be made to a MT segment to transform it into the desired final translation normalized by the number of words in the reference sentence.

- **Post-editing time:** This indicator accounts for the time required by a professional translator to transform the MT output into the desired final text. We give the estimates in milliseconds per segment.

- **Edit types:** This indicator provides individual information for each type of edit, i.e., insertions, deletions, substitutions and shifts, to be introduced to the MT output as computed by HTER. Although the mathematical approach used by the HTER metric to calculate the edits often differs from the linguistically-motivated instinct of translators, they might prove useful in gauging the complexity of the expected post-editing effort.

- **Number of edits:** This is a raw indicator of the number of edits to be made to the MT output to reach the desired quality as computed by HTER. Whereas the edit types are more informative, this provides a rawer measurement of the overall changes.

The second set of experiments is devoted to building and measuring the capacity of classification models to suggest whether a source segment should be translated or post-edited.

However, given the limited data available, we expected these models to have low accuracy. For this reason, we also proposed and evaluated a third set of experiments in which the features of the second data set are incremented with indicators of post-editing effort. To do so, we train the models with real post-editing effort indicators even if these are not available for new segments. Then we apply the regression models described in the first set of experiments to predict these additional features for the new segments before testing (See Figure 1). We expect that the accuracy of the classifiers will increase with these additional features.

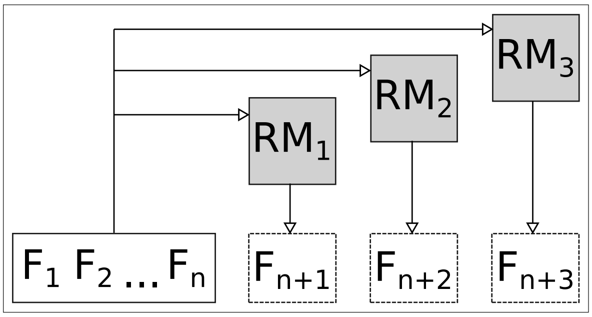


Figure 1: Representation of the extension of the number of features from n to $n+3$ using three previously trained regression models.

In all the experiments the learning and testing process was carried out using 10-fold cross-validation over the PE and PR data sets. The accuracy of the regression models was measured using the correlation coefficient (ρ) which measures the strength and the direction of a linear relationship between two variables. On the other hand, the accuracy of the classifiers was measured using the area under the curve ROC. Each experiment was repeated 10 times and we report the average (μ_ρ and μ_{ROC}) and the standard deviation (σ_ρ and σ_{ROC}). We also performed a paired t-test ($p < 0.05$) to check statistical significance of the results of the algorithms (in bold). We also checked, for each algorithm, if the addition of linguistic features is significant (with the symbol †) .

4.2 Regression and Classification Algorithms

There are countless machine learning algorithms to train regression and classification models. As the purpose of our experiments is to explore the ability of these algorithms to train the recommendation system, we selected six of the most used ones to have an insight into their individual performance.

- Linear regression (LR): It is used for *regression* and it works by estimating coefficients for a line or hyperplane that best fits the training data. It is fast to train and can have great performance if the output is a linear combination of the inputs.

- Logistic regression (LG): It is a regression model (used for *classification*) that estimates the probability of class membership as a multi-linear function of the features.

- k-Nearest Neighbors (k-NN): This algorithm supports both *classification* and *regression*. It works by storing the training dataset and locating the k most similar training patterns to perform a prediction.

- Classification And Regression Trees (CART): They work by creating a tree to evaluate an instance of data, starting at the root of the tree and moving down to the leaves until a prediction can be made. They support both *classification* and *regression*.

- Support Vector Machine (SVM): This is an algorithm for *classification* which finds a line that best separates the training data into classes. The adaptation of SVM for *regression* is called Support Vector Regression (SVR) and works by finding a line that minimizes the error of a cost function. In both cases we use a polynomial kernel.

- Multi-Layer Perceptron (MLP): This algorithm supports both *regression* and *classification* problems using neural networks.

We want to remark that this is a first attempt to measure the quality of the predictions leaving as future work the fine tuning of these algorithms.

5 Results of Regression Models for Quality and Post-editing Work

In this section we present the regression models that aim to predict the post-editing effort. We report the results for each indicator, namely, overall quality, time and edits, separately using both the PE–17 and PE–107 data sets.

5.1 Overall Quality with HTER

Let us start by analyzing the results to estimate segment quality (HTER) by focusing on the PE–17 data set (see Table 3). The results show that the correlation coefficient obtained by k-NN is the highest at 0.71, closely followed by CART. LR and SVR obtain the poorest results with a notably lower correlation coefficient of 0.35 and 0.32, respectively. This suggests that neither LR nor SVR are able to model the relation between the features and the HTER values. In order to confirm this, we performed a test to measure the correlation between the features and HTER, which showed that except for three cases, correlations were lower than 0.1. Indeed, these algorithms are best fitted to capture liner relations and a quick test using a non-linear kernel in SVR revealed an increase of the average correlation to 0.68±0.04 in PE–17 and to 0.70±0.04 in PE–107.

	PE–17		PE–107	
Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ
LR	0.3499	0.0399	0.4509†	0.0373
k-NN	**0.7146**	**0.0220**	**0.7144**	**0.0218**
CART	0.6704	0.0367	0.6685	0.0347
SVR	0.3211	0.0415	0.4126†	0.0335
MLP	0.4704	0.0517	0.5870†	0.0456

Table 3: Regression results for the HTER model.

If we compare these results with those obtained using the PE–107 data set to analyze the impact of the linguistic features in the learning process, we observe that for the best performing algorithms in PE–17, k-NN and CART, the contribution of the new features is non-existent. However, the remaining three algorithms do benefit from the addition of the new features significantly. This suggests a stronger linear relation between the features and the HTER values. This was confirmed by testing this relation, which showed that the number of features with a correlation higher than 0.1 with the HTER values had increased to 25.

5.2 Post-editing Time

Previous attempts at estimating time have shown that it is quite an objective indicator for post-editing effort. Looking at the results for the PE-17 data set (see Table 4) we see that, unlike for HTER, all the algorithms perform very similarly (differences not statistically significant) and obtain a correlation coefficient of around 0.71. In this case, the correlation between the features and time is higher than 0.2 for 8 of the features, which explains the good behavior of LR and SVR.

	PE–17		PE–107	
Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ
LR	0.7137	0.0402	**0.7238**	**0.0372**
k-NN	0.7106	0.0362	0.7131	0.0366
CART	0.7081	0.0392	0.7092	0.0383
SVR	0.7135	0.0405	**0.7265**	**0.0388**
MLP	0.7122	0.0380	0.6955	0.0436

Table 4: Regression results for the time model.

If we examine the results for PE-107, we notice that the contribution of the new linguistic features is not significant for any of the algorithms. We observe, however, that LR and SVR benefit the most from them and obtain the highest results, which are statistically significant in this data set. An analysis of the relation between the features and time showed a correlation higher than 0.2 for 47 features and higher than 0.1 for another 21.

5.3 Edit Types and Total Number

Not much has been published on estimating the different types of edits required to transform the MT output into the desired final version. Avramidis (2014; 2017) trained models for each edit type to then combined them, to try to obtain a higher accuracy HTER model. However, the potential value of the individual models was not considered. If we look at the results, we see that, given their accuracy, we could in fact include them in the recommendation system as part of the information about post-editing effort provided to translators.

Let us consider the different edit types in PE–17 first (see Table 5). We observe that for all the models, k-NN is the best performing algorithm. However, the level of accuracy of the models varies considerably for the different edit types. The model for substitutions is by far the best performing one, with a correlation coefficient of 0.80. Shifts and insertions also get good results. But the

		INSERTIONS		DELETIONS		SUBSTITUTIONS		SHIFTS		TOTAL EDITS	
Dataset	Alg	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ	μ_ρ	σ_ρ
PE–17	LR	0.5685	0.0427	0.4537	0.0421	0.7336	0.0180	0.6167	0.0266	0.8029	0.0180
	k-NN	**0.7011**	**0.0687**	**0.5214**	**0.0435**	**0.8035**	**0.0182**	**0.7422**	**0.0238**	**0.8660**	**0.0168**
	CART	0.6556	0.0930	0.4896	0.0459	0.7876	0.0187	0.7164	0.0252	0.8550	0.0176
	SVR	0.5625	0.0273	0.4517	0.0415	0.7325	0.0179	0.6147	0.0274	0.8020	0.0187
	MLP	0.6633	0.0740	0.4731	0.0488	0.7404	0.0205	0.6344	0.0268	0.8125	0.0198
PE–107	LR	0.6167	0.0610	0.4840	0.0394	0.7566	0.0187	0.6710†	0.0257	0.8247	0.0175
	k-NN	**0.7010**	**0.0688**	**0.5214**	**0.0435**	**0.8035**	**0.0182**	**0.7423**	**0.0238**	**0.8660**	**0.0167**
	CART	0.6571	0.0893	0.4874	0.0452	0.7872	0.0190	0.7184	0.0249	0.8558	0.0182
	SVR	0.5750	0.0306	0.4723	0.0373	0.7505	0.0190	0.6567	0.0274	0.8193	0.0179
	MLP	0.6664	0.0758	0.4746	0.0466	0.7364	0.0332	0.6674	0.0370	0.8210	0.0268

Table 5: Regression results for the individual edit type and total edits models.

coefficient score for deletions is low at 0.52, showing that the algorithms are not able to capture the relation between the features and this indicator.

If we take a look at the total number of edits, irrespective of their type, we observe that the prediction models perform very well. Again, k-NN is the best performing algorithm with a correlation coefficient of 0.86 but all five score above 0.80.

As with previous regression models, we notice that the new linguistic features added in PE–107 make no or only a marginal contribution to the learning process and in no case improve the results of the best performing algorithm.

5.4 Summary

In summary, we see that we are able to train models that predict HTER and time with a relatively high accuracy and within the range reported by other research despite the limited training data. It is true that the overall performance should be improved, and the models trained and tested on additional data sets before these indicators are provided to translators. However, the results are very promising as there is ample room for tuning.

In reference to edits, regression models perform well in general, although there is strong variation across types. Room from tuning aside, it is worth considering that not all edit types may have the same weight for translators when assessing the work involved during post-editing. Insertions and substitutions require intensive work where translators either add missing information or replace incorrect MT output. Shifts are lower intensity edits, where the correct translation is present, just not in the correct place. These three edit types achieve correlation coefficients of over 0.70 and we could provide them with confidence after additional tuning tests. Deletions, however, score poorly but these could be viewed as very low intensity edit types where translators would easily identify the incorrect elements to eliminate. Therefore, they might not be the edit type that represents the most laborious aspect of post-editing. It remains to be tested which of the types translators find most informative regarding post-editing effort.

Predicting the total number of edits has been much more successful. It may not be as informative as having predictions for the different types but considering the distinct nature of the approach to editing used by humans and machines, it might prove a good compromise that measures the effort in terms of raw changes.

What is interesting to see is the difference in performance between the HTER and the total edit number models, as the latter is based on HTER information. For some reason, the regression models and features seem to be better suited for predicting the errors without considering the length of the final translations. The significantly higher scores obtained for the total edits makes us consider whether providing HTER scores as indication of post-editing effort is appropriate or whether providing raw edit numbers together with sentence lengths would be more accurate and informative.

Finally, it is worth noting that features accounting for the frequencies of POS and dependency relations only contribute to the learning process in a few cases further than the 17 baseline features.

6 Results of Classifiers for Editing Approach

In this section we present the results for the classifiers that aim to predict the editing approach, post-editing or translation, a translator should follow when addressing a new segment. We report results with and without additional linguistic features and also analyse the impact of using post-editing effort indicators as features. We predict label-sets of varying numbers of classes.

6.1 Baseline classification models

We first present the results for the baseline classification models. We trained the models with all available segments in the productivity data set.

We check the results obtained for the 2 label task (2L) first (see Table 6). For both the PR–17 and PR–107 sets, all algorithms perform very poorly with μ_{ROC} below 0.60, with SVR lagging behind (statistical difference). However, if we consider the PR–107 data set, we see that thanks to the additional linguistic features SVR has caught up with the other algorithms (statistical significance between PR–17 and PR–107).

	PR–17 (2L)		PR–107 (2L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	**0.56**	**0.15**	0.60	0.15
k-NN	**0.58**	**0.11**	0.56	0.11
CART	**0.51**	**0.11**	0.49	0.09
SVR	0.50	0.02	$0.57^{\dagger}$	0.11
MLP	**0.59**	**0.13**	0.58	0.17

Table 6: Results of the classification algorithms for 2 labels.

Let us now take a look at the results for 3 labels (3L) (see Table 7). In this case there is no statistically significant difference between the algorithms. Same as before, adding linguistic features only benefits SVR (statistical significance).

	PR–17 (3L)		PR–107 (3L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	0.53	0.17	0.50	0.17
k-NN	0.56	0.11	0.55	0.13
CART	0.50	0.11	0.49	0.10
SVR	0.48	0.06	$0.57^{\dagger}$	0.13
MLP	0.58	0.14	0.56	0.16

Table 7: Results of the classification algorithms for 3 labels.

We observe the same trend of poor results for the 5 label task (5L) (see Table 8). However, in this task, additional linguistic features do not bring any improvement. In fact, the only statistical significance is the setback for LG.

Overall, we conclude that the performance of the classification models is far from being accurate enough to prove useful in a real set-up. Even with room for tuning, we believe that the current features do not properly inform the algorithms for the classification task.

	PR–17 (5L)		PR–107 (5L)	
Alg	μ_{ROC}	σ_{ROC}	μ_{ROC}	σ_{ROC}
LG	$0.57^{\dagger}$	0.25	0.40	0.24
k-NN	0.57	0.15	**0.56**	**0.15**
CART	0.54	0.13	**0.55**	**0.13**
SVR	0.57	0.22	**0.52**	**0.22**
MLP	0.69	0.22	**0.55**	**0.22**

Table 8: Results of the classification algorithms for 5 labels.

6.2 Classification models using predictions for post-editing work

In an attempt to improve the performance of the classification models, we propose to use indicators of post-editing effort as features for training. We believe that these indicators reflect more closely the reasons why a translator would choose one editing approach over the other. For that, we first analyse whether the previous classifiers perform better by adding original HTER, total edits and time as features to the PR set. Secondly, as these three features are not available until translation is completed, we test new classifiers with predicted post-editing features (see Section 4.1).

We summarize the results of adding the three post-editing effort indicators as features in Table 9. Results are given by μ_{ROC}, which corresponds to the average results of the training set. For the sake of space we omit the standard deviations of the learning process. We can see that in the PR–20 set the accuracy of the models varies from fair to excellent regardless of the number of labels. This is a large improvement over the baseline classifiers that reveals the potential of adding HTER, time and edit number as features. Whereas k-NN, CART and MLP are the best performing algorithms across all sets, LG and SVR are the worst scoring for PR-20. However, as in the regression

	PR–20			PR–110		
Alg	2L	3L	5L	2L	3L	5L
LG	0.76	0.74	0.80	**0.99**†	**1.00**†	**0.99**†
k-NN	**0.99**	**0.99**	**0.98**	**1.00**†	**1.00**†	**1.00**†
CART	**0.98**	**0.99**	**1.00**	**0.99**	**1.00**	**1.00**
SVR	0.64	0.63	0.77	$0.91^{\dagger}$	$0.93^{\dagger}$	$0.96^{\dagger}$
MLP	**0.97**	**0.96**	**0.95**	**1.00**	**0.99**	0.96

Table 9: Results of the classification algorithms using the post-editing effort features given by μ_{ROC}.

experiments, we see that when new linguistic features are added (PR-110), the performance of LG and SVR improves (statistical significance). Interestingly, the performance of k-NN also benefits from the linguistic features.

Given the promising results obatined when adding the post-editing effort indicators, we test this approach using a scenario viable for deployment. We divide the productivity data set into a training and a test set. Out of the 153 unique source segments, we randomly include 80% in the training set and 20% in the test set. The training set includes all the available data for each of the unique segments. The test set, in turn, only includes one instance of each unique segment with HTER, time and total edits as predicted by the best-performing models in Section 6 on top of the initial features (PR–20 and PR–110 after adding the 3 new features). Results, given by T_{ROC}, the ROC value obtained after applying one of the learnt models to the test set, are summarized in Table 10.

	PR–20			PR–110		
Alg	2L	3L	5L	2L	3L	5L
LG	0.66	0.69	0.75	0.70	0.73	0.75
k-NN	**0.89**	0.89	0.90	**0.89**	**0.99**	0.90
CART	0.88	**0.90**	0.89	**0.89**	0.90	0.90
SVR	0.56	0.55	0.60	0.83	0.85	0.87
MLP	0.88	0.83	**0.92**	0.88	0.90	**0.99**

Table 10: Results of the classification algorithms using the post-editing effort features given by T_{ROC}.

With this set-up, the best scoring models range between good and excellent for both PR–20 and PR–110 sets and for all the label sets. Notice that our predictions carry over the margin of error of the regression models, but still their level of accuracy is very high. In the PR–20 set, MLP is the best performing algorithm and LG and SVR perform very poorly. In the PR–110 set, the same best performers remain on top but k-NN and CART perform particularly well for 2 and 3 labels, and MLP for 5 labels. It is also worth noting the improvement of SVR in this data set. Overall we can argue that the results for the classification model are promising and could be useful in a real setting. Even more, we expect further improvement from tuning the classifiers and from obtaining more accurate predictions from the regression models.

7 Conclusions and Future Work

In this paper we tested the feasibility of training a number of estimation models that go beyond the usual MT quality level to build a recommendation system that helps speed up the decision time of professional translators to decide whether to post-edit or translate. In particular, we studied if reasonable results could be obtained for the Spanish–Basque pair, for which MT quality is low, and thus not widely used within the professional sphere.

We trained regression models to predict HTER, time, types and total number of edits as indicators of post-editing effort using a limited data set. We show that relatively high correlation coefficients can be achieved for almost all indicators. The total edit number seems the easiest to predict whereas accuracy is lower for each of the individual types, particularly for deletions. Results also reveal that adding POS and dependency relation frequencies as features did not generally improve the majority of our models. k-NN was the best performing algorithm, with the best results for HTER and all the models involving edits and with no contribution from the new linguistic features. For the time model, LR and SVR performed best, obtaining marginal gains with the new features.

Besides providing post-editing effort indicators, we also trained a classification model that would recommend translators the editing approach to take. Whereas the baseline models performed poorly, we showed that including post-editing effort indicators as features largely improves the results. As this information is not available for new segments, we successfully used previously trained regression models to add these features in new test sentences. k-NN, CART and MLP consistently show the best performance across all the data sets.

Given the good results achieved, the next step would involve tuning and testing the models in further data sets. Our aim is to investigate to what extent HTER, post-editing time and edit types are valuable indicators for professionals translators.

Acknowledgements: The research leading to this work was partially funded by the *TIN2015-70214-P* project (MINECO-FEDER) and the *KK-2017/00094* project (Basque Government).

References

Agerri, Rodrigo, Josu Bermudez and German Rigau. 2014. IXA pipeline: Efficient and Ready to Use

Multilingual NLP tools. *LREC2014, 9th Language Resources and Evaluation Conference*, Reykjavik, Iceland. 3823–3828.

Aranberri, Nora, Gorka Labaka, Arantza Diaz de Ilarraza, and Kepa Sarasola. 2014. Comparison of post-editing productivity between professional translators and lay users. *Third Workshop on Post-editing Technology and Practice*, Vancouver, Canada. 20–33.

Aranberri, Nora. 2017. What Do Professional Translators Do when Post-Editing for the First Time? First Insight into the Spanish-Basque Language Pair. *HERMES-Journal of Language and Communication in Business*, (56):89–110.

Aranberri, Nora, Gorka Labaka, Arantza Diaz de Ilarraza, and Kepa Sarasola. 2017. Ebaluatoia: crowd evaluation for English-Basque machine translation. *Language Resources and Evaluation*, 51(4):1053–1084.

Avramidis, Eleftherios. 2017. Sentence-level quality estimation by predicting HTER as a multi-component metric. *WMT-2017, Conference on Machine Translation*, Copenhagen, Denmark. 534–539.

Avramidis, Eleftherios. 2014. Efforts on Machine Learning over Human-mediated Translation Edit Rate. *WMT-2014, 9th Workshop on Statistical Machine Translation*, Baltimore, Maryland. 302-306.

Avramidis, Eleftherios. 2012. Quality estimation for machine translation output using linguistic analysis and decoding features. *WMT-2012, Seventh workshop on statistical machine translation*, Montreal, Canada. 84–90.

Avramidis, Eleftherios, Maja Popovic, David Vilar, and Aljoscha Burchardt. 2011. Evaluate with confidence estimation: machine ranking of translation outputs using grammatical features. *WMT-2011, Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland. 65–70.

Beck, Daniel, Kashif Shah and Lucia Specia. 2014. SHEF-Lite 2.0: Sparse Multi-task Gaussian Processes for Translation Quality Estimation. *WMT-2014, Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland. 307-312.

Bicici, Ergun and Andy Way. 2014. Referential Translation Machines for Predicting Translation Quality. *WMT-2014, Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland. 313-321.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. *ACL-2004, 42th Annual Meeting of the Association for Computational Linguistics*, Geneva, Switzerland. 315–321.

Garmendia, Lierni, Naroa Lasarte and Maialen Pinar. 2017. Situación actual y viabilidad de la TA en euskera: posedición y análisis de los resultados de un motor de TABR español-euskara. *Master Thesis*, Universitat Autònoma de Barcelona.

Hardmeier, Christian, Joakim Nivre and Jorg Tiedemann. 2012. Tree kernels for machine translation quality estimation. *WMT-2012, 7th Workshop on Statistical Machine Translation*, Montreal, Canada. 109–113.

He, Yifan, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation. *ACL-2010, 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden. 622–630.

Moreau, Erwan and Carl Vogel. 2012. Quality estimation: an experimental study using unsupervised similarity measures. *WMT-2012, 7th Workshop on Statistical Machine Translation*, Montreal, Canada. 120–126.

Otegi, Arantxa, Nerea Ezeiza, Iakes Goenaga, and Gorka Labaka. 2016. A Modular Chain of NLP Tools for Basque. *TSD 2016, 19th International Conference on Text, Speech and Dialogue*, Brno, Czech Republic. 93–100.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *AMTA-2006, 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts. 223–231.

Specia, Lucia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. *EAMT-2009, 13th Conference of the European Association for Machine Translation*, Barcelona, Spain. 28–37.

Specia, Lucia, and Atefeh Farzindar. 2010. Estimating Machine Translation Post-Editing Effort with HTER. *AMTA-2010, Workshop Bringing MT to the User: MT Research and the Translation Industry*, Denver, Colorado. 33–41.

Specia, Lucia, Najeh Hajlaoui, Catalina Hallett and Wilker Aziz. 2011. Predicting machine translation adequacy. *Machine Translation Summit XIII*, Xiamen, China. 19–23.

Specia, Lucia. 2011. Exploiting objective annotations for measuring translation post-editing effort. *EAMT-2011, 15th Conference of the European Association for Machine Translation*, Leuven, Belgium. 73–80.

Specia, Lucia and Mariano Felice. 2012. Linguistic features for quality estimation. *WMT-2012, Seventh workshop on statistical machine translation*, Montreal, Canada. 96–103.

Specia, Lucia, Gustavo Henrique Paetzold and Carolina Scarton. 2015. Multi-level Translation Quality Prediction with QuEst++. *ACL-IJCNLP 2015 System Demonstrations*, Beijing, China. 115–120.

Compositional Source Word Representations for Neural Machine Translation

Duygu Ataman
FBK, Trento, Italy
University of Trento, Italy
ataman@fbk.eu

Mattia A. Di Gangi
FBK, Trento, Italy
University of Trento, Italy
digangi@fbk.eu

Marcello Federico
MMT Srl, Trento, Italy
FBK, Trento, Italy
federico@fbk.eu

Abstract

The requirement for neural machine translation (NMT) models to use fixed-size input and output vocabularies plays an important role for their accuracy and generalization capability. The conventional approach to cope with this limitation is performing translation based on a vocabulary of sub-word units that are predicted using statistical word segmentation methods. However, these methods have recently shown to be prone to morphological errors, which lead to inaccurate translations. In this paper, we extend the source-language embedding layer of the NMT model with a bi-directional recurrent neural network that generates compositional representations of the source words from embeddings of character n-grams. Our model consistently outperforms conventional NMT with sub-word units on four translation directions with varying degrees of morphological complexity and data sparseness on the source side.

1 Introduction

Neural machine translation (NMT) has improved the state-of-the-art performance in machine translation of many languages (Bentivogli et al., 2016; Junczys-Dowmunt et al., 2016). However, current NMT systems still suffer from poor performance in translating infrequent or unseen words, limiting their deployment for translating low-resource and morphologically-rich languages. This problem is mainly caused by the fundamental design of the model, which requires observing many examples of a word until its input representation (*i.e.* embedding) becomes effective. Moreover, the convention of limiting the input and output vocabularies to few tens of thousands of words to control the computational complexity of the model leads to coverage issues. In fact, a word can be translated only if an exact match of it is found in the vocabulary.

To cope with this well-known problem, several studies have suggested to redefine a new model vocabulary in terms of the interior orthographic units compounding the words, such as character n-grams (Costa-Jussa and Fonollosa, 2016; Lee et al., 2016; Luong and Manning, 2016) or statistically-learned sub-word units (Sennrich et al., 2016; Wu et al., 2016; Ataman et al., 2017). In spite of providing an ideal open vocabulary solution, the former set of approaches mostly failed to achieve competitive results. This might be related to the semantic ambiguity caused by solely relying on embeddings of character n-grams which are generally learned by disregarding any lexical context, hence, morphology. In fact, building a vocabulary of sub-word units for training the NMT model and performing translation based on subword embeddings has now become the prominent approach. However, many studies have shown that statistical word segmentation methods can break the morphological structure of words, leading to loss of semantic and syntactic information in the sentence and, consequently, inaccurate translations (Niehues et al., 2016; Ataman et al., 2017; Pinnis et al., 2017; Huck et al., 2017; Tamchyna et al., 2017). Principally, these solutions are unsupervised methods and can never reach the accuracy of morphological analyzers, which, on the other hand, are not available in every language and can-

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 31–40
Alacant, Spain, May 2018.

not provide sufficiently compact vocabularies for the large training sets typically used in NMT.

In order to increase the accuracy in translating rare and unseen words with NMT, in this paper, we propose to learn information about the *source language* morphology directly from the bilingual lexical context and use this information to compose word representations from a minimal set of input symbols. In addition to improving the quality of input word representations, our approach also aims at eliminating the necessity of using a separate and sub-optimal word segmentation step on the source language. The approach of learning word embeddings compositionally has recently been applied in language modeling and has found to be promising (Vania and Lopez, 2017). In this study, which extends (Ataman and Federico, 2018b)[1], we present and evaluate an approach for improving the source language input representations in NMT by augmenting the *embedding layer* with a *bi-directional recurrent neural network* (bi-RNN), which can learn compositional input word representations from embeddings of character n-grams. We compare our approach against conventional embedding-based representations of sub-word units learned from statistical word segmentation methods in official evaluation benchmarks, under low to medium resource conditions, by pairing English with four languages: Czech, German, Italian and Turkish, where each language represents a distinct morphological typology. The experimental findings show that our compositional input representations provide significantly and consistently better translation quality for rare and unknown words than the prominent sub-word embedding based NMT approaches in all language directions.

2 Neural Machine Translation

The NMT model we use in this paper (Sutskever et al., 2014) is based on the idea of predicting the conditional probability of translating a source sentence $x = (x_1, x_2, \dots x_m)$ of length m, into a target sentence $y = (y_1, y_2, \dots y_j \dots y_l)$ of length l, using the decomposition

$$p(y|x) = \prod_{j=1}^{l} p(y_j | y_{j-1}, .., y_0, x_m, .., x_1) \quad (1)$$

The model is trained by maximizing the log-likelihood of a training dataset consisting of parallel sentence pairs in two languages using stochastic gradient descent methods (Bottou, 2010) and the backpropagation through time (Werbos, 1990) algorithm .

The inputs of the model are one-hot vectors, which have a single bit set to 1 to identify a given word in the vocabulary. Each word vector is mapped to an embedding, a continuous representation of the word in a lower-dimensional but more dense space. Then, the *encoder*, a stacked bi-RNN, learns a distributed representation of the source sentence x in the form of m dense vectors corresponding to its hidden states. The output states of a stacked RNN encoder with L layers is computed using the following equations:

$$h_i^k = RNN(h_i^{k-1}, h_{i-1}^k) \quad (2)$$

where h_i^0 is the embedding of the input word i ($l = 1..L$ and $i = 1..m$). The output of the encoder is fed to the *decoder*, a unidirectional stacked RNN, in order to predict the target sentence y word by word. Each target word y_j is predicted by sampling from a word distribution computed from the previous target word y_{j-1}, the previous hidden state of the decoder, and the *source-context vector*, which is a linear combination of the encoder hidden states. The weights of each hidden state are dynamically computed by the *attention* model (Luong et al., 2015) on the basis of the current decoder hidden state h_t and the corresponding encoder hidden states $\bar{h}_s$. During the generation of each target word y_j, its probability is normalized via a softmax function.

The number of parameters used by the model are mainly defined by the sizes of the source and target vocabularies, which requires to use fixed-size vocabularies in order to control the computational complexity. However, this limitation creates an important bottleneck when translating from and to low-resource and morphologically-rich languages, due to the sparseness of the lexical distribution.

3 Related Work

In order to improve the translation accuracy of rare words in NMT, previous studies have pro-

[1] This paper extends (Ataman and Federico, 2018b) in four ways: with a new and more efficient implementation of the model, with experiments with deeper and wider NMT networks, with results on new translation directions and under significantly larger training data conditions, and by reporting results on sentences containing rare words.

posed several approaches which share the representations of word pieces among different words. These approaches include either engineering new NMT models that efficiently work at the character level, or performing a pre-processing step where words are segmented into smaller units using supervised or statistical tools before computing the NMT vocabulary.

3.1 Character-level NMT

The first set of statistical approaches that attempted to overcome the fixed-size vocabulary problem in NMT is based on the idea of constructing the translation model directly at the level of characters. Most of these approaches are based on the character-level language model of Kim et al. (2016), which uses convolutional and highway networks for transforming character embeddings into feature representations of sentence segments. Costa-Jussa and Fonollosa (2016) applied this approach to NMT for learning the source language input representations with a convolutional neural network while still maintaining the translation model as the same bi-RNN based encoder-decoder network (2016). Lee et al. (2016) further extended this approach to achieve fully character-level NMT by changing the decoder with a character-based one (Chung et al., 2016). Another approach that also implements fully character-level NMT based on convolutional neural networks is ByteNet (Kalchbrenner et al., 2016), which performs translation in linear time steps with respect to the source sentence length.

The main problem with these approaches is that they generally disregard lexical boundaries while learning distributed representations of the input units. Nevertheless, it is controversial whether semantics, and therefore morphology, can be modeled without maintaining a context defined at the lexical level. An additional drawback related to these methods resides in the increased sequence lengths caused by processing the sentences as sequences of characters, which also augments the computational cost despite the reduced complexity in the softmax layer. Moreover, using solely convolution cannot capture information about the relative position of each interior unit inside the word, which could provide important cues about their morphological roles. An earlier approach to character-level NMT was developed by Ling et al. (2015), which instead learns compositional input representations of words using two additional layers of bi-LSTMs in the source and target sides of the NMT model. The decoding is implemented using a softmax over the character vocabulary in the target language. Although this approach allows to maintain NMT at the lexical level, the overall computational complexity of the resulting model becomes too high to be deployed in practical tasks.

3.2 Unsupervised Word Segmentation

A more straight-forward and faster method to cope with the high computational complexity in NMT is to apply a statistical word segmentation method as a data pre-processing step before training the model. This step reduces the size of the corpus vocabulary to a maximum number of sub-word units. Although the original NMT model was designed to translate sequences of words, it is now common to perform NMT at the sub-lexical level based on input representations learned from a vocabulary of sub-word units. Indeed, learning embeddings of sub-word units which are more frequently observed in different lexical contexts allows to reduce the data sparseness and improve the quality of input representations (Ataman and Federico, 2018a). In this paper, we discuss two of such approaches: Byte-Pair Encoding (BPE) (Sennrich et al., 2016) and Linguistically-Motivated Vocabulary Reduction (LMVR) (Ataman et al., 2017).

Byte-Pair Encoding is originally a data compression algorithm which aims to minimize the length of a sequence of bytes by finding the most frequent consecutive byte pairs and encoding them using the unused byte values (Gage, 1994). This algorithm was adapted to NMT by Sennrich et al. (2016) for achieving open vocabulary translation. In the modified algorithm, the most frequent character sequences are iteratively merged for a predetermined number of times in order to generate a fixed-size vocabulary of sub-word units. This purely statistical method is based on the hypothesis that many types of words can be translated when segmented into smaller units, such as named entities and loanwords. However, by solely relying on corpus frequency, one cannot provide a sufficiently compact vocabulary that can generalize among the inflected surface forms commonly observed in morphologically-rich languages (Ataman et al., 2017; Huck et al., 2017; Tamchyna et al., 2017). Moreover, many studies have showed that splitting words into sub-word units at positions that disregard the morpheme boundaries can lead to semantically ambiguous sub-word units, and consequently, inaccurate translations (Niehues et al., 2016; Ataman et al., 2017; Pinnis et al., 2017).

Linguistically-Motivated Vocabulary Reduction also constitutes a pre-processing step to NMT where an unsupervised morphology learning algorithm learns the optimal way of segmenting words into morphs and later uses the lexicon of morphs to build a sub-word vocabulary for the translation engine. The method is an extension of *Morfessor FlatCat* (Grönroos et al., 2014), where a Hidden Markov Model (HMM) models the composition of a word based on the transitions between different morphs and their morphological categories (*i.e.* prefix, stem or suffix). The category-based HMM is essential for a linguistically motivated segmentation, as words are only split considering the possible categories of the morphs and not at positions which may break the morphological structure or generate semantically ambiguous sub-word units. Ataman et al. (2017) have modified this method in order to optimize the morphology model with a constraint on the output vocabulary size, allowing it to be adopted as a vocabulary reduction method for NMT. By manipulating regularities in morphological transformations of the concatenating nature, LMVR aids to improve the NMT of languages with agglutinative or templatic morphology. However, it does not yield significant improvements in fusional languages where the boundaries of morphemes inside the words are not transparent (Ataman and Federico, 2018a).

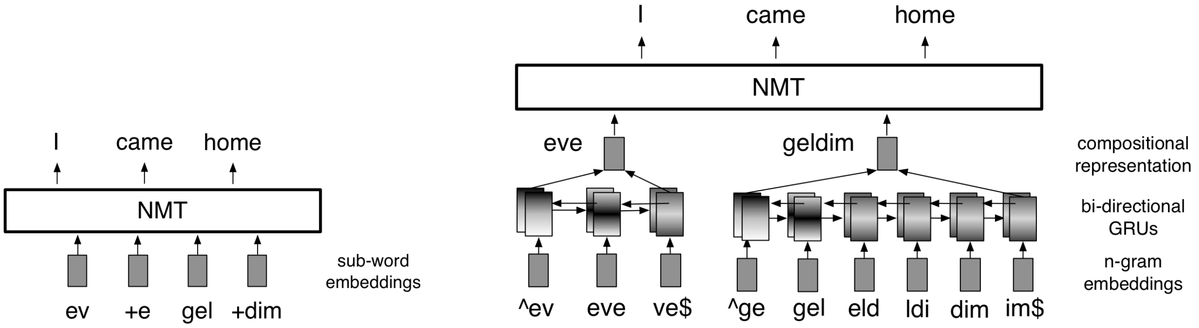
3.3 Morphological Analysis

In contrast to statistical approaches, few studies have opted to use supervised morphological analysis tools in order to reduce data sparseness in NMT. For instance, Sanchez and Toral (2016) have used a supervised morphological segmentation tool for English–Finnish NMT in order to separate words into root and inflection boundaries, whereas Huck and colleagues (2017) suggested to perform NMT based on a vocabulary of morphological features predicted by a morphological analyzer. While such methods aid in predicting a more compact NMT vocabulary in terms of root and affixes, they cannot reduce the vocabulary of a given text to fit any vocabulary size, which obliges one to further reduce the vocabulary using an unsupervised word segmentation method. Moreover, morphological analyzers are language-specific tools and as such they cannot provide general solutions to machine translation.

4 Learning Compositional Input Representations via bi-RNNs

One drawback of using statistical word segmentation methods for vocabulary prediction in NMT is that these methods constitute a pre-processing step to NMT, and hence they are not optimized for the translation task. Moreover, as given in Figure 1a, transforming sentences into sequences of sub-words leads to distributing the probability of a source word among multiple tokens, thus, increases the complexity of the alignment task performed by the attention model. In order to improve the accuracy in translating rare words in NMT, instead, we propose to perform NMT using word representations learned compositionally from smaller orthographic symbols inside the words, such as character n-grams, that can easily fit in the model vocabulary. This composition is essentially a function which can establish a mapping between combinations of orthographic units and lexical meaning, that is learned using the bilingual context, so that it can produce representations that are optimized for machine translation.

In our model (Figure 1b), the one-hot vectors retrieve the corresponding source embeddings for every word and feed them to an additional *composition layer*, which computes the final representations that are input to the encoder. For learning the mapping between the sublexical units and the lexical context, we employ a bi-RNN. Hence, by encoding the context of each interior unit inside the word, we believe that the network be able to capture important cues about their functional role, *i.e.* semantic or syntactic contribution to the word meaning. We implement the network using GRUs (Cho et al., 2014), which have shown comparable performance to LSTM units (Hochreiter and Schmidhuber, 1997) while performing faster computation. As a minimal set of input symbols required to cope with contextual ambiguities, and at the same time optimize the size of the NMT vocabulary, we opt to use intersecting sequences of character trigrams, as recently suggested by Vania and Lopez (2017). Our preliminary experiments (Ataman and Federico, 2018b) also confirmed the stand-alone sufficiency of character tri-



(a) Input: sub-word embeddings.

(b) Input: word representations built from char trigrams.

Figure 1: NMT of the Turkish sentence *Eve geldim* (*I came home*) using different input representations.

grams as fundamental units in the compositional NMT model.

Given a bi-RNN with a forward (f) and backward (b) layer, the input representation $\mathbf{w}$ of a token of t characters is computed from the hidden states $\mathbf{h}_t^f$ and $\mathbf{h}_b^0$, *i.e.* the final outputs of the forward and backward RNNs, as follows:

$$\mathbf{w} = \mathbf{W}_f \mathbf{h}_f^t + \mathbf{W}_b \mathbf{h}_b^0 + \mathbf{b} \tag{3}$$

where $\mathbf{W}_f$ and $\mathbf{W}_b$ are weight matrices and $\mathbf{b}$ is a bias vector (Ling et al., 2015). These parameters are jointly learned together with the internal parameters of the GRUs and the input token embedding matrix to minimize the cost of the overall network while training the NMT model. For an input of m tokens, the computational complexity of the network is increased by $O(Kt_{\max}m)$, where K is the average cost of one bi-RNN layer and $t_{\max}$ is the maximum number of symbols per word.

5 Experiments

In order to evaluate our approach in NMT, we set up an evaluation benchmark which models NMT from four languages: Czech (*CS*), German (*DE*), Italian (*IT*) and Turkish (*TR*) into English (*EN*), where each input language represents a different lexical distribution reflected by its morphological characteristics, simulating conditions ranging from the low-resource and high sparseness (Turkish) to the high-resource and low sparseness (Italian) cases.

For training the Czech–English and German–English NMT models, we use the available data sets from the WMT[2] (Bojar et al., 2017) shared task on machine translation of news, which consist of Europarl (Koehn, 2005), Commoncrawl and News Commentary (Tiedemann, 2009). For achieving a comparable size of training data, we reduce the training set in German–English using the Invitation Model (Cuong and Simaan, 2014). We evaluate these models on the official test sets from 2016. Due to the lack of sufficient amount of news domain data, for the Italian–English and Turkish–English directions, we build generic NMT systems using data collected from TED Talks (Cettolo et al., 2012), EU Bookshop (Skadins et al., 2014), Global Voices, Gnome, Tatoeba, Ubuntu (Tiedemann, 2012), KDE4 (Tiedemann, 2009), Open Subtitles (Lison and Tiedemann, 2016) and SETIMES (Tyers and Alperen, 2010), and reduce the size of the training data for having comparable numbers of tokens (Italian) and types (Turkish) with the other languages. These models are evaluated on the official test sets from the evaluation campaign of IWSLT[3] (Cettolo et al., 2017). The morphological characteristics of the languages used in our study are presented in Table 1, while the statistics of the data sets used in our experiments can be seen in Tables 2 and 3.

We perform NMT by keeping the segmentation

Language	Morphological Typology	Morphological Complexity
Italian	*Fusional*	*Low*
German	*Fusional*	*Medium*
Czech	*Fusional, Agglutinative*	*High*
Turkish	*Agglutinative*	*High*

Table 1: The evaluated languages in our study along with their morphological characteristics.

[2] The First Conference on Machine Translation

[3] The International Workshop on Spoken Language Translation with shared tasks organized between 2003-2017.

Language	# sentences (K)	# tokens (M)	# types (K)
IT-EN	785	21(IT) - 22(EN)	152(IT) - 106(EN)
DE-EN	992	19(DE) - 18(EN)	501(DE) - 261(EN)
CS-EN	965	22(CS) - 25(EN)	385(CS) - 204(EN)
TR-EN	434	6(TR) - 8(EN)	373(TR) - 135(EN)

Table 2: Training sets. (*M*: Million, *K*: Thousand.)

Language	Data sets		# sentences (K)	# tokens (K)
IT-EN	Dev	dev2010 & test2010	3,5	74(IT) - 79(EN)
	Test	test2011 & test2012	3,2	55(IT) - 60(EN)
DE-EN	Dev	test2015	2,2	44(DE) - 46(EN)
	Test	test2016	3,0	62(DE) - 65(EN)
CS-EN	Dev	test2015	2,7	46(CS) - 54(EN)
	Test	test2016	3,0	57(CS) - 65(EN)
TR-EN	Dev	dev2010 & test2010	2,4	34(TR) - 47(EN)
	Test	test2011 & test2012	2,7	39(TR) - 53(EN)

Table 3: Development and Testing Sets. All data set are official evaluation sets from WMT (Czech and German) and IWSLT (Italian and Turkish). (*M*: Million, *K*: Thousand.)

on the English side constant and applying different open vocabulary NMT approaches to the input languages. We segment the English side with LMVR as it provides a segmentation that is more consistent with the morpheme boundaries (Ataman and Federico, 2018b).

The compositional bi-RNN is implemented in PyTorch (Paszke et al., 2017) and integrated into the OpenNMT-py toolkit (Klein et al., 2017). The *simple* NMT model constitutes the baseline in our study and performs translation directly at the level of sub-word units, using a two-layer encoder based on Stacked GRUs, a two-layer GRU decoder, input feeding and the general global attention mechanism (Luong et al., 2015). For segmenting the words in the source side, we chose to use BPE for the fusional languages (Czech, German and Italian), whereas in Turkish we use LMVR, as suggested in (Ataman and Federico, 2018a). The *compositional* model, on the other hand, performs NMT with input representations composed from a vocabulary of character trigrams. All the models use an embedding and GRUs with size 512. In order to achieve a fair comparison, we use a one-layer encoder for the compositional model, which allows the two models to have comparable number of parameters, whereas we use the same settings for the remaining network properties and hyper-parameters. All models are trained using the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.0002 and default values for the other hyper-parameters. We clip the gradient norm at 1.0 (Pascanu et al., 2013) and set the dropout at 0.1 after hyper-parameter tuning. All models are trained with a model vocabulary of 30,000 units. The compositional model uses a trigram vocabulary of the same size whereas the segmentation methods (BPE and LMVR) are trained to fit in this exact vocabulary limit. We evaluate the accuracy of each model output using the (case-sensitive) BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and chrF (Popovic, 2015) metrics. Significance tests are computed only for BLEU with Multeval (Clark et al., 2011).

6 Results and Discussion

The performance of NMT models in translating each language using different types of encoder input representations can be seen in Table 4. The results show that the compositional model achieves the best translation accuracy in translation of all morphologically-rich languages. The overall improvements obtained with this model over the best performing simple model are **0.77** BLEU points in German, **0.74** BLEU points in Czech and **0.11** BLEU points in Turkish to English translation directions. The improvements are more evident for Turkish in terms of other evaluation metrics, where the compositional model improves the translation accuracy by **0.016** TER and **0.009** chrF points. In

Language Direction	Model	BLEU	TER	chrF
IT-EN	Simple (BPE)	**29.02**	**0.501**	**0.5328**
	Compositional	28.66	0.506	0.5293
DE-EN	Simple (BPE)	20.46	0.591	**0.4544**
	Compositional	**21.23**	**0.585**	0.4537
CS-EN	Simple (BPE)	19.59	0.615	0.4724
	Compositional	**20.33**	**0.614**	**0.4780**
TR-EN	Simple (LMVR)	23.02	0.585	0.4613
	Compositional	**23.13**	**0.569**	**0.4703**

Table 4: Experiment Results. Best scores for each translation direction are in bold font. All improvements over the baseline are statistically significant (p-value < 0.01).

Italian to English translation direction, the performance of the simple model is higher than the compositional model by **0.36** BLEU, **0.005** TER and **0.0035** chrF points.

The better performance of the compositional model in translating German, Czech and Turkish suggests that our approach is beneficial in eliminating the morphological errors caused by segmentation in languages with different morphological typologies. The improvements are highest for Czech and German, both of which have a fusional morphology of medium to high complexity, and the source language vocabulary of the training data ranges from around 400,000 to 500,000 types of words, indicating a high level of lexical sparseness. At a comparable vocabulary size, the improvements are generally lower in Turkish to English translation direction, where the input language has an agglutinative morphology with a much higher level of data sparseness. This might be due to the efficient performance of LMVR in generating morphologically-consistent sub-word units in the low-resource setting of agglutinative languages. Nevertheless, the results suggest that our compositional model can learn a higher level of morphological knowledge than LMVR, which was previously found to provide comparable performance to morphological analyzers in Turkish–English NMT using the embedding-based input representations (Ataman et al., 2017). Moreover, it can also generalize over different types of morphology in both low and high resource settings.

In the Italian to English translation direction, despite the comparable size of training data with Czech and German in the high-resource setting, the source word vocabulary is around 150,000 words, which represents the low level of sparseness. The higher overall performance of the NMT model which uses BPE for vocabulary reduction compared to the compositional model suggests that the embedding based sub-word representations are sufficient in reducing this vocabulary to fit into a space of 30,000 units. Nevertheless, in order to observe the actual accuracy in translating rare words, we carry out a focused analysis where we sample from the test sets only the sentences that contain singletons (*i.e.* words that are observed once in the training corpus) in the source side and evaluate the translation accuracy obtained with each NMT model on these sentences. This sampling results in 190 sentences in Italian, 470 sentences in Turkish, 562 sentences in German and 611 sentences in Czech to English directions. The results of this analysis, which can be found in Table 5, show that the compositional model translates sentences containing rare words more accurately than the simple model in all languages, with improvements ranging from **0.53** to **2.72** BLEU points. The improvement obtained also in the Italian to English translation direction shows that although in overall sub-word segmentation achieves higher output accuracy, it is still not as efficient as our approach in translating the small portion of rare words in the Italian corpus.

We extend our analysis in order to also evaluate the performance of different approaches in translating out-of-vocabulary (OOV) words. Similarly, we sample from the test sets only the sentences which contain OOVs, resulting in relatively larger test sets of 443 Italian, 1096 Turkish, 1396 Czech and 1449 German sentences. The evaluation of each NMT model on these sets, results of which are also given in Table 5, show that the compositional model again outperforms the simple NMT

Language Direction	Model	BLEU (Singletons)	BLEU (OOVs)
IT-EN	Simple (BPE)	23.54	23.23
	Compositional	**24.07**	**24.98**
DE-EN	Simple (BPE)	14.19	14.30
	Compositional	**16.91**	**16.76**
CS-EN	Simple (BPE)	16.33	16.83
	Compositional	**16.60**	**17.73**
TR-EN	Simple (LMVR)	19.69	20.31
	Compositional	**20.91**	**21.50**

Table 5: Translation accuracy of NMT models evaluated only on sentences containing singletons and OOVs. Best scores for each translation direction are in bold font. All improvements over the baseline are statistically significant (p-value < 0.01).

model in all languages, where the improvements range from **0.90** to **2.46** BLEU points. These findings suggest that our compositional NMT approach provides a higher generalization capability compared to conventional approaches to open vocabulary NMT.

7 Conclusion

In this paper, we have addressed the problem of translating rare words in NMT and proposed to solve it by replacing the conventional sub-word embeddings with input representations compositionally learned from character n-grams using a bi-RNN. Our approach showed significant and consistent improvements over a variety of languages with different morphological typologies, making it a competitive approach for NMT of low-resource and morphologically-rich languages. In the future, we plan to extend our approach in order to improve also the target side representations used by the NMT decoder and to evaluate it under similar morphological and data sparseness conditions on the target side. Finally, our benchmark and implementation are available for public use.

Acknowledgments

The authors would like to thank NVIDIA for their donation of Titan Xp GPUS that allowed to conduct this research.

References

Ataman, Duygu, Matteo Negri, Marco Turchi and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics* 108.1 (2017): 331-342.

Ataman, Duygu and Marcello Federico. 2018a. An Evaluation of Two Vocabulary Reduction Methods for Neural Machine Translation. *Proceedings of the The 13th Conference of The Association for Machine Translation in the Americas*, Boston, USA. 97-110.

Ataman, Duygu and Marcello Federico. 2018b. Compositional Representation of Morphologically-Rich Input for Neural Machine Translation arXiv preprint arXiv:2251036.

Barone, Antonio Valerio Miceli, Jindrich Helcl, Rico Sennrich, Barry Haddow and Alexandra Birch. 2017. Deep architectures for Neural Machine Translation. *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. 99–107.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, USA. 257–267.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann and Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, and others. 2017. Findings of the 2017 Conference on Machine Translation (WMT) *Proceedings of the Second Conference on Machine Translation.* , Copenhagen, Denmark. 169–214.

Bottou, Léon 2010. Large-Scale Machine Learning with Stochastic Gradient Descent *Proceedings of 19th International Conference on Computational Statistics (COMPSTAT)*, Paris, France. Springer. 177–186.

Cettolo, Mauro, Christian Girardi and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, Trento, Italy.

Cettolo, Mauro, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuitho Sudoh, Koichiro Yoshino, Christian Federmann 2017 Overview of the IWSLT 2017 Evaluation Campaign. *International Workshop on Spoken Language Translation* 2–14.

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches *Syntax, Semantics and Structure in Statistical Translation (2014): 103.*

Chung, Junyoung, Kyunghyun Cho and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany. (Volume 1: Long Papers). 1693–1703.

Clark, Jonathan H., Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL).* 176–181.

Costa-jussà, Marta R. and José A. R. Fonollosa. 2016. Character-based Neural Machine Translation *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).* 357–361.

Cuong, Hoang and Khalil Simaan. 2014. Latent domain translation models in mix-of-domains haystack *Proceedings of Proceedings of the 25th International Conference on Computational Linguistics (COLING).* 1928–1939.

Gage, Philip 1994. A New Algorithm for Data Compression *The C Users Journal.* 12(2):23–38.

Grönroos, Stig-Arne, Sami Virpioj, Peter Smit and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology. *Proceedings of the 25th International Conference on Computational Linguistics (COLING).* 1177–1185.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory *Neural computation.* MIT Press 1735–1780.

Huck, Matthias, Simon Riess and Alexander Fraser. 2017. Target-Side Word Segmentation Strategies for Neural Machine Translation *Proceedings of the 2nd Conference on Machine Translation (WMT)*, Copenhagen, Denmark. 56–67.

Junczys-Dowmunt, Marcin, Tomasz Dwojak and Hieu Hoang. 2016. Is neural machine translation ready for deployment? A case study on 30 translation directions *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*

Kalchbrenner, Nal, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves and Koray Kavukcuoglu. 2016 Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099.*

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, San Diego, USA.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart and Alexander M. Rush. 2017. *OpenNMT: Open-Source Toolkit for Neural Machine Translation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, System Demonstrations, 67-72.

Kim, Yoon, et al. 2016. Character-Aware Neural Language Models. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, USA. 2741-2749.

Koehn, Philipp 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit)* 79–86.

Lee, Jason, Kyunghyun Cho and Thomas Hofmann. 2015. *Fully Character-Level Neural Machine Translation without Explicit Segmentation.* Transactions of the Association for Computational Linguistics (TACL). 5: 365–378

Ling, Wang, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W. Black and Isabel Trancoso. 2015. *Finding function in form: Compositional character models for open vocabulary word representation.* Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 1520–1530.

Ling, Wang, Isabel Trancoso, Chris Dyer and Alan W. Black. 2015. *Character-based neural machine translation.* arXiv preprint arXiv:1511.04586.

Lison, Pierre and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. *Proceedings of the 10th International Conference on Language Resources and Evaluation.* 923–929.

Luong, Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* 1412–1421.

Luong, Thang, and Christopher D. Manning. 2016. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).* 1054–1063.

Niehues, Jan, Eunah Cho, Thanh-Le Ha and Alex Waibel. 2016. *Pre-Translation for Neural Machine Translation.* Proceedings of The 26th International Conference on Computational Linguistics (COLING). 1828–1836.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).* 311–318.

Pascanu, Razvan, Tomas Mikolov and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, USA. 1310–1318.

Pascanu, Razvan, Caglar Gulcehre, Kyunghyun Cho and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga and Adam Lerer. 2017. Automatic differentiation in PyTorch. *NIPS 2017 Autodiff Workshop*, Long Beach, USA.

Pinnis, Mārcis, Rihards Krišlauks, Daiga Deksne and Toms Miks. 2017. Neural Machine Translation for Morphologically Rich Languages with Improved Subword Units and Synthetic Data *Proceedings of the International Conference on Text, Speech, and Dialogue (TSD)* 237–245.

Popovic, Maja. 2015. chrF: Character n-gram F-score for Automatic MT Evaluation. *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT)*, Lisbon, Portugal. 392–395.

Sánchez-Cartagena, Vıctor M. and Antonio Toral. 2016. Abu-MaTran at WMT 2016 Translation Task: Deep Learning, Morphological Segmentation and Tuning on Character Sequences *Proceedings of the First Conference on Machine Translation*, Berlin, Germany. 362-370.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany. 1715–1725.

Snover, Matthew and Dorr, Bonnie and Schwartz, Richard and Micciulla, Linnea and Makhoul, John 2006. A study of translation edit rate with targeted human annotation. *Proceedings of association for machine translation in the Americas*. Vol. 200. No. 6. 223–231.

Skadiņš, Raivis, Jörg Tiedemann, Roberts Rozis and Daiga Deksne. 2014. Billions of Parallel Words for Free: Building and Using the EU Bookshop Corpus. *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland. 1850–1855.

Sutskever, Ilya, Oriol Vinyals and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems.* 3104–3112.

Tamchyna, Aleš, Marion Weller-Di Marco and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation *Proceedings of the 2nd Conference on Machine Translation (WMT)*, Copenhagen, Denmark. 32–42.

Tiedemann, Jörg 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. *In Recent Advances in Natural Language Processing* Amsterdam, Philadelphia. Vol. 5. 237–248.

Tiedemann, Jörg 2012. Parallel Data, Tools and Interfaces in OPUS. *Proceedings of the eighth international conference on Language Resources and Evaluation* Vol. 2012. 2214–2218.

Tyers, Francis M. and Murat Serdar Alperen. 2010. South-east European Eimes: A parallel corpus of balkan languages. *Proceedings of the LREC Workshop on Exploitation of Multilingual Resources and Tools for Central and (South-) Eastern European Languages* 49–53.

Vania, Clara and Adam Lopez. 2009. From Characters to Words to in Between: Do We Capture Morphology? *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)* 2016–2027.

Werbos, Paul J 1990. Backpropagation Through Time: What it does and How to do it *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)* 78:1550–1560.

Wu, Yonghui and Schuster, Mike and Chen, Zhifeng and Le, Quoc V and Norouzi, Mohammad and Macherey, Wolfgang and Krikun, Maxim and Cao, Yuan and Gao, Qin and Macherey, Klaus and others 2016. Googles Neural Machine Translation System: Bridging the Gap between Human and Machine Translation *arXiv preprint arXiv:1609.08144*

Development and evaluation of phonological models for cognate identification

Bogdan Babych
Centre for Translation Studies
University of Leeds, UK
b.babych@leeds.ac.uk

Abstract

The paper presents a methodology for the development and task-based evaluation of phonological models, which improve the accuracy of cognate terminology identification, but may potentially be used for other applications, such as transliteration or improving character-based NMT. Terminology translation remains a bottleneck for MT, especially for under-resourced languages and domains, and automated identification of cognate terms addresses this problem. The proposed phonological models explicitly represent distinctive phonological features for each character, such as acoustic types (e.g., vowel/ consonant, voiced/ unvoiced/ sonant), place and manner of articulation (closed/open, front/back vowel; plosive, fricative, or labial, dental, glottal consonant). The advantage of such representations is that they explicate information about characters' internal structure rather than treat them as elementary atomic units of comparison, placing graphemes into a feature space that provides additional information about their articulatory (pronunciation-based) or acoustic (sound-based) distances and similarity. The article presents experimental results of using the proposed phonological models for extracting cognate terminology with the phonologically aware Levenshtein edit distance, which for Top-1 cognate ranking metric outperforms the baseline character-based Levenshtein by 16.5%. Project resources are released on:
https://github.com/bogdanbabych/cognates-phonology

1 Introduction: development of phonological models for cognate terminology identification

This paper presents a methodology for the development and automated evaluation of linguistic phonological features sets that can extend traditional methods of cognate terminology identification, such as Levenshtein edit distance.

Cognate identification is important for a range of applications. This paper evaluates its use for assisting MT developers in creating cognate term banks used in rule-based and hybrid MT, as well as in computer-assisted translation, development of dictionaries and between closely related languages (e.g., Ukrainian (Uk) and Russian (Ru), Portuguese (Pt) and Spanish (Es), Dutch (Nl) and German (De)). For many of such language pairs one of the languages can be under-resourced, therefore no electronic dictionaries are available, and only small parallel corpora with limited lexical coverage can be collected. Typically these parallel corpora can provide translations for frequently used general words, but miss the 'long tail' of less frequent, often topic-specific or terminological words. However, in closely related languages these words are often cognates, which creates a possibility to rapidly extend bilingual lexicons in semi-automated way using non-parallel, comparable corpora and automated cognate identification techniques. In this task, cognate candidates are generated from word lists created from large monolingual comparable corpora in both languages The assumption is that the developers have good linguistic intuition of both languages and work through lists of cognate candidates, checking which pairs can be added to the bilingual dictionary. Their productivity depends on whether cognates are presented high up in the list of candidates, ideally at the top of the list, or

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 41–48
Alacant, Spain, May 2018.

at least in the top N items, where N should be relatively small, e.g., the number of lines which fit on a single screen.

Other uses of cognates for terminology identification include term extraction from parallel corpora. If multiword source terms are known, the task is to identify the boundaries of the corresponding multiword target terms in the aligned target sentences, where component words or stems of compound words within the target terms may not be necessarily cognate with the corresponding source, so correctly identified cognates can facilitate adding adjacent non-cognate words according to part-of-speech and word order patterns, e.g., En: *'**information** requirements'* ~ Uk: *'**інформаційні** потреби' ('**informatsijni** potreby')*; or splitting and extending compounds which have cognate parts, e.g., En: 'multi**national**' ~ Uk: 'багато**національний**'- ('bahato**natsionalnyj**').

Yet another application of cognate identification is sentence alignment of parallel corpora, where statistical alignment methods are more accurate if cognates are used as an additional data source (Lamraoui and Langlais, 2013:2). Inaccuracies in cognate identification, which are due to orthographic differences, often create unnecessary bottlenecks for this task (Varga et al., 2015: 249). In this scenario identified cognates are not necessarily terms, but they contribute to a more accurate alignment and extraction of non-cognate terminology, produced from word alignment and monolingual terminology detection.

An additional complication for the multilingual terminology extraction scenarios that rely on cognate identification is the use of different writing systems in the source and target (e.g., Cyrillic or Georgian vs. Latin script), which requires transliteration between those languages.

Transliteration is often non-trivial, because of differences in pronounciation of the same letters, the lack of direct graphemic equivalents across languages, contextual dependencies in transliteration rules, different historical conventions for different words (e.g., En/De "h" → Ru "х" (*hockey ~ хоккей,* since borrowed directly from En), or "г" (*hermeneutics ~ герменевтика*, since borrowed via Ukrainian, where En: h → Uk: г [ɣ] → Ru: г [g]). Also, even if languages use the same alphabet, pronounciation of letters and corresponding transliteration rules may differ (e.g., Cyrillic letter "и" = [i] in Ru and [y] in Uk, Latin letter "g" = [g] in En/De, and [ɣ] in Nl), so new transliteration mappings need to be created for each translation direction, each with their potential language-specific problems.

As a result, the complexity of transliteration in some cases is comparable to the complexity of MT, and it is often addressed not via simple character mappings, but via fully developed character-based MT models that require an aligned training corpus for each translation direction, and which are used in MT applications to cover out-of-vocabulary words, such as compounds, morphologically complex words, named entities and cognate terminology (Senrich et al., 2016: 1716)

Transliteration problem resembles a traditional "direct translation" bottleneck in MT: this approach cannot reuse any of the previously created mappings between languages if a new language pair or translation direction need to be covered. A more principled approach to the transliteration problem in the context of automated cognate identification, developed in this paper, is mapping characters for each language into a language-independent ("interlingual") phonological feature space.

2 Related work

The use of phonological features for cognate identification has been initially proposed in the context of dialectological studies (Nerbonne & Heeringa, 1997) and diachronic phonology (Kondrak, 2000: 288), (Kondrak, 2009). Some limitations of these approaches for MT-related tasks have been discussed in (Babych, 2016), such as the need for phonological transcription of orthographic words and the absence of reliable evaluation for different ways of organising the complex phonological feature space and computing similarity between phonological segments. For instance (Kondrak, 2000: 290-293) acknowledges that different phonological features make unequal contribution in computing similarity between segments. To address this problem, in the ALINE phonetic aligner an introspective set of weights for each of the features is adopted from (Ladefoged, 1995). Machine-learning algorithms based on learning phonetic mappings from bilingual texts (Kondrak, and Sherif, 2006) outperform the introspective linguistic model based on weighted phonological features.

However, the most important difference between identification of cognates for dialectological or historical studies of language vs. for MT-oriented tasks of cognate term identification is the range of the compared candidate cognates and therefore the need of the metric to be optimised for both recall and precision on the large dictionary data sets. Addition of phonological features on such tasks often results in overgeneration, so additional features have to be used, such as semantic similarity of terms, WordNet-based and semantic features, clustering (Kondrak, 2009, St Arnaud et al., 2017).

On the large scale for cognate identification for MT, where datasets are not limited only to candidate cognate pairs, a character-based Levenshtein edit distance (Levenshtein, 1966) is typically used, without additional linguistic features. Levenshtein metric calculates the number of insertions, deletions and substitutions between compared word pairs from different languages and determines if they pass a threshold to be considered cognate candidates. For example, if cognate candidates are extracted from a non-aligned or non-parallel corpus, the Levenshtein distance is computed for every pair of words in the two word lists created for each language (the Cartesian product of the lists), the search space may be restricted to comparing words with the same part-of-speech (PoS) codes, if PoS annotation is available for the corpus.

However the problem with the character-based Levenshtein metric is that all characters in comparison are treated as atomic units that do not have any internal structure and therefore, can be substituted only as a whole character. Because of this the Levenshtein metric does not distinguish between the substitutions of characters that correspond to acoustically/articulatory similar sounds vs. the substitution of phonologically distant letters. As a result, words that are intuitively close may receive a large distance score, e.g.,

Uk “*жовтий*” (*zhovtyj*)=‘yellow’
Ru “*жёлтый*” (*zheltyj*) = ‘yellow’
(Lev distance = 3),

where, for historical reasons, articulatory similar sounds are represented by different characters: the sound [o] – by ‘o’ in Uk and ‘ё’ in Ru, the sound [y] – by ‘и’ in Uk and ‘ы’ in Ru. On the other hand, words that are not cognates and are phonologically and intuitively far apart, still receive the same distance scores, such as:

Uk “*жовтий*” (*zhovtyj*) = ‘yellow’ and
Ru “*жуткий*” (*zhutkij*) = ‘dismal’ (Lev = 3).

For example, here no distinction is made between, on the one hand, the substitution “*о*” (o) → “*ё*” (^{i}o) of phonologically similar sounds (which differ only in a peripheral feature – triggering palatalization of the preceding consonant (Uk: -- Ru: +; in addition, this feature is neutralised after the sibilant “*ж*” (zh)), and on the other hand – the substitution “*о*” (o) → “*у*” (u), where sounds differ in core articulatory features of the place of vowel articulation (Uk: middle; Ru: close/high).

Some existing modifications and extensions of the Levenshtein metric introduce weightings for different character mapping, but these weights need to be set or empirically determined for each specific mapping: compared characters still do not have internal structure and there is no way to predict the weights in advance for any possible pair in a principled way.

This paper presents an automated task-based evaluation framework for an extension to the Levenshtein edit distance metric, which explicitly represents linguistic phonological features of compared characters, so the metric can use information about characters’ internal feature structure rather than treat them as elementary atomic units of comparison. Similar sets of distinctive features have been used for comparing transcriptions of spoken words in modeling dialectological variation and historical changes in languages (Nerbonne and Heeringa, 1997). In the proposed approach, phonological feature representations are applied to cognate identification and terminology extraction tasks, transliteration, and as well as modeling morphological variation. Previously it has been shown that there are multiple ways of identifying, representing, structurally arranging and comparing these features in a phonological feature space (Babych, 2016), so there is a need for a methodology for evaluating alternative feature configurations. The results of the previously reported pilot experiment, using a small-scale manual evaluation, indicated the need to use hierarchical phonological feature structures for consonants rather than flat feature vectors previously used in dialectological research.

Manual evaluation methods in previous pilot experiment cannot be used for systematically testing and optimising weights or alternative phonological feature representations used in the Levenshtein phonological metric.

For instance, a serious problem for the proposed phonologically aware metric has been overestimation of its insertion and deletion costs, which is mainly due to the relatively smaller average substitution cost, and no corresponding reduction in the average insertion or deletion costs. E.g., for non-cognates a replacement of a consonant with another phonologically unrelated consonant produces a substitution distance of 0.8, because one feature – "*type:consonant*" does not have to be rewritten (phonological structure of consonants in the proposed models has 5 features). If insertion and deletion costs remain =1, this leads to disproportional under-generation of cognates that contain inserted or deleted characters. Even though the need of adjusting insertion/deletion distances has been highlighted in the pilot stage, manual evaluation methods used then did not allow us to test and optimise multiple parameter settings for the phonological metric, such as a range of different insertion and deletion costs. Their values have to be determined experimentally using an automated evaluation methodology.

This paper develops an automated framework for evaluating different arrangements of phonological features and parameters using the task of cognate identification, which enables us to experimentally find optimal setup of a metric for a given task. Apart from practical applications mentioned above, this methodology creates a framework for feature engineering for phonologically aware character-based models for a wider range of machine translation and machine learning methods and tools, to design and calibrate phonological feature structures in a systematic way tuned for optimal the performance on specific tasks.

The proposed automated evaluation framework uses standard automatically computed evaluation metrics, such as number of cognates in top-N candidates and an average rank of a correct cognate in an ordered candidate list. Evaluation is performed on a larger data set of candidate cognate lists generated from large Ukrainian and Russian corpora on a high-performance computing cluster. The evaluation results show the settings where phonological Levenshtein metrics achieves best performance on the cognate identification task and allow us to rule out some unproductive modifications.

3 Phonological distinctive features and their application for cognate identification

A theory of phonological distinctive features, which was first proposed by Roman Jacobson (Jakobson and Halle, 1956: 46; Anderson, 1995: 116), associates each phoneme (an elementary segmental unit of speech that distinguishes meanings and is intentionally produced by speakers) with its unique set of values for categories, which apply to classes of sounds. For example, the phoneme [t] has the following values for its associated phonological categories:

'type': consonant
'voice': unvoiced
'maner of articulation': plosive;
'active articulation organ': front of the tongue
'passive articulation organ': alveolar

Phoneme [d] has the same set of articulatory features apart from one: it is pronounced with vocal cords vibrating, while organs and manner of articulation remain the same, so it differs only in the value of one distinctive feature,

'voice': voiced.

In historical development of languages and in morphological variation within a language the phonological changes more often apply only to values of certain distinctive features within characters, but much less often extend to the whole category-value system, e.g.: De: "*Tag*" = Nl "*dag*" ('day); De: "*machen*" = Nl "*maken*" ('make'). Therefore, for languages where the writing system is at least partially motivated by pronunciation, for certain character based models, e.g., modelling morphological variation or cognates in different languages, it would be useful to represent phonological distinctive features of characters, in order to differentiate between varying degrees of closeness for their different classes, e.g., vowels, sonants and consonants, or sounds with identical or similar articulation. Greater closeness between characters in terms of their phonological features has important linguistic and technical applications, such as modelling dialectal variation, historical change, morphological and derivational changes in words, e.g., stem alternations in inflected forms.

(1) In past research (Babych, 2016: 123) phonological distinctive features have been integrated into the Levenshtein distance metric in the following way: e.g., to substitute [t] with [d] in Nl: "tag" → De: "dag" there is a need to re-write only one feature out of 5, so the distance is 0.2 rather than 1. However, in the general case different classes of characters use different numbers of features, so substitution distance *Subst* is calculated as:

Subst = 1 – F-score,

where F-measure is the harmonic mean of Precision and Recall of the overlap between sets of their phonological features. This allows the metric to calculate the distance for characters with different numbers of features remaining symmetric.

(2) The order of matching the distinctive features was found to be important. The experiment described in Section 4 compares two different arrangements of features: as flat feature vectors and as feature hierarchies. In the hierarchies the higher level features need to be matched as a pre-condition for attempting to match lower level features. Hierarchical organization consistently achieves better performance compared to flat feature vectors. Intuitively this means that not all feature categories should be treated equally; some are more central, have higher priority, and license comparison of lower level features on the periphery of the phonological feature system.

(3) Insertion and deletion costs have been calibrated for the range between 0.2 and 1 using the proposed evaluation framework, described in this paper in Section 4. Optimal performance on cognate identification was achieved for cost of insertion = deletion = 0.8.

For the task of cognate identification, the introduction of these features distinguishes different types of character substitutions and gives more accurate prediction of the degree of closeness between compared characters and words, e.g., for the word pairs discussed above, where the baseline Levenshtein distance =3 for both (matching features, which do not need to be rewritten, are highlighted in bold):

Graphemic-Phonological (graphonological) feature Uk: “*жовтий*” (*zhovtyj*) = ‘yellow’

ж (zh) 'type:consonant', 'voice:ff-voiced', 'maner:ff-fricative', 'active:ff-fronttongue', 'passive:ff-palatal'

о (o) **'type:vowel'**, **'backness:back'**, **'height:mid'**, **'roundedness:rounded'**, 'palate:nonpalatalizing'

в (v) **'type:consonant'**, 'voice:fl-voiced', 'maner:fl-fricative', 'active:fl-labial', 'passive:fl-bilabial'

т (t) 'type:consonant', 'voice:pf-unvoiced', 'maner:pf-plosive', 'active:pf-fronttongue', 'passive:pf-alveolar'

и (y) **'type:vowel'**, 'backness:front', **'height:closemid','roundedness:unrounded'**, **'palate:nonpalatalizing'**

й (j) 'type:consonant', 'voice:xm-sonorant', 'maner:xm-approximant','active:xm-midtongue', 'passive:am-palatal'

Feature representations for corresponding characters in Ru: “*жёлтый*” (*zheltyj*) = ‘yellow’.

ё (io) **'type:vowel', 'backness:back', 'height:mid', 'roundedness:rounded'**, 'palate:palatalizing'

л (l) **'type:consonant'**, 'voice:lf-sonorant', ' maner:lf-lateral', 'active:lf-fronttongue', 'passive:lf-alveolar'

…

ы (y) **'type:vowel'**, 'backness:central', **'height:closemid','roundedness:unrounded', 'palate:nonpalatalizing'**

It can be seen from the examples above, why for the task of cognate identification it is important that character substitution in the graphonological Levenshtein metric only touches some distinctive feature in a characters’ feature sets. Such feature substitution at the sub-character level still unambiguously changes one character into another, since there is a one-to-one correspondence between a new set of phonological features and the corresponding sound or character: according to Jacobson’s distinctive features model (implemented in the proposed phonological representations), there cannot be two sounds in a language that share exactly the same set of values for their phonological categories.

If only some sub-character features are changed, the substitution cost is < 1, and normally reflects the proportion of phonological features which need to be rewritten.

Calculation of the Graphonological Levenshtein metric for Uk “*жовтий*” (*zhovtyj*) = ‘yellow’ and (Ru) “жёлтый” (zheltyi) = ‘yellow’:

0.0	1.0	2.0	3.0	4.0	5.0	6.0
1.0	**0.0**	1.0	2.0	3.0	4.0	5.0
2.0	1.0	**0.2**	1.2	2.2	3.2	4.2
3.0	2.0	1.2	**1.0**	2.0	3.0	4.0
4.0	3.0	2.2	2.0	**1.0**	2.0	3.0
5.0	4.0	3.2	3.0	2.0	**1.2**	2.2
6.0	5.0	4.2	4.0	3.0	2.2	**1.2**

cf.: Metric calculated for Uk “*жовтий*” (*zhovtyj*) = ‘yellow’ with Ru “*жуткий*” (*zhutkij*) ‘dismal’:

0.0	1.0	2.0	3.0	4.0	5.0	6.0
1.0	**0.0**	1.0	2.0	3.0	4.0	5.0
2.0	1.0	**0.2**	1.2	2.2	3.2	4.2
3.0	2.0	1.2	**1.0**	1.2	2.2	3.2
4.0	3.0	2.2	2.0	**1.8**	2.2	3.0
5.0	4.0	3.2	3.0	2.8	**2.0**	3.0
6.0	5.0	4.2	4.0	3.8	3.0	**2.0**

While the baseline Levenshtein distance Lev=2 for both pairs shown above, the phonologically-aware distance, GLev = 2.0 for non-cognates, which is > 1.2 for cognates.

An additional advantage of using of phonological feature representations for graphemes is a more natural "interlingual" transliteration between different scripts and languages. The phonological models, presented in this paper, map characters from any given language into a universal space of acoustic and articulatory phonological features, which is independent of any specific writing system or a language-pair. This space can be seen as a phonological "interlingua", which shares some advantages with the idea of interlingual MT: graphonological mappings enable implicit cross-lingual transliteration, where mappings from individual languages into the common phonological feature space can be reused when new translation directions are added.

4 Set-up and results of the evaluation experiment

This section presents a methodology for automated performance-based evaluation that is used in testing different settings of phonological categories and values for the extended Levenshtein metric. The experiment is set up in the following way:

(1) Small freely available electronic dictionaries for Ukrainian–Russian and Russian–Ukrainian directions were used to develop a gold-standard translation glossary of 11000 Ukrainian words, each having one or more Russian translation equivalents. All source words and their translation equivalents were used as they appear in the dictionaries (for the Russian–Ukrainian dictionary the translation direction was reversed and the translation equivalents missing from the original Ukrainian–Russian list were added to it. Cognates were not specifically selected or annotated in any way, so the gold standard evaluation set represented a standard introductory size bilingual glossary, such that similar resources could be found or compiled for many other language pairs.

(2) For identification of cognates two large monolingual corpora of Ukrainian and Russian news were used (250 million words each) with a standard morphological annotation of parts-of-speech (PoS) and lemmas. For each language frequency lists of lemmas and PoS codes were generated from these morphologically annotated corpora. After this the source and target words from the Ukrainian—Russian glossary have been intersected with the Russian and Ukrainian word lists compiled from PoS-tagged corpora for corresponding languages. The resulting Ukrainian evaluation set with corresponding gold-standard Russian dictionary equivalents included only those entries that were found both on the source and target sides in the glossary and both in the Ukrainian and Russian monolingual word lists. As a result, the evaluation set contained only the entries that could in principle be found by the cognate identification tool in the word lists and evaluated using the glossary.

(3) An additional requirement has been introduced that in both word lists the cognates should be tagged with the same part-of-speech. This reduces the search space for cognates and computing time needed to calculate phonological Levenshtein distances.

(4) Candidate cognate lists were generated for 809 randomly selected entries from the Ukrainian evaluation set in the following way. For each Ukrainian word in the evaluation set different variants of the Levenshtein edit distances were calculated to each word in the large Russian monolingual word list from the news corpus (around 106.000 unique lemmas, further filtered by their of speech codes). This process is computationally intensive and required parallel processing of the Ukrainian test entries on a high-performance computing cluster. Even though calculation of the baseline traditional Levenshtein distance is relatively fast, calculation of the phonological variant of this metric is much more computationally demanding, as it requires generating and comparing phonological feature sets for each of the compared characters in a large number of strings. For the current implementation, sequential generation of the phonological Levenshtein edit distances between a test Ukrainian entry and each of the 106.000 entries in the Russian monolingual word list takes about 4 minutes of computing time (54 hours of sequential computation for the whole evaluation set of 809 Ukrainian words).

In future, for the task of a large-scale induction of cognates between languages phonological feature representations will be optimised for speed and other techniques such as hashing of phonological features for the searched target entries will be implemented, which is expected to make the developed metric more usable for generation of wide-coverage translation resources.

(5) Candidate cognate lists were ranked according to distance scores produced by the fol-

lowing edit distance metrics: the Baseline Levenshtein edit distance, the phonological Levenshtein distance that used flat feature vectors, and by five variants of the phonological Levenshtein distance metric that used hierarchical phonological feature representations and one of the five possible weights for insertions/deletions: 0.2, 0.4, 0.6, 0.8 and 1.

For each Ukrainian word from the evaluation set, its Russian translation equivalents from the gold standard dictionary translations were automatically searched in the ranked cognate lists generated for that word by different variants of the Levenshtein metric. The position of the top dictionary translation equivalent was recorded in each of the ranked cognate lists.

(6) Even though dictionary equivalents were not necessarily cognates in the evaluation set, the experiment produced meaningful results, because non-cognate equivalents were simply not found and disregarded for the consideration. In this way the experimental set-up automatically focussed on the quality of cognate identification. Importantly, this allows us to avoid expensive manual selection or annotation of cognates: as the evolution methodology is automatic, all translation equivalents available in the gold standard are treated equally: in this stage no distinction is made between cognates and non-cognate equivalents. This removes the need for the manual filtering of the gold standard and also naturally covers ‘near-cognates’ or words with cognate morphemes where only parts of words match. Since the baseline and the modified Levenshtein metric are evaluated on the same gold standard, performance figures are relative and show the difference in finding translation equivalents for any degree of ‘cognateness’.

(7) Different variants of the metric are compared by the following parameters: Median top-N number for the metric; In top-1, top-5, top-10 and top-25.

(8) The following settings were compared:

(a) Baseline Levenshtein edit distance;

(b) Levenshtein distance extended with phonological features with flat feature vectors;

(c) Levenshtein distance extended with hierarchical phonological features (where manner and active place of articulation are treated as top-level features, which need to be matched in order for other features to match;

(d) Variants of the (b) and (c) metric with different insertion / deletion values – between 0.2 and 0.8.

The results of the evaluation experiment are presented in Table 1, where:

BaseL Lev = baseline Levenshtein metric

Phon Lev H = Phonological extension to Levenshtein metric with feature hierarchy

Phon Lev V = Phonological extension to Levenshtein metric with flat feature vectors

PhonLevi=0.X = Phonological extension to Levenshtein metric with modified insertion / deletion cost: i0.2 = the cost of insertion deletion is set to 0.2, i0.8 = is set to 0.8 (it is set to 1 in the Phon Lev metrics.

5 Discussion of the results, conclusion

It can be seen from Table 1 that:

(1) Hierarchical phonological Levenshtein metric outperforms the baseline on the Top 1 and Top 2 measures, the median rank improvements is +5%

(2) Flat phonological feature vector metric on all measures performs worse than the baseline. This can be interpreted as the need to take into account the order of matching higher-level features. Match of low-level features is not meaningful if higher-level features are not matched.

(3) The Hierarchical metric with insertion / deletion cost set to 0.8 outperforms both the baseline and the Levenshtein metric with the insertion/deletion cost = 1, especially on the Median Top N, Top 1 and Top5 measures. This can be interpreted as the need to scale down insertion cost moderately, since the average substitution cost is down.

Experiment	**Median topN**	**Top 1**	**Top 5**	**Top 10**	**Top 25**
BaseL Lev	50	206	328	360	382
Phon Lev V	87.5	**215**	289	319	349
DiffBase L	-75%	+4.4%	-10%	-11%	-9%
PhLev Hierarchy:					
PhLev i=0.2	125.5	216	291	315	342
PhLev i=0.4	54.5	230	307	334	367
PhLev i=0.6	*48*	235	328	354	385
PhLev i=0.8	**40**	**240**	**337**	**359**	**391**
Ph Lev i=1.0	47.5	**240**	334	359	385
Best BaseL Improv	**+20%**	**+16.5%**	**+3%**	**0%**	**2%**

Table 1: Automated evaluation of metric settings.

The results show that phonological extension to the Levenshtein edit distance metric on the task of cognate identification outperforms the character-based baseline. The proposed framework also allows accurate calibration of the feature arrangement and other parameter settings of the metric.

The modified Levenshtein metrics, phonological features sets for several alphabets and sample input files are released as an open-source software on the github repository (Babych, 2018).

Future work will include systematic evaluation of different possible feature hierarchies and costs, and metrics application to other tasks, such as transliteration.

References

Anderson, Stephen R. 1985. *Phonology in the twentieth century: Theories of rules and theories of representations.* University of Chicago Press.

Babych, Bogdan. 2016. Graphonological Levenshtein Edit Distance: Application for Automated Cognate Identification. Baltic Journal of Modern Computing 4.2 (2016): 115.

Babych, Bogdan. 2018. Phonological models for cognate terminology identification. GitHub repository, https://github.com/bogdanbabych/cognates-phonology

Jakobson, Roman, and Moris Halle. 1956. *Fundamentals of language.* Vol. 1. Walter de Gruyter. URL: http://pubman.mpdl.mpg.de/pubman/item/escidoc:2350620/component/escidoc:2350619/Jakobson_Halle_1956_fundamentals.pdf

Kondrak, Grzegorz. 2000. A new algorithm for the alignment of phonetic sequences. In Proceedings of NAACL 2000, pages 288–295.

Kondrak, Grzegorz 2009. Identification of cognates and recurrent sound correspondences in word lists. Traitement automatique des langues et langues anciennes, 50(2):201–235.

Kondrak, Grzegorz, and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In Proceedings of the Workshop on Linguistic Distances. Association for Computational Linguistics

Ladefoged, Peter. 1995. A Course in Phonetics. New York: Harcourt Brace Jovanovich

Lamraoui, Fethi, and Philippe Langlais. 2013. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment. *XIV Machine Translation Summit*. URL: http://rali.iro.umontreal.ca/rali/sites/default/files/publis/MTSummit-2013-Fethi.pdf

Levenshtein, Vladimir I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady.* Vol. 10. No. 8.

Nerbonne, John, and Wilbert Heeringa. 1997. Measuring dialect distance phonetically. *Computational Phonology: Third Meeting of the ACL Special Interest Group in Computational Phonology.*

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers). Vol. 1. URL: http://www.aclweb.org/anthology/P16-1162

St Arnaud, Adam, David Beck, and Grzegorz Kondrak. 2017. Identifying Cognate Sets Across Dictionaries of Related Languages. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Varga, Dániel, Peter Hal´acsy, Andras Kornai, Viktor Nagy, Laszl´o N´emeth and Viktor Tron. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4 292*: pp. 247-253. URL: http://eprints.sztaki.hu/7902/1/Kornai_1762382_ny.pdf

Rule-based machine translation from Kazakh to Turkish

Sevilay Bayatli
Dept. Elec. and Computer Engineering
Altınbaş Üniversitesi
sevilaybayatli@gmail.com

Sefer Kurnaz
Dept. Elec. and Computer Engineering
Altınbaş Üniversitesi
sefer.kurnaz@altinbas.edu.tr

Ilnar Salimzianov
School of Science and Technology
Nazarbayev University
ilnar@selimcan.org

Jonathan North Washington
Linguistics Deptartment
Swarthmore College
jonathan.washington@swarthmore.edu

Francis M. Tyers
School of Linguistics
Higher School of Economics
ftyers@hse.ru

Abstract

This paper presents a shallow-transfer machine translation (MT) system for translating from Kazakh to Turkish. Background on the differences between the languages is presented, followed by how the system was designed to handle some of these differences. The system is based on the Apertium free/open-source machine translation platform. The structure of the system and how it works is described, along with an evaluation against two competing systems. Linguistic components were developed, including a Kazakh-Turkish bilingual dictionary, Constraint Grammar disambiguation rules, lexical selection rules, and structural transfer rules. With many known issues yet to be addressed, our RBMT system has reached performance comparable to publicly-available corpus-based MT systems between the languages.

1 Introduction

In this paper we present a prototype shallow-transfer rule-based machine translation system using the Apertium free/open-source machine translation platform (Forcada et al., 2011) for translating from Kazakh to Turkish.

One of the most common criticisms towards Rule-Based Machine Translation (RBMT) regards the amount of work necessary to build a system for a new language pair (Arnold, 2003). In fact, in a traditional scenario, linguists with expertise in the source and target language need to manually build all the dictionary entries and transfer rules. Conversely, in a corpus-based/statistical MT approach (Koehn, 2010), no such effort is required as the system can be automatically built from parallel corpora providing they exist. If parallel corpora do not exist, then we see two options that remain. The first is to create a new parallel corpus, either by translating millions of words from scratch (requiring effort from translators),[1] or by finding parallel text online and processing it (requiring effort from programmers). The second option is to build a rule-based machine translation system (requiring effort from linguists). The most labour-intensive of these approaches is to translate the data from scratch, although this might be practical in certain situations. Building a rule-based machine translation system and finding and processing parallel texts from the internet are, given equal available expertise, around equally time consuming. As large, freely available parallel corpora are not known to exist between Kazakh and Turkish and we were interested in structural differences between these two languages — and producing new linguistic resources, we chose to build a rule-based MT system. Kazakh and Turkish are different enough that native speakers are not able to make sense of the other language, but also share similar enough structure that an RBMT system is feasible with some level of linguistic knowledge.

This paper demonstrates that, with many known issues yet to be addressed, our RBMT system has already reached performance comparable to publically available SMT systems between the languages. This has been accomplished solely with open source tools and some level of linguistic knowledge about

[1] The millions of words number is taken from Koehn and Knowles (2017) who compare neural MT against phrase-based SMT for English–Spanish for 0.4 million to 385.7 million words. Even for these morphologically-poor languages, even with one million words the performance is poor.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 49–58
Alacant, Spain, May 2018.

the two languages, and without large parallel corpora or machine learning algorithms (although some of the components in our opinion could be significantly improved by using them; see Section 6 for more details).

The paper will be laid out as follows: Section 2 gives a short review of some previous work in the area of Turkic-Turkic language machine translation and an overview of other publically available Kazakh-Turkish machine translators; Section 3 introduces Kazakh and Turkish and compares their grammar; Section 4 describes the system and the tools used to construct it; Section 5 gives a preliminary evaluation of the system; Section 6 describes our aims for future work; and finally Section 7 contains some concluding remarks.

2 Previous work

Within the Apertium project, there is ongoing work on building MT systems (and thus underlying components such as morphological transducers) for translating between two Turkic languages, between a Turkic language and Russian or between a Turkic language and English. Among released MT systems there are: Kazakh-Tatar (Salimzyanov et al., 2013), Tatar-Bashkir (Tyers et al., 2012b), Crimean Tatar-Turkish and English-Kazakh (Sundetova et al., 2015) MT systems.

Several other MT systems have been reported that translate between Turkish and other Turkic languages, including Turkish–Crimean Tatar (Altıntaş, 2001), Turkish–Azerbaijani (Hamzaoglu, 1993), Turkish–Tatar (Gilmullin, 2008), and Turkish–Turkmen (Tantuğ et al., 2007a,b) MT systems. As for the systems for translating to/from Kazakh (besides the ones already mentioned above), there is a bidirectional Kazakh-English machine translation system (Tukeyev et al., 2011) which uses a link grammar and statistical approach. None of these MT systems to our knowledge have been released to a public audience.

Altenbek and Xiao-long (2010) propose a segmentation system for inflectional affixes of Kazakh. Makhambetov et al. (2015), Kessikbayeva and Cicekli (2014) and Kairakbay and Zaurbekov (2013) present work on Kazakh morphological analysis.

Both Kazakh and Turkish are among the languages supported by Google Translate[2] and Yandex Translate[3] tools.

3 The languages

Kazakh is classified as a member of the Northwestern (or Kypchak) branch of the Turkic language family. It is primarily spoken in Kazakhstan, where it is the national language, sharing official status with Russian. Large communities of native speakers also exist in China, neighbouring Central Eurasian republics, and Mongolia. The total number of speakers is at least 10 million people (Simons and Fennig, 2018). The present-day Kazakh Cyrillic alphabet consists of 42 letters, 33 of which are letters found in the Russian alphabet. There are controversial plans to transition to a Latin alphabet by 2025.

Turkish is classified as a member of the Southwestern (or Oghuz) branch of Turkic language family. With over 70 million L1 speakers (Simons and Fennig, 2018), it is the Turkic language spoken by the most people. The Turkish Latin alphabet contains 29 letters.

Kazakh and Turkish exhibit agglutinative morphology, meaning that word forms may consist of a root and a series of affixes.

An MT system between Kazakh and Turkish is potentially of great use to the language communities. Automatic MT can save time and money over going through e.g. Russian or English (and the system is much easier to develop).

We continue with a brief overview of some differences in phonology, orthography, morphology and syntax of the languages. A complete and detailed comparison is out of scope of this work.

3.1 Phonology and orthography

Differences in phonology and orthography are less relevant for this work because relatively high-coverage morphological transducers were available for both of the languages when we started working on the translator. The mutual intelligibility of Kazakh and Turkish, both in their spoken and written forms, is rather low, despite much similar morphology and the existence of many cognates, often with similar meanings.

3.2 Morphology and syntax

3.2.1 Verbals

There are verbal tenses and moods common to both Kazakh and Turkish, like the definite past tense, the imperative mood, and the conditional mood. There are also quite a few differences. For example, Kazakh lacks the definite future tense affix **-{y}{A}c{A}k** known in Turkish; but has the so

[2] http://translate.google.com

[3] http://translate.yandex.com

called *goal oriented future* tense, absent in Turkish: e.g., the Kazakh verb form *бармакпын* ‘I intend to go’ can be translated into Turkish as *gitmeyi düşünüyorum* ‘I intend to go’. Another example of an affix found in one language but not in the other is the affix **-{D}{A}й** in Kazakh, which follows nouns and numbers and indicates resemblance, and can often be translated as the postposition *gibi* ‘like’ in Turkish.

Kazakh has several auxiliary verbs which are used for constructing analytic verbal forms. Four of them, the auxiliary verbs *жатыр, отыр, жүр, тұр* are used to construct the present continuous tense (Muhamedow, 2016), as in the collocation *жауып жатыр* ‘is raining’, translated to Turkish as *yağıyor* ‘is raining’. There are many other cases (i.e., not just due to analytic tenses) in which sequences of two or more Kazakh verbs map to a single verb in Turkish, as in the case of the expression *қуанып кетті* ‘gladdened’, which is translated as *neşelendi* ‘gladdened’ in Turkish.

In the case of non-finite forms, there are one-to-many correspondences. For instance, Kazakh past verbal adjectives (participles) formed with the **-{G}{A}н** suffix can be translated into Turkish in at least three ways: as past verbal adjective with the **-m{I}ş** suffix, as a subject-relative verbal adjective formed with the **-{y}{A}n** suffix or as a past verbal adjective formed with the **-{D}{I}k** suffix. As an example, the Kazakh sentence *Сербия мен Қазақстан арасында шешілме**ген** мәселе жоқ.* ‘There aren’t any **unresolved** issues between Serbia and Kazakhstan’ can be translated into Turkish as *Sırbistan ve Kazakistan arasında çözümlenme**miş** mesele yok.*, whereas the sentence *Екі мемлекет басшылары шағын және кеңейтіл**ген** құрамда келіссөздер жүргізді.* ‘The two leaders held talks in small and **expanded** format.’ in the parallel corpus we constructed (see Section 4.3) is translated as *İki memleket başkanları küçük ve genişletil**diği** kapsamda müzekereler yönetti.*

Similarly, the Kazakh imperfect verbal adjective formed with the suffix **-{E}т{I}н** is translated as either a subject-relative verbal adjective formed with the **-{y}{A}n** suffix or as future verbal adjective constituted with **-{y}{A}c{A}k** suffix. For example, the Kazakh phrase *сөйлейтін* can be translated as *konuşacak* ‘which will speak’ or as *konuşan* ‘(which is) speaking’.

3.2.2 Nominals

Another example of a morphological difference between Kazakh and Turkish is the presence of a four-way distinction in Kazakh’s 2nd person system (both pronouns and agreement suffixes). In other words, in Kazakh there is a distinct word for all combinations of [±plural, ±formal] (Muhamedow, 2016), whereas the Turkish 2nd person singular formal pronoun coincides with the 2nd person plural informal and 2nd person plural formal pronouns, as summarized in Table 1 (both *siz* and *sizler* are used as the plural formal pronoun in Turkish).

	Kazakh		**Turkish**	
	-PLUR	+PLUR	-PLUR	+PLUR
-FRM	сен	сендер	sen	siz
+FRM	сіз	сіздер	siz	siz/sizler

Table 1: Second person personal pronouns in Kazakh and Turkish. Note the extra distinctions in the Kazakh forms.

All of the differences or one-to-many/many-to-one correspondences in morphology and syntax described in this and the preceding subsections are relevant for a shallow-transfer RBMT because to handle them is the main job of the transfer component.

4 System

Our machine translation system is based on the Apertium MT platform (Forcada et al., 2011).[4] The platform was originally aimed at the Romance languages of the Iberian peninsula, but has also been adapted for other, more distantly related, language pairs. The whole platform, both programs and data, are licensed under the Free Software Foundation’s General Public Licence[5] (GPL) and all the software and data for the completed supported language pairs (and the other pairs being worked on) is available for download from the project website.

4.1 Architecture of the system

A typical translator built using the Apertium platform, including the translator described here, consists of a Unix-style pipeline or assembly line with the following modules (see Fig. 1. In Table 2 you can see an example of how a Kazakh sentence passes through the pipeline):

- **De-formatter.** Separates the text to be translated from the formatting tags. Formatting tags

[4] `http://www.apertium.org`

[5] `http://www.fsf.org/licensing/licenses/gpl.html`

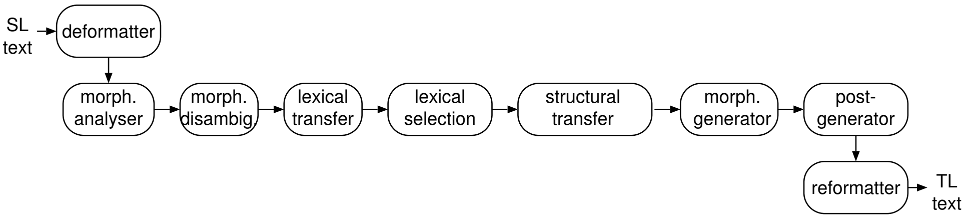


Figure 1: The pipeline architecture of a typical Apertium MT system.

are encapsulated in brackets so they are treated as "superblanks" that are placed between words in such a way that the remaining modules see them as regular blanks.

- **Morphological analyser.** Segments the source-language (SL) text in surface forms (SF) (words, or, where detected, multiword lexical units) and for each, delivers one or more lexical forms (LF) consisting of lemma (dictionary or citation form), lexical category (or part-of-speech) and inflection information.
- **Morphological disambiguator.** A morphological disambiguator, in case of the Kazakh-Turkish translator based on the Constraint Grammar (CG) formalism (Karlsson et al., 1995), chooses the most adequate sequence of morphological analyses for an ambiguous sentence.
- **Lexical transfer.** This module reads each SL LF and delivers the corresponding target-language (TL) LF by looking it up in a bilingual dictionary encoded as an finite-state transducer compiled from the corresponding XML file. The lexical transfer module may return more than one TL LF for a single SL LF.
- **Lexical selection.** A lexical selection module Tyers et al. (2012a) chooses, based on context rules, the most adequate translation of ambiguous SL LFs.
- **Structural transfer.** A structural transfer module, which performs local syntactic operations, is compiled from XML files containing rules that associate an action to each defined LF pattern. Patterns are applied left-to-right, and the longest matching pattern is always selected.
- **Morphological generator.** It transforms the sequence of target–language LFs, produced by the structural transfer, to a corresponding sequence of target–language SFs.
- **Post-generator.** Performs orthographic operations, for example elision (such as *da* + *il* = *dal* in Italian). This module has not been employed in our translator so far.
- **Reformatter.** De-encapsulates any format information.

The Apertium platform provides what can be called a 'vanilla' program and a formalism for describing linguistic data (if the module in question requires it) for each of the modules. We want to emphasise though that modules of the pipeline just described are independent from each other and thus can rely on different programs, different formalisms, and be of rule-based, statistical or hybrid nature. For example, Constraint Grammar-based morphological disambiguator can be considered a drop-in replacement for the Hidden Markov Model-based statistical tagger found in a few other Apertium MT systems. So are the formalisms used for morphological transducers which are described next.

4.2 Morphological transducers

The morphological transducers are based on the Helsinki Finite State Toolkit (Linden et al., 2011) – a free/open-source reimplementation of the Xerox finite-state tool chain, popular in the field of morphological analysis. It implements both the **lexc** formalism for defining lexicons, and the **twol** and **xfst** formalisms for modeling morphophonological rules. This toolkit has been chosen as it — or the equivalent XFST — has been widely used for other Turkic languages (Cöltekin 2010; Altintas and Cicekli 2001; Washington et al. 2012; Tantuğ et al. 2006; Tyers et al. 2012b,Washington et al. 2014, Çöltekin 2014) and is available under a free/open-source licence. The morphologies of both languages are implemented in lexc, and the morphophonologies of both languages are implemented in twol. The same lexc and twol files are used to compile both the morphological analyser and the morphological generator for each language.

(Kazakh) Input	Біз жаттығулар барысын мұқият бақылап отырдық.
Mor. analysis	^Біз/біз<prn><pers><p1><pl><nom>$ ^жаттығулар/жаттығу<n><pl><nom>/жаттығу<n><pl><nom>+е<cop><aor><p3><sp>$ ^барысын/бары<n><px3sp><acc>/барыс<n><px3sp><acc>$ ^мұқият/мұқият<adj>/мұқият<adv>/мұқият<adj>+е<cop><aor><p3><sp>$ ^бақылап/бақыла<v><tv><prc_perf>/бақыла<v><tv><gna_perf>$ ^отырдық/отыр<vaux><ifi><p1><pl>/отыр<v><iv><ifi><p1><pl>$^./.<sent>$
Mor. disambig	^Біз<prn><pers><p1><pl><nom>$ ^жаттығу<n><pl><nom>$ ^бары<n><px3sp><acc>$ ^мұқият<adv>$ ^бақыла<v><tv><prc_perf>$ ^отыр<vaux><ifi><p1><pl>$ ^.<sent>$
Lex. transfer	^Біз<prn><pers><p1><pl><nom>/Biz<prn><pers><p1><pl><nom>$ ^жаттығу<n><pl><nom>/çalışma<n><pl><nom>/egzersiz<n><pl><nom>$ ^бары<n><px3sp><acc>/süreç<n><px3sp><acc>$ ^мұқият<adv>/dikkatlice<adv>$ ^бақыла<v><tv><prc_perf>/gözlemle<v><tv><prc_perf>$ ^отыр<vaux><ifi><p1><pl>/<ifi><p1><pl>/otur<v><iv><ifi><p1><pl>$^.<sent>/.<sent>$
Structural transfer	^Biz<prn><pers><p1><pl><nom>$ ^çalışma<n><pl><nom>$ ^süreç<n><px3sp><acc>$ ^dikkatlice<adv>$ ^gözlemle<v><tv><ifi><p1><pl>$^.<sent>$
Mor. generation	Biz çalışmalar sürecini dikkatlice gözlemledik.

Table 2: Translation process (from Kazakh to Turkish) for the phrase *Біз жаттығулар барысын мұқият бақылап отырдық.* ‘We carefully followed the work process.’ Some analyses are omitted for reasons of space. Note how a transfer rule has transformed a participle + auxiliary construction of Kazakh, *бақылап отырдық* ‘we followed’, to an analytic construction in Turkish, *gözlemledik* ‘we followed’.

The Kazakh morphological transducer used in this work was presented in (Washington et al., 2014). Turkish morphological transducer also comes from the Apertium project. It has not been described in a published work yet. Both transducers were extended to support all stems from the bilingual lexicon we constructed.

We decided to use the Turkish morphological transducer developed in the Apertium project and not the also free/open-source TRMorph (Çöltekin, 2014), because the tagset used in the former is more consistent with morphological transducers developed in the Apertium project for other Turkic languages, including the Kazakh transducer. The consistency of the tagset allows to keep the transfer module relatively simple and pay more attention to the actual differences in the grammar of the languages rather than on differences in the tagset used.

4.3 Bilingual lexicon

The bilingual lexicon currently contains 7,385 stem-to-stem correspondences and was built mostly by hand in the following way. We assembled a parallel Kazakh-Turkish corpus. For this we took all sentences from the Kazakh treebank (Tyers and Washington, 2015) — approximately one thousand sentences — and translated them manually to Turkish. Then, these Kazakh and Turkish sentences were analysed with the `apertium-kaz` and `apertium-tur` morphological transducers. This provided the lemma and the part of speech tag for most of the surface forms in the corpora. The lemmas which were not already in the monolingual lexicons were added to them, and corresponding words were added to the bilingual lexicon. In addition, some of the stems present in the Kazakh lexc file but not found in the parallel corpus were translated into Turkish and added to the bilingual dictionary. Because of the similarity of the languages, the majority of entries in the bilingual dictionary (a file in an XML-based format) are one-to-one mappings of stems, but there are ambiguous translations. For example, the Kazakh word ‘азамат’ has two translations in Turkish: ‘sivil’ and ‘vatandaş’, as shown in Fig. 2.

4.4 Rules

The note made at the end of Section 4.1 on replacability of the components aside, Apertium is primarily a rule-based MT system. Not counting morphophonology (morphotactics) rules required by HFST-based morphological transducers, there are three main categories of rules in our system — morphological disambiguation rules, lexical selection rules and transfer rules. A description of each follows.

```
<e><p><l>қас<s n=”n”/></l>       <r>ruh<s n=”n”/></r></p></e>
<e><p><l>азамат<s n=”n”/></l>    <r>sivil<s n=”n”/></r></p></e>
<e><p><l>азамат<s n=”n”/></l>    <r>vatandaş<s n=”n”/></r></p></e>
<e><p><l>үлкен<s n=”adj”/></l>   <r>büyük<s n=”adj”/></r></p></e>
<e><p><l>ұлттық<s n=”adj”/></l> <r>ulusal<s n=”adj”/></r></p></e>
<e><p><l>дауа<s n=”n”/></l>      <r>çare<s n=”n”/></r></p></e>
<e><p><l>дауа<s n=”n”/></l>      <r>ilaç<s n=”n”/></r></p></e>
<e><p><l>дауа<s n=”n”/></l>      <r>çözüm<s n=”n”/></r></p></e>
<e><p><l>шешім<s n=”n”/></l>     <r>çözüm<s n=”n”/></r></p></e>
<e><p><l>шешім<s n=”n”/></l>     <r>karar<s n=”n”/></r></p></e>
```

Figure 2: Example entries from the bilingual lexicon. Kazakh is on the left, and Turkish on the right. Each stem is accompanied by a part-of-speech tag and there may be many–many correspondences between the stems.

4.4.1 Morphological disambiguation rules

The system has a morphological disambiguation module in the form of a Constraint Grammar (CG) (Karlsson et al., 1995). The version of the formalism used is vislcg3.[6] The goal of the CG rules is to select the correct morphological analysis when there are multiple analyses. We used the Kazakh CG previously developed partially by the authors of this paper and partially by other Apertium contributors. At the time of this writing the file contains 164 rules. Due to closeness of the languages, the majority of ambiguity may be passed through from one language to the other.

4.4.2 Lexical selection rules

In general, lexical selection rules are necessary to handle one-to-many correspondences of the bilingual lexicon. While many lexical items have a similar range of meaning, lexical selection is sometimes necessary when translating between Kazakh and Turkish as well. For example, the Kazakh word *ат* has two meanings: *ат* ‘name’ and *ат* ‘horse’ and can be translated into Turkish as either *ad* ‘name’ or *at* ‘horse’. A lexical selection rule chooses the translation *at* ‘horse’ if the immediate context includes a word *ұста* ‘hold’. Another example is the word *жарық*, which as a noun can mean either ‘light’ or ‘crack’. It is translated to Turkish by default as *ışık* ‘light’, and is translated as *yarık* ‘crack’ only in the immediate context of words like *есік* ‘door’ and *қабырға* ‘wall’. A relatively small number of 92 lexical selection rules were developed and added to the system. The lexical selection module we used (Tyers et al., 2012a) allows inferring such rules automatically from a parallel corpus, but we have not employed this feature of it yet.

4.4.3 Structural transfer rules

Apertium, as a rule, translates lemmas and morphemes one by one. Obviously, this does not always work, even for closely related languages. Structural transfer rules are responsible for modifying morphology or word order in order to produce “adequate” target language.

As seen in Table 2, the structural transfer module takes a sequence of (source language lexical form — target language lexical form) pairs in the following format: `^SL-lemma<SL-tag1><SL-tag2><…><SL-tagN>` `/TL-lemma<TL-tag1><TL-tag2><…><TL-tagN>$` TL lemma and tags are provided by the preceding two modules — lexical transfer and lexical selection. The lexical transfer module looks up the TL lemma and usually the first one or two tags (read: part of speech tag) in the bilingual transducer, the rest of the tags are carried over from the SL.

Figure 3 gives an example of a transfer rule. Any transfer rule consists of two core parts — of a pattern and an action. In this case, the pattern named “gpr_impf” matches Kazakh verbal adjectives formed with the **-{E}т{I}н** affix (for reasons of space, we omitted the definition of the pattern itself). Recall from Section 3.2.1 that Kazakh verbal adjectives ending in **-{E}т{I}н** have two possible translations in Turkish — either with a verbal adjective ending in **-{y}{A}n** suffix or with a verbal adjective ending in **-{A}c{A}k** suffix. The rule in Figure 3 replaces the `<gpr_impf>` tag on the TL side with the `<gpr_rsub>` tag, which corresponds to a **-{y}{A}n** verbal adjective in Turkish. In addition, transfer rules perform chunking, and later transfer stages can operate on chunks of words as if they were single words, but we will not discuss chunking-based rules here since this technique is currently not employed in the Kazakh-Turkish translator.

The patterns are matched on the SL side by the

[6] http://beta.visl.sdu.dk/constraint_grammar.html

```
<rule comment=”REGLA: gpr_impf-5” > <!--сөйлейтін -> konuşan -->
  <pattern><pattern-item n=”gpr_impf”/></pattern>
  <action>
    <let><clip pos=”1” side=”tl” part=”a_impf”/><lit-tag v=”gpr_rsub”/></let>
    <out>
      <chunk name=”gpr” case=”caseFirstWord”>
        <tags><tag><lit-tag v=”SV”/></tag></tags>
        <lu><clip pos=”1” side=”tl” part=”whole”/></lu>
      </chunk>
    </out>
  </action>
</rule>
<rule comment=”REGLA: ger_perf-7” > <!--білгендік -> bildik -->
 <pattern><pattern-item n=”ger_perf”/></pattern>
 <action>
   <let><clip pos=”1” side=”tl” part=”ger_prf”/><lit-tag v=”ger_past”/></let>
   <out>
    <chunk name=”v” case=”caseFirstWord”>
      <tags><tag><lit-tag v=”SV”/></tag></tags>
      <lu><clip pos=”1” side=”tl” part=”whole”/></lu>
    </chunk>
   </out>
 </action>
</rule>
```

Figure 3: Examples of a transfer rule. The first rule translates Kazakh -{E}т{I}н verbal adjectives with -(y)An verbal adjectives by replacing the `<gpr_impf>` tag on the TL side with the `<gpr_rsub>` tag. The second rule transfer Kazakh -{G}{A}нл{I}{K} verbal noun (gerund) into -{D}{I}k verbal noun (Gerund) by replacing the `<ger_perf>` tag on the TL side with the `<ger_past>` tag.

"left–to–right, longest–match" principle. For instance, Kazakh determiner–adjective–noun phrase *Бұл үлкен жетістік* 'This big success' will be matched and processed by a rule having determiner–adjective–noun sequence as its pattern, and not by e.g. a rule matching determiner–adjective or determiner–noun sequences. If there are ties, the rule which is placed higher in the source file is applied first.

Since Kazakh and Turkish have similar syntax, most of the rules are not about reordering but about altering the tags carried over from SL LFs to TL LFs. We used the parallel corpus described in Section 4.3 as a development set for writing and refining transfer rules. In all, we have defined 76 structural transfer rules.

5 Evaluation

The system has been evaluated in two ways. The first is its coverage. The second is translation quality — the error rate of two pieces of text produced by the system when comparing with postedited versions of them.

5.1 Coverage

Coverage of the system was calculated over three freely-available corpora — a dump of articles from Kazakh Wikipedia, a Kazakh translation of the Quran, and the Kazakh translation of the Bible. The corpus extracted from Kazakh Wikipedia[7] contains 22,515,314 words (1,404,467 sentences). Wikipedia is one of the major uses for Apertium translators, especially since some of these are used by the Wikimedia Content Translation Tool[8].

Corpus	Tokens	Coverage (%)
Wikipedia	22,515,314	83.42
Quran.altay	107,451	70.30
Bible.kkitap	577,070	80.89

Table 3: Coverage of the Kazakh-Turkish MT system

The coverage is the percentage of tokens in running text which are translated by the MT system. Apertium MT systems mark tokens that could not be analysed with a special sign, thus coverage was calculated by dividing the number of tokens without that special sign by the total number of words in the text. The coverage results are presented in Table 3. As seen in the table, the coverage of the system is not very high, with an average of 78.20% .

[7] https://kk.wikipedia.org/; kkwiki-20180320-pages-articles.xml.bz2

[8] https://www.mediawiki.org/wiki/Content_translation

5.2 Translation quality

The translation quality was measured using two metrics, the first was word error rate (WER), and the second was position-independent word error rate (PER). Both metrics are based on the Levenshtein distance (Levenshtein, 1965). Metrics based on word error rate were chosen as to be able to compare the system against systems based on similar technology, and to assess the usefulness of the system in a real setting, that is of translating for dissemination.

Besides calculating WER and PER for our Kazakh-Turkish MT system, we did the same for two other publically available Kazakh-Turkish MT systems — from Google Translate and Yandex Translate. The procedure was the same for all three. We took a small (1,025 tokens) Kazakh text, which was a concatenation of several articles from Wikipedia and translated it using the three MT systems. The output of each system was postedited independently to avoid biasing in favour of one particular system. Then we calculated WER and PER for each using the `apertium-eval-translator` tool[9] and BLEU using the `mteval-v13a.pl` script.[10] Note that BLEU score is typically calculated by comparing against a pre-translated reference translation, where here we calculate against posteditted reference translations for each of the systems.

5.2.1 Results

Table 4 shows the results obtained for all three systems — Google, Yandex and Apertium. Google's[11] MT system has the lowest WER. Apertium has a comparable WER despite having much higher number of OOV words. Yandex Translate's[12] WER is higher, but PER is similar to the other two.

These numbers can be compared with scores for other translators based on the Apertium platform. For example, the Kazakh–Tatar system described in (Salimzyanov et al., 2013) achieves post-edition WER of 15.19% and 36.57% over two texts of 2,457 words and 2,862 words respectively. The Tatar–Bashkir system in (Tyers et al., 2012b) reports WER of 8.97% over a small text of 311 words and WER of 7.72% over another text of 312 words. The higher word error rate can be explained by the fact that Kazakh and Turkish are more distantly related than Tatar and Kazakh or Bashkir.

5.3 Error analysis

The majority of remaining errors are mostly due to a lot of unknown words (because the relatively low number of words in the bilingual dictionary), and disambiguation errors.

6 Future work

We intend to continue the development of the system to improve the quality of the translations. There are a number of areas where we believe that more work would yield better results, among which are the following:

- **Coverage**. By expanding the dictionaries with new lists of stems, and providing bilingual correspondences, the error rate will decrease and, consequently, there will be less post-editing work necessary (and translations will be much more intelligible). The principle issue here is adequate Kazakh–Turkish lexicography which is an under-investigated area.

- **Ambiguous transfer rules**. The "left-to-right, longest match" principle by which structural transfer rules are applied (which implies that if there are several rules with the same pattern, only one of them will apply — the first one that matches that pattern), although makes it easy to change the behaviour of the system (simply by reordering rules), in our opinion, is quite limiting. In particular, it limits the ways how one-to-many correspondences can be handled (several examples of such cases were given in Section 3), essentially forcing the developer to hard code one of the several possible translations as the default one and checking the surrounding lemmas or other features in the rule manually if he wishes to select an alternative translation. We conceive a method of selecting the most adequate rules for a given input by setting weights learned with an unsupervised learning algorithm.

7 Conclusion

To our knowledge we have presented the first free/open-source MT system between Kazakh and Turkish. The performance is similar to other translators created using the same technology, and in

[9] `http://wiki.apertium.org/wiki/apertium-eval-translator`

[10] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl`

[11] `https://translate.google.com/#kk/tr/`

[12] `https://ceviri.yandex.com.tr/`

System	OOV	WER (%)	PER (%)	BLEU
Yandex	43	69.73	48.63	2.84
Apertium	128	45.77	41.69	**16.67**
Google	5	**43.85**	**33.67**	16.32

Table 4: Word error rate and Position-independent word error rate; OOV is the number of out-of-vocabulary (unknown) words. The Google system has a similar word error rate to the Apertium system despite the significantly lower number of out-of-vocabulary words. Note that the BLEU scores are computed against a *posteditted* reference translation.

terms of WER to SMT systems available. The system beats the SMT systems on BLEU, while having a much higher out-of-vocabulary rate. This would suggest that given better vocabulary coverage the system would perform significantly better than SMT and NMT systems. The system is available as free/open-source software under the GNU GPL and the whole system may be downloaded from Github.[13]

Acknowledgements

We would like to thank Marzhan Zhakupova for translating the Kazakh sentences in the development corpus.

References

Altenbek, G. and Xiao-long, W. (2010). Kazakh segmentation system of inflectional affixes. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 183–190.

Altıntaş, K. (2001). *Turkish to Crimean Tatar machine translation system*. PhD thesis, Bilkent University.

Altintas, K. and Cicekli, I. (2001). A morphological analyser for Crimean Tatar. In *Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'2001)*, pages 180–189.

Arnold, D. (2003). Why translation is difficult for computers. *Computers and Translation: A translator's guide, Amsterdam and Philadelphia: John Benjamins*, pages 119–42.

Cöltekin, C. (2010). A freely available morphological analyzer for Turkish. In *LREC*, volume 2, pages 19–28.

Çöltekin, Ç. (2014). A set of open source tools for Turkish natural language processing. In *LREC*, pages 1079–1086.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Gilmullin, R. (2008). The Tatar-Turkish machine translation based on the two-level morphological analyzer. *Interactive Systems and Technologies: The Problems of Human-Computer Interaction*, pages 179–186.

Hamzaoglu, I. (1993). *Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language*. PhD thesis, MSc Thesis, Bogazici University, Istanbul.

Kairakbay, B. M. and Zaurbekov, D. L. (2013). Finite state approach to the Kazakh nominal paradigm. In *FSMNLP*, pages 108–112.

Karlsson, F., Voutilainen, A., Heikkilae, J., and Anttila, A. (1995). *Constraint Grammar: a language-independent system for parsing unrestricted text*, volume 4. Walter de Gruyter.

Kessikbayeva, G. and Cicekli, I. (2014). Rule based morphological analyzer of Kazakh language. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 46–54.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., and Pirinen, T. (2011). *HFST–Framework for Compiling and Applying Morphologies*, volume 100 of *Communications in Computer and Information Science*, pages 67–85.

[13] `https://github.com/apertium/apertium-kaz-tur`

Makhambetov, O., Makazhanov, A., Sabyrgaliyev, I., and Yessenbayev, Z. (2015). Data-driven morphological analysis and disambiguation for Kazakh. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 151–163. Springer.

Muhamedow, R. (2016). *Kazakh: A Comprehensive Grammar*. Routledge: Oxford.

Salimzyanov, I., Washington, J., and Tyers, F. (2013). A free/open-source Kazakh-Tatar machine translation system. *Machine Translation Summit XIV*.

Simons, G. F. and Fennig, C. D. (2018). Ethnologue: Languages of the world. Twenty-first edition. *SIL, Dallas, Texas*.

Sundetova, A., Forcada, M., and Tyers, F. (2015). A free/open-source machine translation system for English to Kazakh. In *3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan*, pages 78–91.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2006). Computer analysis of the Turkmen language morphology. In *Advances in Natural Language Processing*, pages 186–193. Springer.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2007a). Machine translation between Turkic languages. pages 189–192. Association for Computational Linguistics.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2007b). An MT system from Turkmen to Turkish employing finite state and statistical methods.

Tukeyev, U., Melby, A., and Zhumanov Zh, M. (2011). Models and algorithms of translation of the Kazakh language sentences into English language with use of link grammar and the statistical approach. In *Proc. of IV Congress of the Turkic World Math. Society*, pages 1–3.

Tyers, F. M., Sánchez-Martínez, F., Forcada, M. L., et al. (2012a). Flexible finite-state lexical selection for rule-based machine translation.

Tyers, F. M. and Washington, J. (2015). Towards a free/open-source universal dependency treebank for Kazakh. In *Proceedings of the 3rd International Conference on Computer Processing in Turkic Languages (TurkLang)*, pages 276–289.

Tyers, F. M., Washington, J. N., Salimzyanov, I., and Batalov, R. (2012b). A prototype machine translation system for Tatar and Bashkir based on free/open-source components. In *First Workshop on Language Resources and Technologies for Turkic Languages*, page 11.

Washington, J., Ipasov, M., and Tyers, F. M. (2012). A finite-state morphological transducer for Kyrgyz. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 934–940.

Washington, J., Salimzyanov, I., and Tyers, F. M. (2014). Finite-state morphological transducers for three Kypchak languages. In *Proceedings of the 9th Conference on Language Resources and Evaluation (LREC)*, pages 3378–3385.

SRL for low resource languages isn't needed for semantic SMT

Meriem Beloucif and **Dekai Wu**
Human Language Technology Center
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
mbeloucif|dekai@cs.ust.hk

Abstract

Previous attempts at injecting semantic frame biases into SMT training for low resource languages failed because either (a) no semantic parser is available for the low resource input language; or (b) the output English language semantic parses excise relevant parts of the alignment space too aggressively. We present the first semantic SMT model to succeed in significantly improving translation quality across many low resource input languages for which no automatic SRL is available —consistently and across all common MT metrics. The results we report are the best by far to date for this type of approach; our analyses suggest that in general, easier approaches toward including semantics in training SMT models may be more feasible than generally assumed even for low resource languages where semantic parsers remain scarce.

While recent proposals to use the crosslingual evaluation metric XMEANT during inversion transduction grammar (ITG) induction are inapplicable to low resource languages that lack semantic parsers, we break the bottleneck via a vastly improved method of biasing ITG induction toward learning more semantically correct alignments using the *monolingual* semantic evaluation metric MEANT. Unlike XMEANT, MEANT requires only a readily-available English (output language) semantic parser. The advances we report here exploit the novel realization that MEANT represents an excellent way to semantically bias expectation-maximization induction even for low resource languages. We test our systems on challenging languages including Amharic, Uyghur, Tigrinya and Oromo. Results show that our model influences the learning towards more semantically correct alignments, leading to better translation quality than both the standard ITG or GIZA++ based SMT training models on different datasets.

1 Introduction

Statistical machine translation (SMT) for low resource languages has been a difficult task due to the unavailability of large parallel corpora. It becomes imperative to make learning from *small* data more efficient by adding additional constraints to create stronger inductive biases—especially linguistically well-motivated constraints, such as the shallow semantic parses of the training sentences. However, while automatic semantic role labeling (SRL) is readily available to produce shallow semantic parses for a high-resource *output* language (typically English), the problem is that SRL is usually not available for low resource *input* languages such as Tigrinya, Oromo, Uyghur or Uzbek.

In this paper, we propose a new method which adopts the monolingual semantic evaluation metric MEANT as a confidence-weighting measure to assess the degree of goodness of training instances, giving a newer strategy than Beloucif and Wu (2016a) who used the degree of compatibility or similarity between the semantic role labeling of the input and output sentences. Their approach might outperform ours for high-resource languages, but is completely inapplicable to low resource languages because XMEANT requires both the input and output semantic parses – whereas MEANT does not require an SRL parse for the low resource input language.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 59–68
Alacant, Spain, May 2018.

Additionally, we also introduce a notion of **semantic role labeling coverage** as a second English monolingual confidence-weighting measure. An SRL coverage score roughly quantifies what proportion of a sentence is accounted for by a shallow semantic parse. The variety of approaches proposed here belong to a family of semantic SMT methods that has recently been advanced, wherein SRL constraints or biases are injected very early in the SMT training pipeline so as to maximize their influence on what translation model is learned. We test our models on multiple difficult low resource translation tasks: Amharic, Somali, Tigrinya, Oromo, Uzbek and Uyghur always translating into English. Despite having SRLs only on the English side, we show that our models influence the learning toward more semantically correct alignments. Our results show that this way of inducing ITGs gives a better translation quality than the conventional ITG (Saers and Wu, 2009) and the traditional GIZA++ (Och and Ney, 2000) alignments.

2 Related work

2.1 Semantic frames in the SMT pipeline

Semantic role labeling (SRL) or shallow semantic parsing, is a task that defines the semantic event structure *who did what to whom*, *for whom*, *when*, *where*, *how* and *why* in a given sentence (Gildea and Jurafsky, 2002). Only a few works integrate information provided by an SRL in SMT. However, most of the approaches do not use SRL for training, but either for tuning, evaluation or post-processing. For instance, Wu and Fung (2009) have empirically shown that including SRL for post-processing the MT output improves the translation quality. Their method maximizes the crosslingual match of the semantic labels between the input and the output sentences. Many tools that use SRL for MT evaluation have been proposed such as the semantic evaluation metric MEANT, which adopts the principle that a good translation preserves the semantic event structure across translations (Lo and Wu, 2011a, 2012; Lo *et al.*, 2012) or XMEANT (Lo *et al.*, 2014), the crosslingual version of MEANT, which uses the foreign input instead of the reference translation.

Liu and Gildea (2010) and Aziz *et al.* (2011) use input language SRL to train a tree-to-string SMT system. Xiong *et al.* (2012) trained a two pass discriminative model to incorporate source side predicate-argument structures into SMT. Komachi *et al.* (2006) and Wu *et al.* (2011) preprocess the input sentence to match the verb frame alternations in the output side. Moreover, Beloucif *et al.* (2015) have shown that including a semantic frame based objective function at an early stage of training SMT systems gives better translations than relying on tuning loglinear weights against a semantic based objective function such as MEANT. All these approaches are *inapplicable* when translating low resource languages since they either require the input language semantic parse or both languages SRL parses.

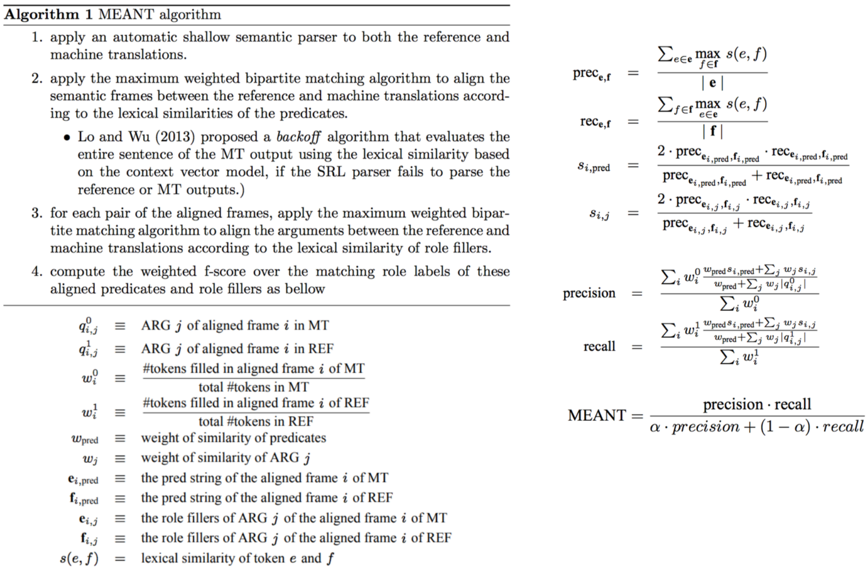
The most recent work that includes SRL during the actual learning of bilingual constituents for *low resource languages* is the one by Beloucif and Wu (2016b). However, our approach is quite different in spirit, and significantly outperforms theirs. Whereas their method for training ITGs penalizes bilingual constituents in the expectation-maximization (EM) biparse forests when they violate an English SRL, our training approach weights entire bilingual sentence pairs by predicting a confidence derived from MEANT. The problem with their approach is that they attempt to demote some partial hypotheses during the ITG training, which can excise relevant parts of the alignment search space aggressively.

2.2 The semantic based evaluation metric MEANT

The main model we propose adopts MEANT (Lo and Wu, 2011a, 2012; Lo *et al.*, 2012) to confidence-weight training instances. MEANT is a semantic frame based evaluation metric which compares the SRL parse of the MT output against the SRL parse of the reference translations provided. Then it produces a score that assesses the degree of similarity between their semantic frame structures. The MEANT algorithm is described in figure 1.

In figure 1, $q^0_{i,j}$ and $q^1_{i,j}$ are the arguments of type j in frame i in MT and REF respectively. w^0_i and w^1_i are the weights for frame i in MT/REF respectively.

The weights mentioned in the algorithm estimate the degree of contribution of each frame to the overall meaning of the sentence. w_{pred} and w_j are the weights of the lexical similarities of the predicates and role fillers of the arguments of type



Algorithm 1 MEANT algorithm

```
1. apply an automatic shallow semantic parser to both the reference and machine translations.
2. apply the maximum weighted bipartite matching algorithm to align the semantic frames between the reference and machine translations according to the lexical similarities of the predicates.
    • Lo and Wu (2013) proposed a backoff algorithm that evaluates the entire sentence of the MT output using the lexical similarity based on the context vector model, if the SRL parser fails to parse the reference or MT outputs.)
3. for each pair of the aligned frames, apply the maximum weighted bipartite matching algorithm to align the arguments between the reference and machine translations according to the lexical similarity of role fillers.
4. compute the weighted f-score over the matching role labels of these aligned predicates and role fillers as bellow
```

$$\begin{aligned}
q^0_{i,j} &\equiv \text{ARG } j \text{ of aligned frame } i \text{ in MT}\\
q^1_{i,j} &\equiv \text{ARG } j \text{ of aligned frame } i \text{ in REF}\\
w^0_i &\equiv \frac{\text{\#tokens filled in aligned frame } i \text{ of MT}}{\text{total \#tokens in MT}}\\
w^1_i &\equiv \frac{\text{\#tokens filled in aligned frame } i \text{ of REF}}{\text{total \#tokens in REF}}\\
w_{\text{pred}} &\equiv \text{weight of similarity of predicates}\\
w_j &\equiv \text{weight of similarity of ARG } j\\
\mathbf{e}_{i,\text{pred}} &\equiv \text{the pred string of the aligned frame } i \text{ of MT}\\
\mathbf{f}_{i,\text{pred}} &\equiv \text{the pred string of the aligned frame } i \text{ of REF}\\
\mathbf{e}_{i,j} &\equiv \text{the role fillers of ARG } j \text{ of the aligned frame } i \text{ of MT}\\
\mathbf{f}_{i,j} &\equiv \text{the role fillers of ARG } j \text{ of the aligned frame } i \text{ of REF}\\
s(e,f) &= \text{lexical similarity of token } e \text{ and } f
\end{aligned}$$

$$\begin{aligned}
\text{prec}_{\mathbf{e},\mathbf{f}} &= \frac{\sum_{e\in\mathbf{e}} \max_{f\in\mathbf{f}} s(e,f)}{|\,\mathbf{e}\,|}\\
\text{rec}_{\mathbf{e},\mathbf{f}} &= \frac{\sum_{f\in\mathbf{f}} \max_{e\in\mathbf{e}} s(e,f)}{|\,\mathbf{f}\,|}\\
s_{i,\text{pred}} &= \frac{2\cdot \text{prec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}\cdot \text{rec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}}{\text{prec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}+\text{rec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}}\\
s_{i,j} &= \frac{2\cdot \text{prec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}\cdot \text{rec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}}{\text{prec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}+\text{rec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}}
\end{aligned}$$

$$\begin{aligned}
\text{precision} &= \frac{\sum_i w^0_i \frac{w_{\text{pred}} s_{i,\text{pred}}+\sum_j w_j s_{i,j}}{w_{\text{pred}}+\sum_j w_j |q^0_{i,j}|}}{\sum_i w^0_i}\\
\text{recall} &= \frac{\sum_i w^1_i \frac{w_{\text{pred}} s_{i,\text{pred}}+\sum_j w_j s_{i,j}}{w_{\text{pred}}+\sum_j w_j |q^1_{i,j}|}}{\sum_i w^1_i}
\end{aligned}$$

$$\text{MEANT} = \frac{\text{precision}\cdot\text{recall}}{\alpha\cdot \mathit{precision} + (1-\alpha)\cdot \mathit{recall}}$$

Figure 1: The MEANT algorithm from left to right.

j of all frame between the reference translations and the machine translations. There is a total of 12 weights for the set of semantic role labels in MEANT as defined in Lo and Wu (2011b). They are determined using supervised estimation via a simple grid search to optimize the correlation with human adequacy judgments (Lo and Wu, 2011a).

3 Core model

The approaches proposed in this work inject a form of semantic parse bias into early stage word alignment using ITG (Wu, 1997) training, which (as shown in the results section) outperforms conventional GIZA++ (Och and Ney, 2000) based intersection/union-of-bidirectional-IBM-word-alignment strategies. Specifically, our defined approaches assume a token based BITG (bracketing ITG) (Wu, 1997) system, a choice based on previous works showing that: (a) BITG based alignments outperform GIZA++ alignments (Saers *et al.*, 2009); (b) ITG alignments have been empirically shown to cover almost 100% of *semantic* frame alternations, while ruling out the majority of incorrect alignments (Addanki *et al.*, 2012). The BITG model used in this work is initialized with uniform structural probabilities, setting aside half of the probability mass for lexical rules. The lexical probability mass is distributed among the lexical rules according to co-occurrence counts from the training data, assuming each sentence contains one empty token to account for singletons. These initial probabilities are refined with 10 iterations of EM, where the expectation step is calculated using beam pruned parsing (Saers *et al.*, 2009) with a beam width of 100. In the last iteration, the alignments imposed by the Viterbi parses are extracted as the final word alignments.

Saers and Wu (2011) showed how to compute expectations for EM re-estimation with outside probabilities as follows:

$$E_\theta = \frac{\alpha(M \rightarrow AL)\beta(M \rightarrow AL)}{\alpha(S_{0,|e|,0,|f|})\beta(S_{0,|e|,0,|f|})} \quad (1)$$

where $\alpha(M \rightarrow AL)$ and $\beta(M \rightarrow AL)$ are the inside and the outside probabilities of the derivation $M \rightarrow AL$ respectively. $\alpha(S_{0,|e|,0,|f|})$ is the initial inside probability, while $\beta(S_{0,|e|,0,|f|})$ represents the initial outside probability. Traditionally, the outside probability $\beta(S_{0,|e|,0,|f|})$ in the inside-outside algorithm is set to 1.0 as it represents the number of observations of a training instance (each bisentence is observed once). An intuitive way to distinguish good from bad sentences would be to favor sentences that have a good semantic parse, by setting the outside probability to be a weight (a fractional count between 0 and 1) that somehow reflects the goodness of the semantic parse better than a unified fractional count. Therefore, biasing the learning towards training instances which have a good SRL parse.

4 MEANT as a training objective function

4.1 Injecting MEANT

A more robust way to assess the degree of goodness of training instances has been shown to be the crosslingual evaluation metric XMEANT Beloucif and Wu (2016a). Unfortunately, this is not applicable in low resource settings since XMEANT assesses the compatibility between the English output and the input foreign language—for which the semantic parse is unavailable. Instead of computing the crosslingual compatibility between the input and the output semantic parses, we adopt the monolingual semantic frame evaluation metric MEANT as a confidence measure.

The evaluation metric MEANT computes the semantic frame coverage between the input and the MT reference. We propose to use MEANT as a confidence-weight measure by computing the semantic frame coverage in the English sentence. We obtain the SRL coverage of a sentence by computing the MEANT score between the input English sentence and the same sentence as a reference. We do not take into account the chunks that have no semantic parse (*backoff* was mentioned in figure 2).

Figure 2 illustrates two out of three possible situations for applying MEANT as a confidence-weight measure. The sentences that are fully semantically parsed like [ARG0 I][TARGET ate][ARG1 an apple]. have a MEANT score equal to 1.0. If the sentence is partially SRLed, the MEANT score is less than 1.0. For instance, the MEANT score for the parse Where do [ARG0 I][TARGET get][ARG2 off] to go to Union Square? is less than 1, but higher than 0. Furthermore, we note that a few sentences have a 0 MEANT score. In fact, we have experimented with three automatic SRLs: ASSERT (Pradhan *et al.*, 2004), MATE (Björkelund *et al.*, 2009) and MATEPLUS (Roth and Woodsend, 2014); we have observed that these SRL systems completely fail to parse sentences containing the verb to be; sentences like the light was red are ignored. However, we show that even while ignoring sentences containing to be, our systems are still outperforming conventional models on multiple challenging low resource languages.

4.2 Injecting monolingual SRL coverage

The second new strategy for judging the reliability of training instances using semantics is the monolingual SRL coverage, which looks at the proportion of a sentence that is accounted for by the English semantic parse. In its simplest, monolingual form, we define the monolingual coverage as follows:

$$\varphi_1 = (\text{\# labels / \# words labelled}) + \beta_0 \qquad (2)$$

where β_0 is a hyperparameter that is manually set to avoid eliminating sentences with 0 probability. The intuition in this approach is to give a higher SRL coverage to sentences that are easily SRLed and a low coverage to complex sentences that are hard to parse by an automatic SRL. For instance, the SRL parse: okay, sure. [TARGET pay][ARG1 this] up front when you are ready. take your time would have a low coverage. These are the kind of sentences that we do not want to rely on during the training. This sentence is hard to semantically parse automatically and it is a bit colloquial which makes it a less favorable training instance, especially in a low resource setting where good training instances are hard to obtain. We have also experimented with another version of the coverage, which computes the coverage over the number of all the words instead of all the words that were labelled. The version described in equation (3) slightly outperforms the second model, thus we only report the former.

4.3 Injecting sentence length

The purpose of our experiments is to show that injecting a monolingual semantic based objective function for deriving ITG induction helps learn more semantically correct bilingual correlations. We propose an intuitive approach to evaluate the degree of goodness of sentence pairs based on the sentence length of the English side.

This method simply counts the number of words in a sentence; we then take the reverse sentence length as a confidence-weight. We claim that having long sentences makes the data more sparse when we train on a small corpus. This might prevent the system from efficiently learning from the data and thus hurts the translation quality. The reverse sentence length is calculated as follows:

$$\text{L} = (1 / \text{\# words}) \qquad (3)$$

We experiment this method with the Chinese–English translation task. We show in table 1 that using reverse sentence length as a confidence-weighting measure slightly improves the SMT

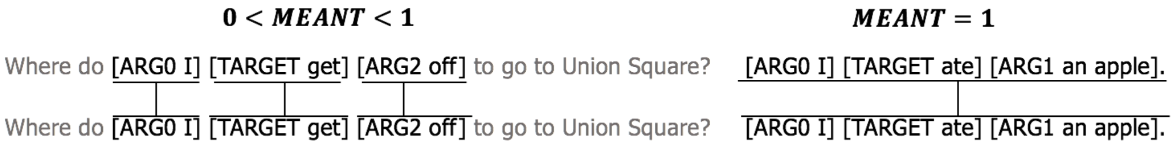

$0 < MEANT < 1$
$MEANT = 1$
Where do [ARG0 I] [TARGET get] [ARG2 off] to go to Union Square?
Where do [ARG0 I] [TARGET get] [ARG2 off] to go to Union Square?
[ARG0 I] [TARGET ate] [ARG1 an apple].
[ARG0 I] [TARGET ate] [ARG1 an apple].

Figure 2: MEANT score in different situations.

Table 1: The monolingual SRL coverage model greatly outperforms the sentence length one.

Alignments	BLEU	TER
GIZA++	19.23	63.40
BITG	20.05	63.19
+Sentence length	20.54	62.49
+SRL$_{en}$	**23.60**	**61.68**

quality in terms of BLEU and TER scores in comparison to GIZA++ and BITG based models. This shows that confidence-weighting the training instances even with a simple measure like sentence length helps improve SMT for low resource languages. However, we note that our monolingual SRL coverage based model substantially improves the translation quality compared to using a simple heuristic such as sentence length.

5 Experimental setup

5.1 Training data

Our experiments aim to show that adopting MEANT as a semantic objective function to bias ITG induction at an early stage the SMT models' training helps reduce the need of extremely large corpora as typically used in SMT training. We focus on the generalization from *only* low resource data and thus focus our work on unpreprocessed data.

Table 2 represents the size of all datasets used in our experimental setup. Except for Chinese and Latvian, which are from IWSLT07 data and Europarl data Koehn (2005) respectively, all the other datasets are from the DARPA LORELEI program. The LORELEI data is diverse; it is composed of forums data and some Quranic verses. The IWSLT07 data is mainly spoken language. The size of the training data varies between 2K (Oromo) and 630K (Latvian) bisentences.

We purposely experiment with different language families including Turkic, Afro-asiatic, Indo-European and Sino-Tibetan languages to show that our approach is not language dependent and can easily be generalized across different languages. We deliberately experiment on a relatively small corpus for the two high-resource languages Chinese and Turkish; all the other languages are considered as low resource languages.

5.2 SMT pipeline

We test the different *alignments* described above using the standard MOSES toolkit (Koehn *et al.*, 2007), and a 6-gram language model learned with the SRI language model toolkit (Stolcke, 2002) trained on the English side of the training data of each language respectively. To tune the loglinear mixture weights, we use *k*-best MIRA (Cherry and Foster, 2012), a version of margin-based classification algorithm and MIRA (Chiang, 2012).

5.3 NMT pipeline

Neural machine translation or NMT has been considered as a hot topic in machine translation over the past few years. NMT is a new encoder-decoder architecture for getting machines to learn to translate based on neural networks. Despite being relatively new, NMT has already shown promising results, achieving state-of-the-art performance for various language pairs (Luong and Manning, 2015; Sennrich *et al.*, 2015; Luong and Manning, 2016). For the sake of comparison, we set up a simple NMT baseline based on Neubig's toolkit lamtram (Neubig, 2015).

5.4 Tuning the hyperparameter for the monolingual SRL coverage based model

For the monolingual SRL coverage model, we tune the hyperparameter β_0 on Uzbek–English and Uyghur–English to find the best value of β_0. We test the model with the obtained hyperparameter with different language pairs. The tuning results are reported in table 3; although the difference in the results between the different values of β_0 is insignificant, we note that β_0=0.7 gives the best results across both language pairs. Therefore, we set

Table 2: The size of the different datasets in sentence pairs (foreign–English).

	Amharic	Chinese	Oromo	Somali	Tigrinya	Turkish	Uyghur	Uzbek	Latvian
Training	60,300	39,953	2,308	50,194	13,807	180,578	97,367	153,408	637,599
Tuning	3,016	1,512	116	2,510	691	1,000	2,000	1,200	2,000
Testing	3,015	489	116	2,510	691	500	1,000	600	2,000

Table 3: Tuning β_0 for the SRL coverage model.

	Uzbek–English		Uyghur–English	
Alignments	BLEU	TER	BLEU	TER
+SRL$_{en}$ 1, β_0=0	18.29	**74.01**	23.67	66.02
+SRL$_{en}$ 1, β_0=0.1	18.14	74.16	23.12	66.42
+SRL$_{en}$ 1, β_0=0.5	18.11	74.18	23.70	65.74
+SRL$_{en}$ 1, β_0=0.7	18.24	**74.03**	**23.85**	**65.57**
+SRL$_{en}$ 1, β_0=1	**18.32**	74.56	23.43	66.75

β_0 to 0.7 in the remaining parts of the paper.

6 Results

Adopting MEANT for confidence-weighting gives the best results for translating low resource languages. We compare the performance of the MEANT and the monolingual English SRL coverage based BITG alignments against the conventional BITG and the traditional GIZA++ alignments. To efficiently assess the quality of our different systems, we evaluate using surface based metrics such as BLEU (Papineni *et al.*, 2002), edit-distance based metrics such as CDER (Leusch *et al.*, 2006), WER (Nießen *et al.*, 2000), PER (Tillmann *et al.*, 1997), TER (Snover *et al.*, 2006) and the semantic evaluation metric MEANT (Lo *et al.*, 2012).

6.1 Adopting MEANT gives the best results across multiple challenging low resource languages

Our experiments show that injecting the monolingual semantic evaluation metric MEANT as a training objective function gives the best results compared to any monolingual confidence-weighting model proposed so far since it consistently improves the translation quality for multiple challenging low resource languages. This can be explained by the fact that XMEANT and MEANT have the same constraints and thus we expect them to have the same behavior.

We note from table 4 that the alignments based on our proposed models (SRL$_{en}$ is the monolingual SRL coverage and SRL$_{MEANT}$ is the MEANT based model) achieve a much higher performance than the traditional GIZA++ and the unbiased BITG baseline across all metrics. The impact of MEANT or SRL coverage on the translation quality depends on the data size and on the nature of the language. Translation tasks like Oromo–English have harsher conditions than the Turkish–English task since Oromo data is harder to obtain. The highest scores that we managed to obtain on Oromo–English are 8.26 for BLEU and 11.33 for MEANT, which reflects the difficulty of the task we study here. In most cases, the difference varies between 2 BLEU points like in Amharic and Uzbek translations to 5 BLEU points like in the Chinese–English translation task. One exception is the Somali–English translation where we only note a small improvement (0.5 BLEU points); the reason is that the test data is too large (2500 sentences) in proportion to the size of the training data. Our methods seem to have a higher impact on error-rate metrics; we improved by around 13 PER points and 6 WER points on the Amharic–English translation task. We also improved semantic SMT by obtaining better MEANT scores on all our SRL based models.

However, the difference between the SRL coverage and the MEANT based models is small. The MEANT based model is better most of the time except for the Uzbek–English translation task, where the SRL coverage model is slightly better in terms of BLEU and TER.

Table 4: Adopting MEANT as a confidence-weighting measure produces the best results across all commonly used metrics.

	Amharic–English					
	MEANT	BLEU	TER	WER	PER	CDER
GIZA++	10.85	11.68	101.85	103.08	90.18	93.72
BITG	10.92	13.00	98.27	101.82	88.10	93.63
+ SRL$_{\text{en}}$	11.57	13.59	98.00	100.31	87.55	92.37
+ SRL$_{\text{MEANT}}$	**12.28**	**14.72**	**92.12**	**94.44**	**77.55**	**86.40**
	Chinese–English					
GIZA++	22.77	19.23	63.40	62.08	55.75	59.79
BITG	23.90	20.05	63.19	61.63	54.07	59.61
+ SRL$_{\text{en}}$	23.99	23.60	61.68	61.90	54.40	59.40
+ SRL$_{\text{MEANT}}$	**24.10**	**24.94**	**60.96**	**61.50**	**54.40**	**59.41**
	Uzbek–English					
GIZA++	14.47	17.09	80.91	87.71	64.61	78.11
BITG	16.55	17.66	78.12	84.60	62.86	75.51
+ SRL$_{\text{en}}$	17.04	**19.07**	**72.56**	78.99	57.34	70.36
SRL$_{\text{MEANT}}$	**17.35**	18.24	74.03	**78.63**	**57.00**	**70.00**
	Oromo–English					
GIZA++	9.59	5.16	134	134	110	124
BITG	10.04	7.80	131	131	113	121
+ SRL$_{\text{en}}$	10.40	7.92	126	129	111	122
SRL$_{\text{MEANT}}$	**11.33**	**8.26**	**123**	**125**	**105**	**119**
	Somali–English					
GIZA++	18.25	19.80	69.00	79.60	56.91	67.66
BITG	18.47	19.85	68.80	79.00	56.72	66.23
+ SRL$_{\text{en}}$	18.59	**20.24**	68.70	78.04	56.62	66.50
SRL$_{\text{MEANT}}$	**18.87**	20.06	**68.50**	**78.00**	**56.42**	**66.20**
	Tigrinya–English					
GIZA++	12.39	11.52	98.44	93.11	77.14	86.43
BITG	14.10	11.75	99.06	93.17	77.19	86.40
+ SRL$_{\text{en}}$	14.90	12.28	94.87	94.49	77.70	87.73
SRL$_{\text{MEANT}}$	**14.93**	**12.85**	**93.52**	**92.94**	**76.50**	**85.90**
	Turkish–English					
GIZA++	14.37	12.72	74.63	81.36	55.86	72.23
BITG	16.24	14.12	74.92	82.23	55.59	72.37
+ SRL$_{\text{en}}$	16.80	14.50	74.50	80.97	**53.78**	70.82
SRL$_{\text{MEANT}}$	**17.62**	**14.95**	**73.12**	**80.83**	54.12	**70.63**

Table 5: NMT models perform worse than SMT models for the Tigrinya–English translation task.

	BLEU	TER
SMT	11.52	98.44
SMT + SRL$_{\text{MEANT}}$	**12.85**	**94.87**
NMT	1.51	118
NMT + SRL$_{\text{MEANT}}$	1.91	99.16

6.2 NMT models are weak when translating low resource languages

Our goal is to investigate apples-to-apples comparison: (a) ability to generalize from *only* low resource data *without* transfer from related high-resource languages, and (b) ability to work with unpreprocessed data. We ran a simple NMT baseline with low resource languages. Neural NLP models in general and neural machine translation models in particular tend to need huge data to work

Table 6: MEANT based models perform well ina high resource setting, but the impact is higher in a low resource setting.

	MEANT	BLEU	TER
GIZA++	19.48	30.13	56.63
BITG	20.35	34.03	50.94
+ $\text{SRL}_{\text{MEANT}}$	**20.43**	**34.27**	**50.35**

properly since it is based on generalization. We use MEANT to confidence-weight the training data for the Tigrinya–English translation task then shuffle the data so that the identical sentence pairs are not in the same batch. Table 5 shows that the SMT model highly outperforms the NMT model for both the unbiased models and the MEANT constrained models. The results might seem very low for an NMT model, but, we highlight the point that to maintain the apples-to-apples low-resource generalization comparison we are using raw data without any preprocessing and without any additional high-resource dependent techniques like knowledge transfer from similar high-resource languages.

6.3 Our models also perform well in a high resource setting

We tested the MEANT based model with Latvian–English translation task (results in table 6), which is not low resource in this case since it has more than 600K sentence pairs. Table 6 shows that our approach slightly improves the translation quality compared to BITGs, but highly outperforms GIZA++ based model. This shows that, although our novel approach improves the MT quality in a high resource setup, it definitely has a higher impact when dealing with low resource languages.

6.4 Translation examples

In example 1 (figure 3), the MEANT based model produces a translation that is as good as the reference. However, both BITG and GIZA++ based translations completely fail to capture the word opera. Example 2 (figure 3) is from the Turkish–English translation task. In this example, the MEANT based model only fails at translating the name of the city Belede; otherwise, the translation sounds better than the two other systems. The BITG model output has Yangon, which does not appear in the Turkish input (see gloss).

7 Conclusion

We have shown that adopting the monolingual semantic evaluation metric MEANT as an objective function for driving ITG induction yields a high improvement compared to the conventional alignment methods on many challenging low resource languages. We have also proposed another heuristic for evaluating how good an English semantic parse is, then used it to induce ITGs. We have experimented with several challenging low resource languages from different language families and have demonstrated that using a monolingual semantic frame based objective function during the actual learning of the translation model helps learn good bilingual correlations with a relatively small dataset in contrast to conventional SMT systems. The promising results we report in this new line of research make it seem that learning more semantically motivated translation models might be less challenging than generally assumed and is worth exploring.

8 Acknowledgment

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under LORELEI contract HR0011-15-C-0114, BOLT contracts HR0011-12-C-0014 and HR0011-12-C-0016, and GALE contracts HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the Horizon 2020 grant agreement 645452 (QT21) and FP7 grant agreement 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF16210714, GRF16214315, GRF620811 and GRF621008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

References

Karteek Addanki, Chi-kiu Lo, Markus Saers, and Dekai Wu. LTG vs. ITG coverage of cross-lingual verb frame alternations. In *16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, Trento, Italy, May 2012.

Wilker Aziz, Miguel Rios, and Lucia Specia. Shallow semantic trees for SMT. In *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, 2011.

Meriem Beloucif and Dekai Wu. Driving inversion transduction grammar induction with semantic evaluation. In *5th Joint Conference on Lexical and Computational Semantics at ACL*, 2016.

Example 1		**Example 2**	
Input	在 这儿 能 买到 歌剧 的 票 吗 ?	**Input**	depremin merkez üssünün Belede kenti olduğu belirtildi
Gloss	at here can buy opera ticket ?	**Gloss**	earthquake's center base of Belede city of happened stated
Reference	can I get an opera ticket here ?	**Reference**	the earthquake's epicenter is reported to have been the city of Belede
GIZA++	where can I buy tickets for " The here ?	**GIZA++**	the it was reported that the epicenter of the city
BITG	where can I buy a ticket for the here ?	**BITG**	the epicenter of the earthquake was in the city of Yangon
MEANT based	where can I buy a ticket for the opera here ?	**MEANT based**	the epicenter of the earthquake was reported to be the base of the city

Figure 3: Examples comparing the output from the three discussed alignment systems extracted from the Chinese–English and the Turkish–English translation tasks.

Meriem Beloucif and Dekai Wu. Improving word alignment for low resource languages using english monolingual srl. In *Sixth Workshop on Hybrid Approaches to Translation (HyTra-6) at COLING. Osaka, Japan*, 2016.

Meriem Beloucif, Markus Saers, and Dekai Wu. Improving semantic smt via soft semantic role label constraints on itg alignments. In *Machine Translation Summit XV (MT Summit 2015)*, pages 333–345, Miami, USA, October 2015.

Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 43–48, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*. Association for Computational Linguistics, 2012.

David Chiang. Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 13:1159–1187, April 2012.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Interactive Poster and Demonstration Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 177–180, Prague, Czech Republic, June 2007.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *The Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand, September 2005.

Mamoru Komachi, Yuji Matsumoto, and Masaaki Nagata. Phrase reordering for statistical machine translation based on predicate-argument structure. In *International Workshop on Spoken Language Translation (IWSLT 2006)*, 2006.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. CDer: Efficient MT evaluation using block movements. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.

Ding Liu and Daniel Gildea. Semantic role features for machine translation. In *23rd International Conference on Computational Linguistics (COLING 2010)*, 2010.

Chi-kiu Lo and Dekai Wu. MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, 2011.

Chi-kiu Lo and Dekai Wu. SMT vs. AI redux: How semantic frames evaluate MT more accurately. In *Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, 2011.

Chi-kiu Lo and Dekai Wu. Unsupervised vs. supervised weight estimation for semantic MT evaluation metrics. In *Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6)*, 2012.

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. Fully automatic semantic MT evaluation. In *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

Chi-kiu Lo, Meriem Beloucif, Markus Saers, and Dekai Wu. XMEANT: Better semantic MT evaluation without reference translations. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, 2014.

Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *The International Workshop on Spoken Language Translation (IWSLT15)*, 2015.

Minh-Thang Luong and Christopher D Manning. Achieving open vocabulary neural machine translationwith hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 1054–1063, 2016.

Graham Neubig. lamtram: A toolkit for language and translation modeling using neural networks, 2015.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. A evaluation tool for machine translation: Fast evaluation for MT research. In *The Second International Conference on Language Resources and Evaluation (LREC 2000)*, 2000.

Franz Josef Och and Hermann Ney. Improved statistical alignment models. In *The 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, pages 440–447, Hong Kong, October 2000.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318, Philadelphia, Pennsylvania, July 2002.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. Shallow semantic parsing using support vector machines. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, 2004.

Michael Roth and Kristian Woodsend. Composition of word representations improves semantic role labelling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 407–413, Doha, Qatar, October 2014. Association for Computational Linguistics.

Markus Saers and Dekai Wu. Improving phrase-based translation via word alignments from stochastic inversion transduction grammars. In *Third Workshop on Syntax and Structure in Statistical Translation (SSST-3)*, pages 28–36, Boulder, Colorado, June 2009.

Markus Saers and Dekai Wu. Reestimation of reified rules in semiring parsing and biparsing. In *Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-5)*, pages 70–78, Portland, Oregon, June 2011. Association for Computational Linguistics.

Markus Saers, Joakim Nivre, and Dekai Wu. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *11th International Conference on Parsing Technologies (IWPT'09)*, pages 29–32, Paris, France, October 2009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. volume abs/1508.07909, 2015.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *7th Biennial Conference Association for Machine Translation in the Americas (AMTA 2006)*, pages 223–231, Cambridge, Massachusetts, August 2006.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing (ICSLP2002 - INTERSPEECH 2002)*, pages 901–904, Denver, Colorado, September 2002.

Christoph Tillmann, Stephan Vogel, Hermann Ney, Arkaitz Zubiaga, and Hassan Sawaf. Accelerated DP based search for statistical translation. In *Fifth European Conference on Speech Communication and Technology (EUROSPEECH 1997)*, 1997.

Dekai Wu and Pascale Fung. Semantic roles for SMT: A hybrid two-pass model. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 13–16, 2009.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. Extracting preordering rules from predicate-argument structures. In *The 5th International Joint Conference on Natural Language Processing (IJCNLP2011)*, 2011.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Deyi Xiong, Min Zhang, and Haizhou Li. Modeling the translation of predicate-argument structure for SMT. In *50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, 2012.

M3TRA: integrating TM and MT for professional translators

Bram Bulté
CCL – KU Leuven

Tom Vanallemeersch
CCL – KU Leuven

Vincent Vandeghinste
CCL – KU Leuven

firstname.lastname@ccl.kuleuven.be

Abstract

Translation memories (TM) and machine translation (MT) both are potentially useful resources for professional translators, but they are often still used independently in translation workflows. As translators tend to have a higher confidence in fuzzy matches than in MT, we investigate how to combine the benefits of TM retrieval with those of MT, by integrating the results of both. We develop a flexible TM-MT integration approach based on various techniques combining the use of TM and MT, such as fuzzy repair, span pretranslation and exploiting multiple matches. Results for ten language pairs using the DGT-TM dataset indicate almost consistently better BLEU, METEOR and TER scores compared to the MT, TM and NMT baselines.

1 Introduction

While software for professional translators has included translation memories (TMs) since several decades, especially in the context of specialized documents, the use of machine translation (MT) in such software is more recent. Even though certain commercial translation tools now offer functionalities such as automatic fuzzy match repair, TM and MT technologies are often still used independently, i.e. either a match for a query sentence or an MT output is provided. This is not ideal, as translators tend to have a higher confidence in 'human' TM than in MT. It has to be kept in mind, however, that only exact matches provide a translation of the query sentence; 'fuzzy' matches offer a translation of a similar sentence. In contrast, MT systems provide a translation for any sentence, but they have problems with a number of, often linguistic, issues, such as complex morphological phenomena, long distance dependencies and word order (Bisazza and Federico, 2016; Sudoh et al., 2010). We investigate how to combine the confidence in fuzzy match retrieval with full sentence translation by integrating TM and MT output. We develop M3TRA,[1] a method which performs a TM match preprocessing step before running a standard phrase-based statistical MT (PBSMT) system trained on the TM. M3TRA combines different approaches, and is flexible in several respects: it applies various fuzzy match score thresholds, allows for more than one match to be used per query sentence, and can use several fuzzy metrics. It comprises two main components: (a) **fuzzy repair**, automatically editing high-scoring fuzzy matches, and (b) **span pretranslation**, constraining MT output by including certain consistently aligned spans of one or more TM matches.

We perform tests on ten language pairs which involve multiple language families, using the DGT-TM dataset (Steinberger et al., 2013). We apply PBSMT without span pretranslation as a baseline, as well as 'pure' TM and a standard NMT system, and evaluate the translations using several metrics. M3TRA is integrated in a prototype translation interface providing translators with more 'informed' MT output (Coppers et al., 2018).

The following sections describe the research context, system architecture, experimental design and results. The final sections contain a discussion, overview of work in progress and conclusions.

[1] MeMory + Machine TRAnslation

2 Research context

The baseline approach to TM-MT integration uses MT to translate a query sentence in case no sufficiently similar translation unit is found in the TM (Simard and Isabelle, 2009). This can be augmented by using an estimation of the *usefulness* of MT and TM output (He, 2011). Other studies focus on correcting close matches from a TM using PBSMT, based on a set of learned edit operations (Hewavitharana et al., 2005). Ortega et al. (2016) propose a *patching approach* to correct TM matches with any kind of SMT system, and Espla-Gomis et al. (2015) a more translator-oriented method that offers *word keeping* recommendations based on information coming from an MT system. Example-based MT systems have also been used to leverage sub-segmental TM data (Simard and Langlais, 2001).

Of particular relevance are approaches that constrain a PBSMT system to use relevant parts of a fuzzy match (Zhechev and Van Genabith, 2010), for example by adding XML markup to Moses input (He, 2011; Koehn and Senellart, 2010; Ma et al., 2011) or by using a constrained word lattice (Li et al., 2016). Related to these are methods that augment the translation table of a PBSMT system with aligned spans from a retrieved TM match, yet without forcing the SMT system to incorporate (parts of) these aligned spans (Bicici and Dymetman, 2008; Simard and Isabelle, 2009). Alternatively, information from the fuzzy matches can also be integrated in the SMT system itself (Wang et al., 2013), for example using *sparse features* (Li et al., 2017). Recent studies focus on how to leverage TM information for NMT systems. These approaches work, for example, by imposing lexical constraints on the search algorithms used by NMT (Hokamp and Liu, 2017), by augmenting NMT systems with an additional lexical *memory* (Feng et al., 2017), or by explicitly providing the NMT system with access to retrieved TM matches (Gu et al., 2017).

M3TRA combines different elements from these approaches, which is its main novelty. In this paper we focus on (a) repairing close fuzzy matches, and (b) augmenting the MT input with information derived from the parallel corpus (the TM) used to train the MT system, thus constraining the translation of certain (parts of) sentences. We use a PBSMT system as basis for TM-MT integration because SMT allows a straightforward application of pretranslation (e.g. explicit alignment information is used in the process).

3 System architecture

M3TRA consists of four components: (a) a TM system, (b) a PBSMT engine, (c) a system for fuzzy repair (FR) and (d) a system for pretranslation span search (PSS). We elaborate on each of these components in the following sections. The sentence to translate can follow a number of routes, depending on the fuzzy match score of the best retrieved match and the success or failure of certain attempted operations (see Figure 1). First, FR is attempted for sentences that have at least one match which meets the relevant threshold (θ_{FR}). If FR is performed, it may modify the translation of the fuzzy match by deleting, inserting or substituting words. In case FR is not performed or fails, there are three options: (a) if the score of the highest match satisfies the TM threshold (θ_{TM}), the translation of the TM match becomes the final output, (b) if the score is between the TM and MT thresholds, PSS is attempted, and (c) if the score is below the MT threshold (θ_{MT}), or PSS fails (i.e. the query sentence as such becomes input to MT), the 'pure' MT output is used as final output.

Each of the four M3TRA components is described in detail below, followed by an overview of the parameter tuning process.

3.1 Translation Memory System

The TM is defined as a set $\mathcal{M}$ consisting of tuples of source and target sentences (s, t), i.e. translation units. Let q be the sentence to be translated (query sentence). It is looked up in the TM using a similarity function Sim, according to Equation 1, resulting in a set $\mathcal{M}_q$ of translation units the source sentence s of which is sufficiently similar to q, according to threshold θ_{Sim}. The best match for q is determined according to Equation 2.[2]

$$\mathcal{M}_q = \{(s,t) \in \mathcal{M} : Sim(q,s) \geq \theta_{Sim}\} \quad (1)$$

$$(s_b, t_b) = \underset{(s,t) \in \mathcal{M}_q}{\arg\max}\, Sim(q,s) \quad (2)$$

Matches are retrieved from the TM using two different similarity metrics: Levenshtein distance (Levenshtein, 1966) and METEOR (Lavie

[2] In case there are several matches with the same score, the first match encountered in the TM is taken as best match.

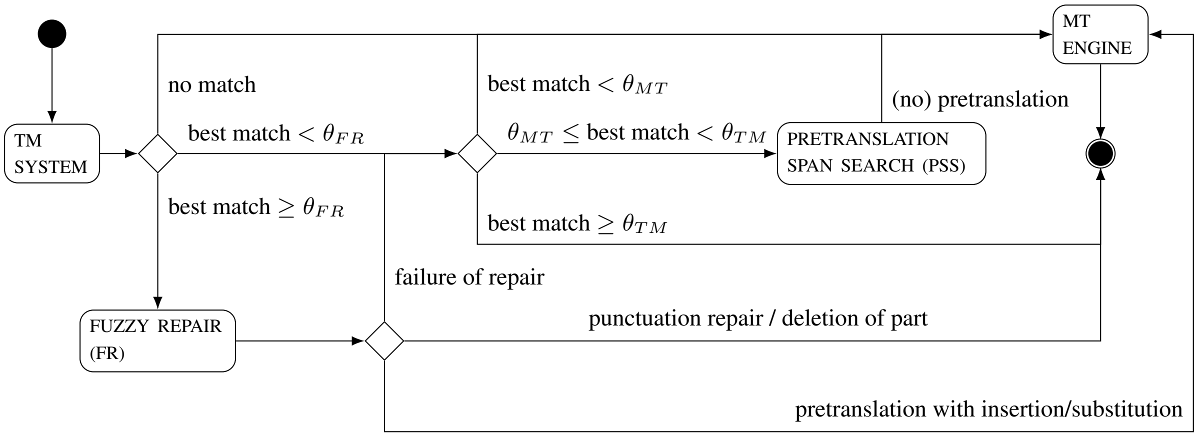

MT ENGINE
no match
best match < θ_{MT}
(no) pretranslation
TM SYSTEM
best match < θ_{FR}
$\theta_{MT} \leq$ best match < θ_{TM}
PRETRANSLATION SPAN SEARCH (PSS)
best match ≥ θ_{FR}
best match ≥ θ_{TM}
failure of repair
FUZZY REPAIR (FR)
punctuation repair / deletion of part
pretranslation with insertion/substitution

Figure 1: M3TRA workflow

and Agarwal, 2007). We limit the size of $\mathcal{M}_q$ to n, i.e. we only keep the tuples with the n best matches (plus any additional tuples with matches that have the same score as the nth best match).

As shown in Figure 1, we compare $Sim(q, s_b)$ to thresholds like θ_{FR} to decide whether to send q to FR or to PSS.

3.2 MT engine

We train a Moses PBSMT system (Koehn et al., 2007) from the TM sentence pairs.[3] We build a 5-gram KenLM language model, set the distortion limit to 6, and apply a maximal phrase length of 7.[4] During decoding, we set the maximum phrase length to 100. This is necessary to be able to pretranslate long word sequences using XML markup. The GIZA++ word alignment (using the grow-diag-final heuristic), the lexical probabilities and the principle of consistently aligned spans (Koehn, 2009) based on which the Moses phrase table is constructed are also used in the FR and PSS components (with an additional constraint, as explained later on).

3.3 Fuzzy repair

Let $\mathcal{M}_{FR}$ be the set of high-scoring translation units retrieved for q.[5] Three types of editing operations are attempted to arrive at the final output o: *substitution*, *deletion* and *insertion*. First, however, a number of specific operations aimed at repairing punctuation are performed.

Punctuation repair: since (simple) punctuation is arguably different from other linguistic phenomena, it is tackled by a dedicated subcomponent. We rank the tuples $(s, t) \in \mathcal{M}_{FR}$, according to $Sim(q, s)$, and iterate through the ranked list in order to verify whether simple punctuation issues can be resolved to produce o:

- if the only difference between q and s is due to casing, or one additional comma, we consider them as identical sentences, and set o to t; hence, we could say this is a type of 'void' repair;
- if q ends in punctuation,[6] and both s and t do not, we set o to t followed by the corresponding punctuation; if, however, t already contains punctuation in final position, we set o to t (another type of 'void' repair);
- if s and t end in punctuation, and q does not, we set o to t minus the final punctuation.

We stop iterating as soon as we produced o. In case of failure, we look at the more general mechanisms of substitution (sub), deletion (del) and insertion (ins). Since both del and ins can be considered more specific versions of sub (i.e. replacement of a part of s or t by the empty string), we focus on sub first.

Substitution: the basic idea behind the sub operation is to translate non-matching tokens of q and s in the context of tokens in t. sub is attempted when both q and s contain one sequence of one or more unmatched tokens q_i^j and $s_i^{j'}$ that end at potentially different positions j and j'. We check whether $s_i^{j'}$ is consistently aligned to a sequence

[3] Minus the development set used for tuning the parameters.

[4] These are 'default' settings.

[5] To limit potential negative effects of erroneously aligned translation units, $\mathcal{M}_{FR}$ is filtered by imposing a threshold on the percentage of aligned source tokens per translation unit.

[6] One of the tokens . , ? ! : ; -

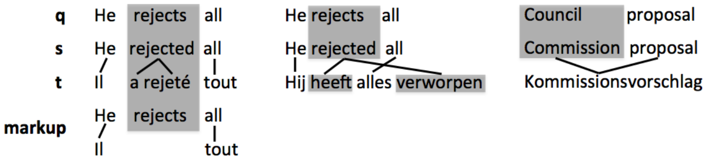


Figure 2: Examples of (attempted) substitution

t_k^l, i.e. whether each token in $s_i^{j'}$ is either aligned to a token in t_k^l or unaligned, and vice versa.[7] In addition, we impose the condition that the first and last token of $s_i^{j'}$ be aligned; the same goes for the first and last token of t_k^l. We assume that an alignment satisfying this condition, which we will call a *border-link alignment* in the remainder of this article, increases the likelihood of translation equivalence between sequences.

The sub operation is illustrated by the simplified examples in Figure 2. In the first example, both q and s contain a one-word sequence that is not shared (*rejects* and *rejected* respectively). In both cases, this sequence starts at the second position. The word *rejected* is aligned with the adjacent French target tokens *a* and *rejeté*, which in turn are only aligned with *rejected*. This allows for translating *rejects* in the context of *Il* and *tout*. In the second example, substitution fails since *rejected* is aligned with two Dutch target words, *heeft* and *verworpen*, which do not form an uninterrupted sequence. In the third example, substitution is impossible: $s_i^{j'}$ consists of *Commission*, which is aligned with *Kommissionsvorschlag*, while the German word is aligned with both *Commission* and *proposal*, the latter word not being part of $s_i^{j'}$.

To translate a span of q in the context of tokens of t, we proceed as follows. We block all retained tokens from t as pretranslation, by annotating q_1^{i-1} with the tokens of t_1^{k-1} using XML markup (unless $i = 1$), and annotating q_{j+1}^{v} with the tokens of t_{l+1}^{w}, unless j equals v; v and w stand for the number of tokens in q and t. The annotated q is then sent to the MT system, which translates q_i^j in the context of t_1^{k-1} and/or t_{l+1}^{w} (*Il* and *tout* in Figure 2).

To verify multiple potential substitutions, a *sliding window* is applied by a stepwise decrease of i and increase of j and j'. Each o resulting from a successful substitution is scored using the language model of the PBSMT system, in order to pick the best alternative o. The size of the sliding window is a model parameter. Two additional parameters[8] are put in place to limit the applicability of sub operations: a threshold for the maximum length of the span t_k^l and one for the maximum percentage of unaligned tokens within that span.

Deletion: the del operation consists of removing a sequence from t to yield o. If s is identical to q, apart from one additional sequence s_i^j (which may be a prefix, infix or suffix of s), and the latter has a border-link alignment with a target sequence t_k^l, the target sequence can be deleted. Two safeguard rules control the modification. If the token t_{k-1} is not aligned with a token in s, it is also deleted. The second rule is optional and ensures that t_k^l is not removed if it consists of only one token with less than 4 characters; [9] this leads o to be equal to t, which is another instance of 'void' repair.

The two safeguard rules are illustrated in Figure 3. In the leftmost example, the first occurrence of the Dutch word *de*, which precedes the sequence identified for deletion, is not aligned with any token in s. It is therefore also deleted. The rightmost example shows that the only difference between q and s is the token *the*, which has less than 4 characters. t is thus left unchanged.

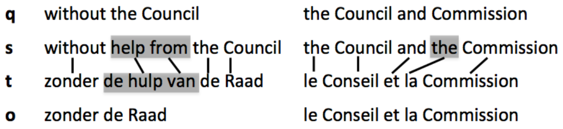

q
s
t
o
without the Council
without help from the Council
zonder de hulp van de Raad
zonder de Raad
the Council and Commission
the Council and the Commission
le Conseil et la Commission
le Conseil et la Commission

Figure 3: Examples of (attempted) deletion

Insertion: the ins operation can be performed when q is identical to s, apart from a sequence q_i^j (which may be a prefix, infix or suffix of q). Key to ins is determining where to insert the translation of q_i^j in t. For this to be possible, all of the following conditions need to be satisfied: (a) the token s_{i-1} is aligned to one or more tokens, the rightmost of which we call t_k, (b) s_i is aligned to

[7] With the understanding that at least one token in $s_i^{j'}$ is aligned.

[8] Added after a qualitative analysis of development set output.

[9] This heuristic was implemented to deal with articles in particular, in the absence of part-of-speech information.

one or more tokens, the leftmost of which we call t_l, and (c) k and l are adjacent (i.e. $l = k + 1$). If we found the insertion position k, we annotate q_1^{i-1} with the tokens in t_1^k, and annotate q_{j+1}^v with the tokens in t_{k+1}^w. This is illustrated in Figure 4. q contains an additional sequence compared to s (*European*), starting at the second position. We verify with which German word the first source token (s_{i-1}, *the*) is aligned, and with which word the second source token (*Parliament*) is aligned. As the aligned German words are adjacent, *the* can be annotated with *das* and *Parliament* with *Parlament*.

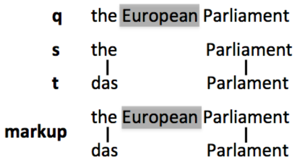


Figure 4: Example of insertion

If i is 1 (i.e. the non-matching part q_i^j is the prefix of the sentence), we apply a different procedure. If token s_1 is aligned with one or more target tokens, we annotate the sequence q_{j+1}^v with t_k^w, k being the position of the leftmost aligned token. If j is v (i.e. the non-matching part is the suffix of the sentence), and the last token of s is aligned to one or more target tokens, we annotate the sequence q_1^{i-1} with t_1^k, k being the position of the rightmost aligned token.

For any q that is not repaired and for which $Sim(q, s_b) \geq \theta_{TM}$, we set o to the most frequent t_b. Otherwise, q is sent to PSS.

3.4 Pretranslation span search

PSS consists of annotating (pretranslating) spans of q based on matches in $\mathcal{M}_q$, and subsequently constraining the MT system to respect the translations of these spans while producing o. PSS is applied in case the following condition is satisfied: $\theta_{MT} \leq Sim(q, s_b) < \theta_{TM}$ (see Figure 1). If so, a subset $\mathcal{M}_p$ is established according to Equation 3.

$$\mathcal{M}_p = \{(s,t) \in \mathcal{M}_q : Sim(q,s) \geq \theta_{PSS}\} \quad (3)$$

Based on the sentence pairs in $\mathcal{M}_p$, we define another set $\mathcal{P}_q$, which contains pretranslation tuples $(s, t, i, j, i', j', k, l)$. These are tuples for which all of the following conditions are valid: (a) the sentence pair belongs to $\mathcal{M}_p$, (b) q_i^j matches the source span $s_{i'}^{j'}$[10] and (c) $s_{i'}^{j'}$ has a border-link alignment with the target span t_k^l. A specific pair of source and target span may occur in multiple sentence pairs (see the frequency check below). Some of the tuples in $\mathcal{P}_q$ will be used for pretranslation, as described below.

Filtering pretranslation tuples: a tuple $p \in \mathcal{P}_q$ is filtered out if it satisfies one of the following conditions: (a) given all tuples $\mathcal{P}'_q \subseteq \mathcal{P}_q$ that involve the sentence pair of p, the total length of the source and target spans in $\mathcal{P}'_q$ does not satisfy a minimum length, (b) the length of the source and/or target span in p does not satisfy a minimum value, (c) the source and/or target span in p do not contain any content word (i.e. noun, adjective, verb or adverb), (d) the percentage of words aligned between the source and target span in p is too low, or (e) the one-to-many alignment score of p, defined in Equation 4, is too low. In this equation, y_x represents the number of tokens aligned to s_x, a token in the source span $s_{i'}^{j'}$ of p.

$$\frac{1}{j-i+1} \sum_{x=i}^{j} \frac{1}{y_x} \quad (4)$$

Combining pretranslation tuples: after filtering, each tuple $p \in \mathcal{P}_q$ is scored according to the weighted sum of (a) the length of the target span, (b) the frequency of the pair of source and target span, i.e. the number of tuples in $\mathcal{P}_q$ in which the pair occurs, and (c) the maximal fuzzy match score for the span pair, i.e. the maximal similarity $Sim(q, s)$ for all tuples in which the span pair occurs. The weights of the three above factors are model parameters. Subsequently, the tuples are ranked according to score, and used in the following iterative procedure. The spans of the first ranked tuple are used for pretranslation, i.e. the span t_k^l is used to annotate the q_i^j span. This tuple is removed from $\mathcal{P}_q$. The system then looks for the first ranked tuple in which the q_i^j span does not overlap with the already annotated span of q. This process is repeated until $\mathcal{P}_q$ only contains tuples with overlapping spans, or until the threshold for number of annotations has been reached. Figure 5

[10] Matching q to s given some similarity function leads to the identification of a number of matching parts. These parts are typically sequences which are identical in q and s. A matching span q_i^j refers to such a matching part, or one of its prefixes, infixes or suffixes. For instance, if two sentences have a matching part *The EC was*, matching spans include *The EC was*, *The EC*, *EC* etc.

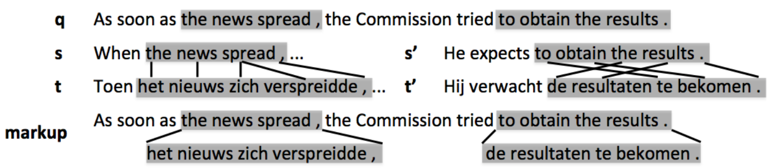

q As soon as the news spread , the Commission tried to obtain the results .
s When the news spread , ...
s' He expects to obtain the results .
t Toen het nieuws zich verspreidde , ...
t' Hij verwacht de resultaten te bekomen .
markup As soon as the news spread , the Commission tried to obtain the results .
het nieuws zich verspreidde ,
de resultaten te bekomen .

Figure 5: Example of pretranslation span search

provides an example of how two non-overlapping spans of a query sentence (*the news spread ,* and *to obtain the results .*) are pretranslated by two Dutch target spans (*het nieuws zich verspreidde ,* and *de resultaten te bekomen .*) originating from two different translation units. The PBSMT system is constrained to use these target spans in its final output.

3.5 Parameter setting and tuning

Many of M3TRA's components involve parameters (such as θ_{FR}) that can either be manually fixed or whose optimal value can be determined on the basis of an automated parameter tuning process. Initial tests were run on subsets of the development sets using random parameter initializations. Manual spot-checks of system outputs with different configurations were performed to verify the quality of the resulting translations (in comparison to pure MT output). To make the spot checks potentially more informative, differences in METEOR scores (compared to the MT baseline) were used as a criterion to select sentences with pretranslations that either led to large gains in translation quality or that appeared to result in worse translations.

In addition, a local hill-climbing algorithm was used to help determine the best parameter settings. The methodology followed here involved a stepwise narrowing of the search interval per parameter based on a combination of random initializations and runs of the hill-climber (with increasingly small step size). BLEU scores (Papineni et al., 2002) were used as tuning criterion.

4 Experimental design

This section describes the empirical tests that were carried out. We first describe the dataset and evaluation procedures, before turning to the results.

4.1 Data

We use the TM of the Directorate-General for Translation of the European Commission (Steinberger et al., 2013), for 5 language pairs in 2 directions: EN ↔ NL, FR, DE, HU, PL.[11] To ensure consistency, we only use the cross-section of each of these datasets, resulting in 1.6 million sentence pairs per language combination. 2000 sentence pairs are set aside for development, and the test set consists of 3207 sentences.[12] We tokenized and lowercased all sentences before training Moses and tuning its parameters.

Table 1 shows the percentage of q's categorised on the basis of $Sim(q, s_b)$. For only 5 to 7% of q's no match is found in the TM. For the majority a match below 70% is retrieved, but for around 28-35% a high-scoring match ($> 70\%$) exists.

	None	<70	70-79	80-89	90-99
EN	5.9%	59.0%	9.4%	13.6%	12.1%
NL	5.0%	62.5%	8.9%	11.4%	12.3%
PL	6.7%	64.5%	8.0%	12.1%	8.7%
DE	6.3%	62.9%	9.6%	12.0%	9.2%
FR	4.5%	67.2%	9.3%	11.2%	7.8%
HU	6.6%	64.8%	8.7%	11.1%	8.9%

Table 1: Percentage of test sentences per match range

4.2 Baseline systems

We use three baselines to compare M3TRA with: (a) 'pure' TM matching, which involves selecting the (most frequent) t_b for q as o,[13] (b) the 'pure' Moses PBSMT system, and (c) a *standard* neural translation model.

For the neural MT model, we use OpenNMT (Klein et al., 2017) with default settings, i.e. a seq2seq RNN model with global attention consisting of 50000 words on the source as well as the target side, word embeddings of 500 dimensions, a hidden layer of 500 LSTM nodes, and learning through stochastic gradient descent with a learning rate of 1, and we ran the model for 20 epochs. We chose the best performing model, selected using a development set (different from the validation set)

[11] Note that the original source language may differ and that not all EC documents are translated directly.

[12] We were strict in filtering the test sets: any q for which a 100% match existed in any source language was left out for all language pairs.

[13] If no match is found in the TM, no translation is provided.

	EN-NL	**EN-PL**	**EN-DE**	**EN-FR**	**EN-HU**	**NL-EN**	**PL-EN**	**DE-EN**	**FR-EN**	**HU-EN**
θ_{TM}	0.79	0.87	0.79	0.83	0.70	0.79	0.93	0.71	0.72	0.70
θ_{FR}	0.77	0.63	0.55	0.54	0.39	0.52	0.57	0.53	0.49	0.40
Min % aligned tok FR	0.83	0.85	0.63	0.63	0.65	0.70	0.64	0.66	0.66	0.50
Window shift L	2	2	2	2	1	1	2	1	3	4
Window shift R	0	3	3	2	1	1	0	2	3	1
Max % non-aligned tok FR	0.50	0.42	0.74	0.24	0.72	0.53	0.75	0.48	0.44	0.67
θ_{PSS}	0.48	0.45	0.43	0.73	0.45	0.50	0.69	0.52	0.24	0.35
Min span length PSS	4	6	4	12	4	8	9	5	9	3
Min % aligned tok PSS	74	67	67	56	75	53	58	76	55	62
Min alignment score PSS	0.83	0.64	0.62	0.79	0.64	0.64	0.59	0.55	0.78	0.71

Table 2: Parameter settings after tuning

which was evaluated on BLEU, TER (Snover et al., 2006) and METEOR. The model that scored best on the majority of the metrics was chosen. When all three metrics differ, we chose the best scoring model according to BLEU.

4.3 Evaluation

BLEU scores are used as main evaluation criterion.[14] In addition, we report TER and METEOR scores to verify whether related yet different metrics point to similar trends. We only use one reference translation. To verify whether differences in BLEU scores between the baselines and M3TRA are statistically significant, we use the bootstrap resampling method described by Koehn (2004).

5 Results

5.1 Tuning

Table 2 provides an overview of the parameter settings that were found to lead to the highest BLEU scores on the development sets. We retained ten free parameters, the others were either fixed at certain values or disabled.[15] The results for METEOR as a fuzzy metric were found to be similar to the results using Levenshtein. For the current study, we decided to continue with Levenshtein as metric.

Looking more closely at the retained parameter settings, some observations can be made. First, θ_{TM} varies between 0.70 and 0.93. Second, the value of θ_{FR} lies between 0.39 and 0.77. Third, for any language pair at least half of the source tokens in a translation unit need to be aligned to perform FR. Fourth, for all language pairs, working with a sliding window for substitution was beneficial. Fifth, between 3 and 12 tokens per span are needed to provide beneficial pretranslations. Sixth, imposing restrictions on alignments proved to be positive for translation quality. Finally, the imposed threshold for minimum percentage of aligned words at source side varied between 50 and 83$\%$.

5.2 Tests

Table 3 provides an overview of the evaluation scores for the ten language combinations of M3TRA compared to three baselines: pure TM, pure SMT, and NMT. For 9 of the 10 language combinations, M3TRA scores significantly better than the best baseline (SMT) in terms of BLEU. The increase in BLEU varies between 0.2 (for EN-PL; non-significant difference) and 5.47 points (for EN-HU). METEOR scores actually decrease for FR-EN, and are practically unchanged for EN-PL (+0.06). For EN-HU they increase with 3 points. TER scores consistently decrease for all language pairs. The decrease lies between 0.25 points (for EN-PL) and 5.33 points (EN-HU). Compared to the baseline SMT system, M3TRA affects between 9 and 39$\%$ of the sentences in the test set.

Looking at BLEU (see also Figure 6), baseline SMT also consistently outperforms baseline NMT, with the exception of EN-HU. With TER as evaluation criterion, NMT scores better for EN-HU and FR-EN. In terms of METEOR, SMT consistently outperforms baseline NMT. The quality of pure TM is estimated to be the lowest for all language pairs, which is not surprising, since e.g. a q for which $\mathcal{M}_q$ is empty is left untranslated.

Figure 7 presents the performance of the different systems for different subsets defined on the basis of $Sim(q, s_b)$ for one language pair (DE-EN).[16] With $Sim(q, s_b)$ below 70$\%$, M3TRA does not lead to better scores compared to SMT. Pure

[14] We acknowledge that using BLEU is not ideal, especially when comparing SMT and NMT (Shterionov et al., 2017).

[15] θ_{Sim} = 0.2; n-best matches = 15; PSS weights: length = 0; frequency = 0.83; match score = 0.17.

[16] For reasons of space we restrict ourselves to one language pair. For the other languages, similar trends are observed.

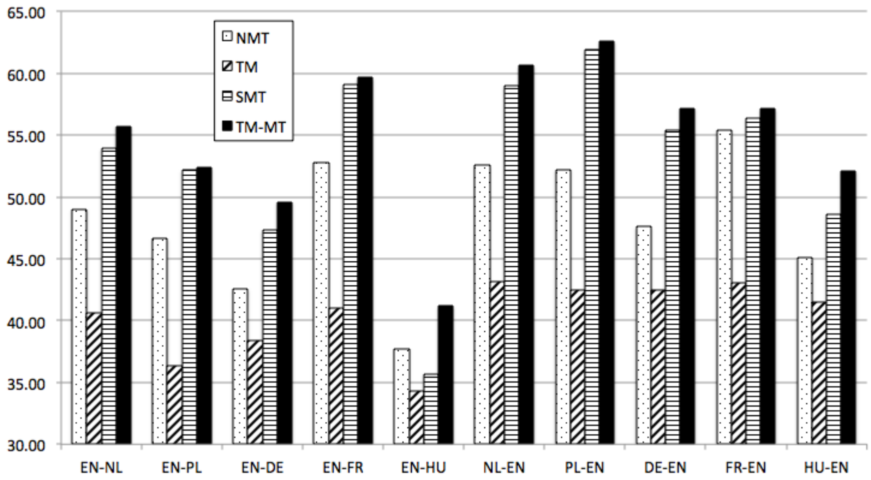


Figure 6: Overview BLEU scores

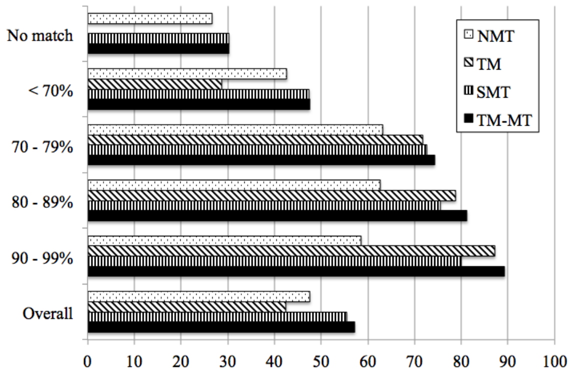


Figure 7: BLEU scores per match range (DE-EN)

TM starts scoring better than SMT in the range 80-89%. Thanks to FR, M3TRA also outperforms pure TM in the two highest match ranges.

6 Discussion

The main novelty of M3TRA is in its adaptable parameters, threshold values and safeguards, as well as in its combination of various features that are present in a number of approaches described in Section 2. Most notably, the use of XML markup to add pretranslation spans to input sentences is also used by He (2011), Koehn and Senellart (2010) and Ma et al. (2011). In M3TRA, Moses is constrained to include these pretranslated spans in the final output (the so-called exclusive mode is used). The fuzzy repair feature is closely related to the work of Ortega et al. (2016). Also the option to simply use TM target matches above a certain match score threshold has been implemented before (Simard and Isabelle, 2009). Moreover, by making use of the information obtained during the alignment process, M3TRA can be adapted easily to provide translators with information on the origin of parts of the proposed translations, possibly indicating which sentences should most likely be post-edited (Espla-Gomis et al., 2015). Finally, the combination of information from different fuzzy matches is also present in previous research (Wang et al., 2013; Li et al., 2016).

The test results show that integrating TM with MT can lead to better MT output, provided that sufficient high-scoring matches are retrieved from the TM. We argue that M3TRA is especially beneficial in a context with enough repetition and where the focus is (at least to a certain extent) on consistency and formulaic language use. Looking at the results for the different language pairs, the potential for improvement is highest for EN-HU and HU-EN,[17] which is most likely due to the (morphological) structure of the Hungarian language and its associated problems for (S)MT. The significant improvements for almost all language combinations indicate that M3TRA potentially works with different language families (Germanic, Romance, Finno-Ugric). The smallest improvement was found for the only Slavic language we tested (Polish).

With regard to the relatively low scores obtained by our NMT baseline, a number of comments are in order. First, we only tested certain standard/recommended settings in OpenNMT. It is likely that higher scores can be reached by tuning other NMT hyperparameters to better fit the dataset used. Second, SMT uses BLEU scores as tuning criterion, whereas in NMT perplexity is

[17] We realise one has to be careful when comparing BLEU scores across (target) languages.

		NMT	TM	SMT	TM-MT	Altered
EN-NL	*BLUE*	49.02	40.66	53.91	55.72**	25.5%
	TER	38.16	56.57	36.90	34.96	
	MET.	67.67	52.37	71.04	72.25	
EN-PL	*BLUE*	46.64	36.31	52.18	52.38	17.87%
	TER	39.57	60.85	37.79	37.54	
	MET.	35.45	26.39	38.67	38.73	
EN-DE	*BLUE*	42.57	38.37	47.32	49.59**	30.50%
	TER	44.81	59.13	44.43	41.95	
	MET.	55.56	45.05	60.11	61.71	
EN-FR	*BLUE*	52.76	41.00	59.08	59.65*	19.15%
	TER	35.79	57.63	32.96	32.22	
	MET.	67.31	50.16	72.97	73.45	
EN-HU	*BLUE*	37.75	34.33	35.71	41.18**	39.16%
	TER	48.01	61.72	55.31	49.98	
	MET.	55.23	45.66	55.67	58.67	
NL-EN	*BLUE*	52.55	43.17	59.00	60.63**	20.95%
	TER	35.11	55.13	32.32	30.56	
	MET.	41.65	30.28	44.95	45.51	
PL-EN	*BLUE*	52.21	42.49	61.95	62.57**	9.17%
	TER	35.28	55.54	29.42	28.86	
	MET.	42.17	29.94	46.60	46.85	
DE-EN	*BLUE*	47.59	42.50	55.44	57.17**	25.69%
	TER	39.90	55.73	36.49	34.67	
	MET.	38.70	30.17	43.05	43.46	
FR-EN	*BLUE*	55.42	43.11	56.39	57.12**	23.57%
	TER	32.42	55.14	35.33	34.23	
	MET.	44.02	30.37	45.81	45.70	
HU-EN	*BLUE*	45.09	41.51	48.62	52.10**	35.11%
	TER	43.35	56.13	44.25	40.37	
	MET.	37.51	29.60	40.10	40.93	

(p <0.01; ** p <0.001)*

Table 3: Results (significance tests for SMT vs TM-MT). Altered: % of sentences affected by TM-MT vs SMT

used to train the system. Third, BLEU evaluation focuses on precision (arguably the strength of SMT), and less on fluency (NMT's forte).[18] Finally, it is possible that SMT is more suited than NMT for contexts in which there is a considerable amount of repetition, and where adequacy and precision are crucial.

This study is limited in a number of ways: (a) the coverage of certain M3TRA components could still be improved, such as fuzzy repair, which could be extended to cover multiple edits per TM match or to also target non-sequential tokens, (b) only one dataset was used for testing, (c) only automatic metrics were used for evaluation, (d) BLEU scores were used for both training and testing, (e) no previously developed TM-MT integration method was used as baseline, and (f) the time spent on developing the NMT baseline was restricted. These limitations can be seen as suggestions for future research. For example, it would be interesting to see how professional translators appreciate M3TRA's output and indications of the origin of proposed translations, and what effect this has on translation efficiency. Some preliminary tests have been carried out (Coppers et al., 2018), but an in-depth study is still lacking. Such a study would also require us to take issues such as the positioning of formatting (and other types of tags) into consideration, which was outside the scope of the current paper. The same holds for a more qualitative evaluation of M3TRA's output (e.g. paying attention to certain morphological features).

7 Conclusions

We designed and tested a system for the integration of MT and TM, M3TRA, with a view to increasing the quality of MT output. M3TRA contains two main components, fuzzy repair and span pretranslation, which both make use of a TM with fuzzy matching techniques and an SMT system with related alignment information. The system uses the option to add XML markup to sentences sent to a Moses SMT system. Tests on ten language combinations using the DGT-TM dataset showed that it is clear that this approach has potential. Significantly higher BLEU scores for 9 of the 10 language combinations were observed, and METEOR and TER scores showed comparable patterns. In a next step, M3TRA has to be evaluated in an actual translation environment involving professional translators.

Acknowledgements

This research was done in the context of the SCATE project, funded by the Flemish Agency for Innovation and Entrepreneurship (IWT project 13007).

References

Biçici, E. and M. Dymetman. 2008. Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. *International Conference on Intelligent Text Processing and Computational Linguistics*, 454–465.

Bisazza, A. and M. Federico. 2016. A survey of word reordering in statistical machine translation: Computational models and language phenomena. *Computational Linguistics, 42*(2), 163-205.

Coppers, S., J. Van den Bergh, K. Luyten, I. van der Lek-Ciudin, T. Vanallemeersch and V. Vandeghinste. 2018. Intellingo: An Intelligible Translation Environment. *ACM conference on Human Factors in Computing Systems*, 1–13.

[18] It can be argued, however, that BLEU scores are a good evaluation metric in a context in which precision is important.

Espla-Gomis, M., F. Sánchez-Martínez and M.L. Forcada. 2015. Using machine translation to provide target-language edit hints in computer aided translation based on translation memories. *Journal of Artificial Intelligence Research, 53*(1), 169–222.

Feng, Y., S. Zhang, A. Zhang, D. Wang and A. Abel. 2017. Memory-augmented Neural Machine Translation. *arXiv preprint arXiv:1708.02005.*

Gu, J., Y. Wang, K. Cho and V.O. Li. 2017. Search Engine Guided Non-Parametric Neural Machine Translation. *arXiv preprint arXiv:1705.07267.*

He, Y. 2011. *The Integration of Machine Translation and Translation Memory*. Doctoral dissertation. Dublin City University.

Hewavitharana, S., S. Vogel and A. Waibel. 2005. Augmenting a statistical translation system with a translation memory. *10th Annual Conference of the European Association for Machine Translation*, 126–132.

Hokamp, C. and Q. Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. *arXiv preprint arXiv:1704.07138.*

Klein, G., Y. Kim, Y. Deng, J. Senellart and A.M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810.*

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. *Proceedings of EMNLP 2004*, 388-395.

Koehn, P. 2009. *Statistical machine translation*. Cambridge: Cambridge University Press.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, ... and C. Dyer. 2007. Moses: Open source toolkit for statistical machine translation. *45th annual meeting of the Association of Computational Linguistics*, 177–180.

Koehn, P. and J. Senellart. 2010. Convergence of translation memory and statistical machine translation. *2nd Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*, 21–31.

Lavie, A. and A. Agarwal. 2002. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. *2nd Workshop on Statistical Machine Translation*, 228–231.

Levenshtein, V.I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10*(8), 707–710.

Li, L., C.P. Escartín, A. Way and Q. Liu. 2017. Combining translation memories and statistical machine translation using sparse features. *Machine Translation, 30*(3), 183–202.

Li, L., A. Way and Q. Liu. 2016. Phrase-level combination of SMT and TM using constrained word lattice. *54th Annual Meeting of the Association for Computational Linguistics*, 275–280.

Ma, Y., Y. He, A. Way and J. van Genabith. 2011. Consistent translation using discriminative learning - A translation memory-inspired approach. *49th Annual Meeting of the Association for Computational Linguistics*, 1239-1248.

Ortega, J.E., F. Sánchez-Martínez, and M.L. Forcada. 2016. Fuzzy-match repair using black-box machine translation systems: what can be expected? *12th Biennial Conference of the Association for Machine Translation in the Americas*, Vol. 1, 27–39.

Papineni, K., S. Roukos, T. Ward and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Shterionov, D., P. Nagle, L. Casanellas, R. Superbo and T. ODowd. 2017. Empirical evaluation of NMT and PBSMT quality for large-scale translation production. *20th Annual Conference of the European Association for Machine Translation*, 74–79.

Simard, M. and P. Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. *Machine Translation Summit XII*, 120–127.

Simard, M. and P. Langlais. 2001. Sub-sentential exploitation of translation memories. *Machine Translation Summit VIII*, 335–339.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of the Association for Machine Translation in the Americas*, Vol. 200, No. 6.

Steinberger, R., A. Eisele, S. Klocek, S. Pilos and P. Schlüter. 2013. DGT-TM: A freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226.*

Sudoh, K., K. Duh, H. Tsukada, T. Hirao and M. Nagata. 2010. Divide and translate: improving long distance reordering in statistical machine translation. *Joint Fifth Workshop on Statistical Machine Translation and Metrics*, 418–427.

Wang, K., C. Zong and K.Y. Su. 2013. Integrating translation memory into phrase-based machine translation during decoding. *51st Annual Meeting of the Association for Computational Linguistics*, 11–21.

Zhechev, V. and J. Van Genabith. 2010. Seeding statistical machine translation with translation memory output through tree-based structural alignment. *4th Workshop on Syntax and Structure in Statistical Translation*, 43–51.

Reading Comprehension of Machine Translation Output: What Makes for a Better Read?

Sheila Castilho
ADAPT Centre
Dublin City University
sheila.castilho@adaptcentre.ie

Ana Guerberof Arenas
ADAPT Centre/SALIS
Dublin City University
ana.guerberof@adaptcentre.ie

Abstract

This paper reports on a pilot experiment that compares two different machine translation (MT) paradigms in reading comprehension tests. To explore a suitable methodology, we set up a pilot experiment with a group of six users (with English, Spanish and Simplified Chinese languages) using an English Language Testing System (IELTS), and an eye-tracker. The users were asked to read three texts in their native language: either the original English text (for the English speakers) or the machine-translated text (for the Spanish and Simplified Chinese speakers). The original texts were machine-translated via two MT systems: neural (NMT) and statistical (SMT). The users were also asked to rank satisfaction statements on a 3-point scale after reading each text and answering the respective comprehension questions. After all tasks were completed, a post-task retrospective interview took place to gather qualitative data. The findings suggest that the users from the target languages completed more tasks in less time with a higher level of satisfaction when using translations from the NMT system.

1 Introduction

Recently, there has been an increase in Neural Machine Translation (NMT) research as contemporary hardware supports much more powerful computation during the creation process. Research on the translation quality of NMT engines show that, in general, when compared against Statistical Machine Translation (SMT) engines, the output quality of NMT systems is higher when measured using automatic metrics (Bahdanau et al., 2014; Jean et al., 2015; Bojar et al., 2016; Koehn and Knowles, 2017). However, results are not as positive when human evaluators compare these outputs (Bentivogli et al., 2016; Castilho et al., 2017a; Castilho et al., 2017b).

Human evaluation of MT output, although not always implemented in quality evaluation, has been increasingly endorsed by researchers who acknowledge the need for human assessments. Some of the most commonly-used manual metrics are fluency and adequacy, error analysis, translation ranking, as well as post-editing effort. Despite the considerable focus on MT quality evaluation, the impact of MT on the end user has been under-researched. Measuring the usability of MT output allows for identification of the impact that the translation might have on the end user (Castilho et al., 2014). With the intention of exploring the cognitive effort required to read texts originating from SMT and NMT engines by the end users of those texts, we set-up a pilot experiment that aims to measure the reading comprehension of Spanish and Simplified Chinese users of texts produced by both paradigms using an eye-tracker (using the English users' data as a baseline).

The remainder of this paper is organized as follows: in Section 2, we survey the existing literature concerning reading comprehension for MT evaluation and the use of eye-tracking techniques for translation assessment; in Section 3, we describe the research questions and hypotheses which guide this pilot experiment, as well as the methodology applied to carry out the experiment with English

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 79–88
Alacant, Spain, May 2018.

(EN), Spanish (ES) and Simplified Chinese (ZH) native speakers; the results are discussed in Section 4, and finally, in Section 5, we draw the main conclusions of the pilot study and outline promising avenues for future work.

2 Related Work

2.1 Reading Comprehension for Machine Translation Evaluation

Despite the considerable focus on MT quality evaluation, there has not been much research focused on the impact of MT on the end user. With the current shift of paradigm in the MT landscape, it has become essential to also test the reading comprehension of NMT models by the end users of those translations. A few studies have attempted to measure reading comprehension (Scarton and Specia, 2016) and usability of MT output. Tomita et al. (1993) use reading comprehension tests to compare different MT systems. The content for reading and comprehension was extracted from an English proficiency exam and then translated into Japanese via three commercial MT systems as well as through the process of human translation. Sixty native speakers of Japanese were asked to read the text and answer the questions. The authors show that reading comprehension is a valid evaluation methodology for MT; however, their experiment only takes into consideration the informativeness, i.e. the number of correct answers for the comprehension questions.

Fuji (1999) proposes reading comprehension tasks in order to measure informativeness and, moreover, the author adds comprehensiveness and fluency to the evaluation measures. The content used comprises several texts from official examinations of English language designed for Japanese students. Participants were asked to read the text, answer the comprehension questions and judge how comprehensible and how fluent the text is, using a 4 point scale. Following on from this, Fuji et al. (2001) examined the "usefulness" of machine-translated text from two commercial MT systems compared to the English version. The experiment consisted of participants reading the texts and answering comprehension questions. The authors claim that presenting the source with the MT output results in higher comprehension performance.

Jones et al. (2005) ask 84 English native speakers to answer questions from a machine-translated and human-translated version of the Defense Language Proficiency Test for Arabic language. Task time and subjective rating were also measured. Their results suggest that MT may enable a limited working proficiency but it is not suitable for a general professional proficiency.

Usefulness, comprehensibility, and acceptability of MT technical documents are examined by Roturier (2006). The author claims that a text is deemed useful when readers are able to solve their problem with the help of the translation. The study uses a customer satisfaction questionnaire to determine whether controlled English rules can have a significant impact from a Web users perspective. The main drawback of Roturiers approach is that there is no task being performed by the end user as the methodology consists of an online questionnaire.

2.2 Eye tracking in Translation Research

Doherty and O'Brien (2012) is the first study to use eye-tracking techniques to measure the usability of translated texts via the end user. They conduct a study to compare the usability of raw machine-translated output for four target languages (Spanish, French, German and Japanese) against the usability of the source content (English). The result of this first phase compared the machine-translated group against the source group, and found significant difference for goal completion, efficiency, and user satisfaction between the source and the MT output. In the second phase of the study, Doherty and O'Brien (2014) analyse the results according to target languages compared to the source. The results show that the raw MT output scores lower for usability measurements, requiring more cognitive effort for all target languages when compared with the source language content.

Stymne et al. (2012) present a preliminary study using eye tracking as a complement to MT error analysis. In this methodology, although the main focus is to identify and classify MT errors, a comprehension task is also applied. For the perception questions, the human translation scored better than all the MT options. For both perceived and actual reading comprehension questions, their results show that participants are more efficient when using the MT output of a system trained using a large corpus. Regarding gaze data, MT errors are associated with both longer gaze times and more fixations than correct passages, and average gaze time is dependent on the type of errors which may suggest that some error types are more disturbing for readers than others.

Klerke et al. (2015) present an experimental eye-tracking usability test with text simplification and machine translation (for both the original and simplified versions) of logic puzzles. Twenty native speakers of Danish were asked to solve and judge 80 different logic puzzles while having their eye movements recorded. A greater number of fixations on the MT version of the original text (with no simplification) was observed and participants were less efficient when using the MT version of the original puzzles; however, the simplified MT version seemed to ease task performance when compared to the original English version.

Castilho et al. (2014) had two groups of 9 users each performing tasks using either the raw MT or the post-edited version of instructions for a PC-based security product, and cognitive and temporal effort indicators were gathered using an eye-tracker. Their results show that lightly post-edited instructions present a higher level of usability when compared to raw MT. Building on this, Castilho and O'Brien (2016) perform similar experiments with German and English native speakers, with instructions for spreadsheet software. Results show that the post-editing group is faster, more efficient, and more satisfied than the MT group. No significant differences appear in cognitive effort between raw and post-edited instructions, but differences exist between the post-edited versions and the source language. Moreover, the authors claim that the cognitive data should not be viewed in isolation, and highlight the importance of collecting qualitative data for measuring usability. Finally, Castilho (2016) extended previous experiments using Simplified Chinese, Japanese, German and English for the same set of instruction of the spreadsheet software. Results show that participants who used the post-editing instructions were more effective, more efficient, and faster than participants who used the raw MT instructions, especially for Simplified Chinese and German. Another interesting finding is that the source mostly did not differ from the post-editing groups, suggesting that the post-editing output is of equivalent quality. Regarding satisfaction, the author reports that German participants who use the MT instructions, even though they are able to successfully perform more tasks than other MT groups, are the least satisfied with the instructions, while the Japanese participants do not present any difference between the MT and post-editing groups for satisfaction even though the MT group was the least efficient. The author notes that these findings are likely to be related to cultural characteristics, as the Japanese participants are more tolerant and less likely to complain. Another interesting finding is that all groups, including the English-speaking participants, suggest that the instructions need improvements.

Finally, Jordan-Nez et al. (2017) compare three MT systems for assimilation, namely Systran (hybrid corpus based and rule-based MT); Google Translate (at the time of the experiment, a SMT system); and Apertium (a rule-based system), against professional translations. Results show that the MT output into a language in the same family as the readers first language may facilitate comprehension of texts originally written in a language from a different family. The authors note, however, that the level of usefulness depends on the field and on the MT system used as well as on the level of speciality.

Following previous work, we expect that the MT system that shows closer efficiency measures to the source text and lower task time, as well as lower cognitive effort indicators, is more likely to be rated higher for the satisfaction.

3 Methodology

Hypothesis and Research Questions As mentioned in Section 1, the primary aim of this experiment is to gather more information about the user experience when reading for comprehension machine-translated texts. With this aim in mind, we identified the following research questions:

RQ1: Which MT engine offers better efficiency to participants, i.e. with which one are they able to successfully answer more comprehension questions? Or with which one are they able to complete the tasks faster?

RQ2: To what extent are there differences in participants cognitive processes due to different engines (NMT and SMT)?

RQ3: What is the participants level of satisfaction with SMT and NMT when reading for comprehension?

Content and Design In order to answer the research questions, we measured participants reading comprehension according to the number of correct answers (goal completion) to a set of comprehension questions about each text, and task time. Eye-tracking fixation count and duration are also computed, as well as satisfaction indexes after each reading task. After all tasks were completed, we interviewed the participants by means of a semi-structured retrospective interview to gauge the understanding of the texts from a qualitative perspective.

For this pilot, we recruited two native speakers per language, a total of six participants (English, Spanish and Simplified Chinese languages). In this case, we used a sample of convenience. The participants were part of the student and staff body of Dublin City University. There were three female and three male participants, average age was 30.6 years, and all of them had received education to a post-graduate level. Half of them had previous experience in reading comprehension tests, either as part of their education or work. The Spanish and Simplified Chinese participants had a university level standard of English as they have taken English Proficiency tests and have been working and studying in an English-speaking country for some time.

As for the reading texts, two were taken from the International English Language Testing System (IELTS)[1] that measures English language proficiency by assessing four language skills: listening, reading, writing and speaking. IELTS has two types of tests: General and Academic. Since we were trying to assess the reception of raw output for a general user, we decided to use the General Training IELTS, reading modality, which contains a text and comprehension questions about that same text. The total number of words in the source content amounted to 1090 words.

The two English texts selected and their accompanying comprehension questions were then translated using Microsoft Translator Try and Compare feature[2] that allowed one to generate output in both SMT and NMT, and compare their quality. The first text (Text 2), entitled "Beneficial work practices for the keyboard operator", contained seven comprehension questions in which the users were required to choose the correct heading for each section of the text from a list of headings. The second text (Text 3), entitled "Workplace dismissals", contained five comprehension questions for which the users were required to match each description from a list with a correct term displayed in a box. One short text was also extracted from the IELTS website to be used as baseline. This baseline text (Text 1) was available in English, Spanish and Simplified Chinese on the IELTS website.[3] Moreover, ten questions in the style of the test (write True, False or Not given) were created in English for this baseline text and translated into Spanish by a Spanish translator and into Simplified Chinese by a native speaker. The baseline was used to test participants attention and reading comprehension with a human-translated version. The total number of words in the source baseline text amounted to 229 words. The baseline text was presented first followed by the Text 2 and Text 3 (SMT and NMT) which were randomised.[4] Figure 1 shows the set up of the task.

After each task (text and comprehension questions), four statements were presented (in English) in a three-point Likert scale (1- disagree, 2- neither agree or disagree, 3- agree) for the participants:

1. The subject of the text was easy to understand.
2. The language was easy to understand.
3. The question was easy to understand.
4. I was able to answer the question confidently.

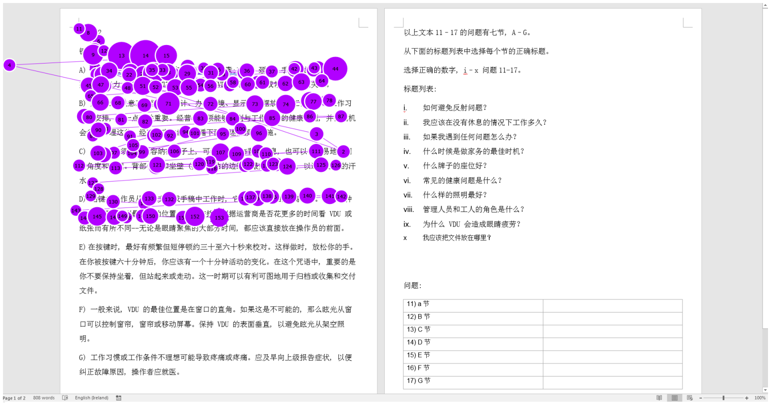
The eye tracker used was a Tobii T60XL with the filter set for I-VT (Velocity-Threshold Identification), as this is the filter recommended by Tobii for reading experiments. The participants were recorded during the post-task interview using the Flashback application that allows recording of all movements, sounds, and webcam output on the computer. This retrospective post-task interview was designed so that participants could watch their recordings and give their feedback regarding the subject matter, language used, questions, and personal experience when completing the whole task.

[1] https://www.ielts.org

[2] The feature on the website has changed to a comparison between Microsoft's production and research engines. See https://translator.microsoft.com/neural.

[3] As this text was available on the target languages on the IELTS official website, we assume that the translations were either direct human translation of the source or they were comparable texts, i.e. texts with the same information but originally written in the target language.

[4] The same order of texts were presented for the English participants (Text 1, Text 2 and Text 3) but in the source EN language.



[illegible]

A) [illegible]

B) [illegible]

C) [illegible]

D) [illegible] 据运营商是否花更多的时间看 VDU 或纸张而有所不同--无论是眼睛聚焦的大部分时间，都应该直接放在操作员的前面。

E) 在按键时，最好有频繁但短停顿约三十至六十秒来校对。这样做时，放松你的手。在你被按键六十分钟后，你应该有一个十分钟活动的变化。在这个咒语中，重要的是你不要保持坐着，但站起来或走动。这一时期可以有利可图地用于归档或收集和交付文件。

F) 一般来说，VDU 的最佳位置是在窗口的直角。如果这是不可能的，那么眩光从窗口可以控制窗帘，窗帘或移动屏幕。保持 VDU 的表面垂直，以避免眩光从架空照明。

G) 工作习惯或工作条件不理想可能导致疼痛或疼痛。应及早向上级报告症状，以便纠正故障原因，操作者应就医。

以上文本 11 - 17 的问题有七节，A - G。

从下面的标题列表中选择每个节的正确标题。

选择正确的数字，i - x 问题 11-17。

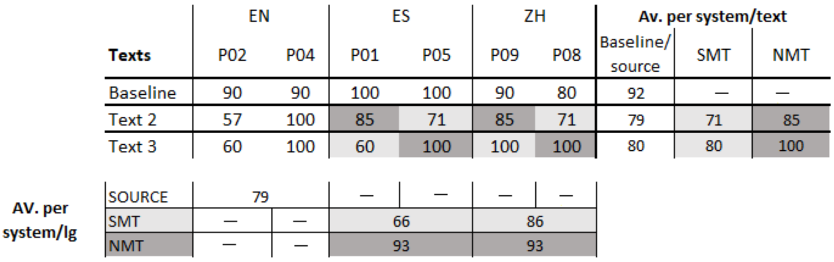
标题列表：

i. 如何避免反射问题？
ii. 我应该在没有休息的情况下工作多久？
iii. 如果我遇到任何问题怎么办？
iv. 什么时候是做家务的最佳时机？
v. 什么牌子的座位好？
vi. 常见的健康问题是什么？
vii. 什么样的照明最好？
viii. 管理人员和工人的角色是什么？
ix. 为什么 VDU 会造成眼睛疲劳？
x 我应该把文件放在哪里？

问题：

11) a 节	
12) B 节	
13) C 节	
14) D 节	
15) E 节	
16) F 节	
17) G 节	

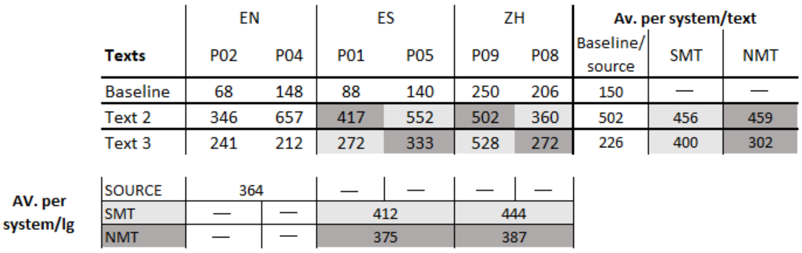
Figure 1: Task set-up



	EN		ES		ZH		Av. per system/text		
Texts	P02	P04	P01	P05	P09	P08	Baseline/ source	SMT	NMT
Baseline	90	90	100	100	90	80	92	—	—
Text 2	57	100	85	71	85	71	79	71	85
Text 3	60	100	60	100	100	100	80	80	100

AV. per system/lg						
SOURCE	79		—	—	—	—
SMT	—	—	66		86	
NMT	—	—	93		93	

Figure 2: Goal Completion (%)



	EN		ES		ZH		Av. per system/text		
Texts	P02	P04	P01	P05	P09	P08	Baseline/ source	SMT	NMT
Baseline	68	148	88	140	250	206	150	—	—
Text 2	346	657	417	552	502	360	502	456	459
Text 3	241	212	272	333	528	272	226	400	302

AV. per system/lg						
SOURCE	364		—	—	—	—
SMT	—	—	412		444	
NMT	—	—	375		387	

Figure 3: Task Time (in seconds)

4 Results

4.1 Comprehension

As mentioned previously, the baseline (Text 1) contained 10 questions, while Text 2 contained 7 questions, and Text 3 contained 5 questions. Goal completion is the number of successfully completed tasks, while task time is the total task time the participants needed to complete the tasks.

Goal Completion Figure 2 shows the results for goal completion for all participants (P01, P02, P04 and so on), where light gray cells are SMT while dark gray cells are NMT results. We can see that on average, participants who read the NMT text had a higher rate of goal completion (ES and ZH: 93%) when compared to the participants who read the SMT texts (ES: 66%, ZH: 86%), even when compared to participants who used the English source (79%). Interestingly, Simplified Chinese participants who used the SMT tests also had higher rates of goal completion when compared to the average for the English text.

When looking at the average score per system for each text (last column), participants of all languages had higher goal completion when reading Text 3 when compared to Text 2, which may indicate that Text 3 was easier to understand[5]. This is mentioned during the retrospective interviews by the participants (see Section 4.4).

Task Time Regarding the amount of time required for participants to read the texts and answer the comprehension questions, Figure 3 shows that,

[5] Text 3 contained 5 questions, whereas Text 2 contained seven question which could also have impacted goal completion

on average, participants who read the NMT output (ES: 375, ZH: 387) were faster than participants who read the SMT output (ES: 412, ZH: 444). Additionally, participants who used the NMT texts, for both ES and ZH, have closer average task time to participants who used the source text. Interestingly, the Simplified Chinese participants seemed to spend slightly more time on the task than the ES and EN participants, which could be related to the fact that the ZH participants were able to answer more questions correctly.

4.2 Eye-Tracking Data

As previously mentioned, we used an eye tracker to collect empirical data to analyse cognitive effort. Due to the low number of participants for the first part of this study, it is not possible to report any statistically significant results. However, we believe that these preliminary results may indicate a tendency in cognitive effort between NMT and SMT.

Fixation Duration (FD) is the length of fixations (in seconds) within an area of interest (AOI). The longer the fixations are, the higher the cognitive effort may be expected. Figure 4 shows the results for the length of fixations. The average fixation duration per system indicates that SMT presents longer fixations (sum) when compared to the NMT system for both ES and ZH. However, the mean length does not seem to differ much, and, in fact, for ZH it presents a slightly shorter mean (0.25 secs) than the NMT system (0.26 secs). In general, ZH participants present longer FD mean results when compared to ES and EN for both systems, including for the baseline (Text 1), which correlates with the time ZH participants spent on tasks (Figure 3).

Fixation Count (FC) is the total number of fixations within an AOI. The more there are, the higher the cognitive effort is deemed to be. The average FC per system for each language in Figure 5 indicates that, in general, SMT presents a higher number of fixations when compared to the fixation for the NMT system for both ES and ZH languages. Interestingly, ZH does not show higher means for FC as previously observed for FD. In fact, ZH participants show lower FC when compared to Spanish, and in the case of NMT, lower than the English as well.

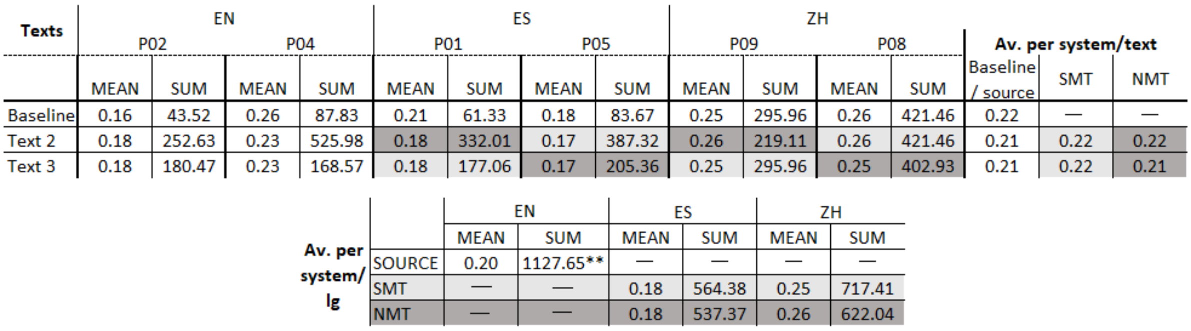
4.3 Satisfaction

As stated previously in Section 3, after the participants had completed each text and answered the comprehension questions, they were presented with four statements that measured their level of satisfactions with the subject of the text (the subject of the text was easy to understand), language (the language was easy to understand), questions (the question was easy to understand) as well as their perceived confidence (I was able to answer the question confidently) when answering the questions, in a 3-point Likert scale (3-agree, 1-disagree). Figure 6 presents the results for all languages.

In Figure 6, the average per system for each language shows that participants who used the EN texts have the highest satisfaction levels (2.56). For ES, participants who used the NMT system seem to be slightly more satisfied (1.6) than participants who used the SMT system (1.5). The same pattern can be seen in the ZH participants' satisfaction scores, the average for the NMT was considerably higher (2.37) than for the SMT system (1.37). This is in line with the task time (Figure 1) and goal completion (Figure 2) for the ZH language, in which participants were able to complete 93% of the tasks in an average of 387 secs using NMT translations, while using SMT translation they were able to complete 86% of the tasks in over 444 seconds. These results also illustrate the comments from the participants presented in the following section.

4.4 Retrospective Interviews

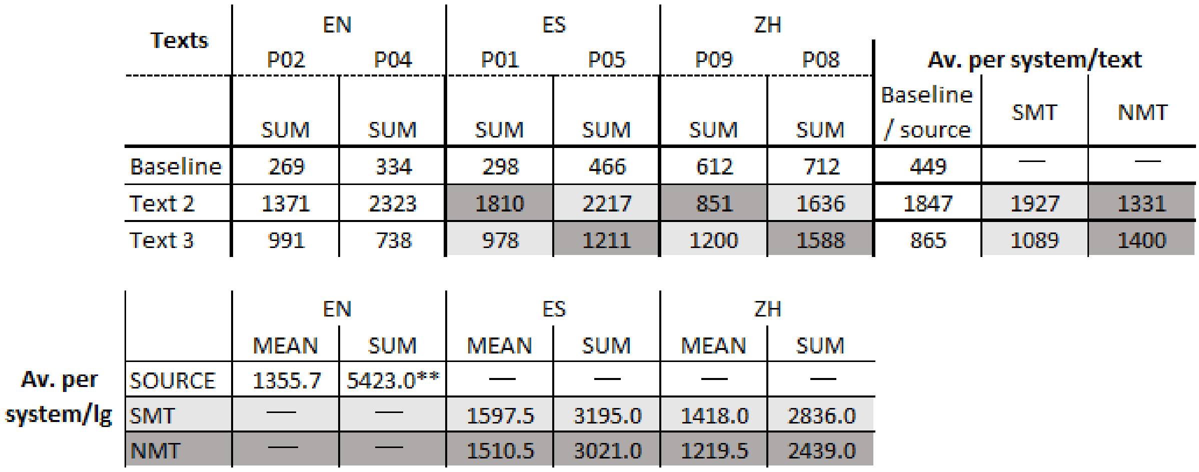
To triangulate the data from the eye-tracker and the statements presented to the participants after each task is completed (satisfaction scores), and obtain a more accurate account of the differences between SMT and NMT in reading comprehension tests, we carried out retrospective interviews with all participants. After each participant had completed the three tasks, we replayed the video of their eye movements in the Replay window of Tobii Studio, and recorded these interviews using Flashback as part of a Retrospective Think Aloud protocol. We asked the participants to watch the video showing their fixations on the screen and to describe freely their recollection of what they were thinking or doing at that time in the exercise. We clarified that they should not be worried about any grammar mistakes since four out of six of the



Texts	EN				ES				ZH				Av. per system/text		
	P02		P04		P01		P05		P09		P08				
	MEAN	SUM	MEAN	SUM	MEAN	SUM	MEAN	SUM	MEAN	SUM	MEAN	SUM	Baseline / source	SMT	NMT
Baseline	0.16	43.52	0.26	87.83	0.21	61.33	0.18	83.67	0.25	295.96	0.26	421.46	0.22	—	—
Text 2	0.18	252.63	0.23	525.98	0.18	332.01	0.17	387.32	0.26	219.11	0.26	421.46	0.21	0.22	0.22
Text 3	0.18	180.47	0.23	168.57	0.18	177.06	0.17	205.36	0.25	295.96	0.25	402.93	0.21	0.22	0.21

Av. per system/ lg		EN		ES		ZH	
		MEAN	SUM	MEAN	SUM	MEAN	SUM
	SOURCE	0.20	1127.65**	—	—	—	—
	SMT	—	—	0.18	564.38	0.25	717.41
	NMT	—	—	0.18	537.37	0.26	622.04

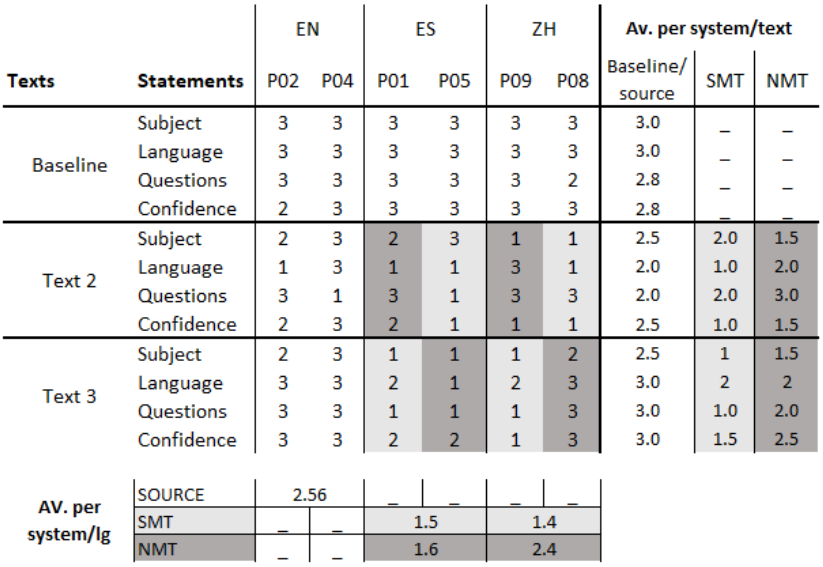
Figure 4: Fixation Duration - in seconds.(** is the sum for both EN participants for both Text 2 and 3)



Texts	EN		ES		ZH		Av. per system/text		
	P02	P04	P01	P05	P09	P08			
	SUM	SUM	SUM	SUM	SUM	SUM	Baseline / source	SMT	NMT
Baseline	269	334	298	466	612	712	449	—	—
Text 2	1371	2323	1810	2217	851	1636	1847	1927	1331
Text 3	991	738	978	1211	1200	1588	865	1089	1400

Av. per system/lg		EN		ES		ZH	
		MEAN	SUM	MEAN	SUM	MEAN	SUM
	SOURCE	1355.7	5423.0**	—	—	—	—
	SMT	—	—	1597.5	3195.0	1418.0	2836.0
	NMT	—	—	1510.5	3021.0	1219.5	2439.0

Figure 5: Fixation Count (** is the sum for both EN participants for both Text 2 and 3)



		EN		ES		ZH		Av. per system/text		
Texts	Statements	P02	P04	P01	P05	P09	P08	Baseline/ source	SMT	NMT
Baseline	Subject	3	3	3	3	3	3	3.0	_	_
	Language	3	3	3	3	3	3	3.0	_	_
	Questions	3	3	3	3	3	2	2.8	_	_
	Confidence	2	3	3	3	3	3	2.8	_	_
Text 2	Subject	2	3	2	3	1	1	2.5	2.0	1.5
	Language	1	3	1	1	3	1	2.0	1.0	2.0
	Questions	3	1	3	1	3	3	2.0	2.0	3.0
	Confidence	2	3	2	1	1	1	2.5	1.0	1.5
Text 3	Subject	2	3	1	1	1	2	2.5	1	1.5
	Language	3	3	2	1	2	3	3.0	2	2
	Questions	3	3	1	1	1	3	3.0	1.0	2.0
	Confidence	3	3	2	2	1	3	3.0	1.5	2.5

AV. per system/lg		EN		ES		ZH	
	SOURCE	2.56		_	_	_	_
	SMT	_	_	1.5		1.4	
	NMT	_	_	1.6		2.4	

Figure 6: Ratings of Satisfaction (the higher score, the better)

participants did not have English as their mother tongue, the language in which the interviews were conducted. At the time of writing this paper, we have not completed a full qualitative analysis of these interviews, that is transcription and coding of the recordings, therefore what we provide here is a summary of the preliminary results.

All participants in all languages indicated that

Text 1 (the baseline text: original English or human translation) was easy to understand. They found the text to be short, the content easy to understand, and the language clear. Regarding Text 2, although most participants mentioned that it was more time consuming mainly due to the number of questions and options available (seven questions and ten options to choose from), their assessment of the language quality varied depending on the language and the type of engine used for this experiment. The same applies for Text 3, although the participants indicated that it was faster to complete because there were fewer questions and they already knew the dynamic of the exercises.

In the case of the English-speaking participants, they did not mention any aspects of the language or content that they found particularly difficult, although one participant (P02) had difficulties with the coding system to answer the questions in Text 1 (True, False, Not given). This participant also mentioned that he was not happy with certain commas or double negatives on Text 2. He did not find any linguistic issues on Text 3. The other English participant, P04, found the language to be satisfactory.

If we look at the Spanish language, P01 mentioned that Text 2 (NMT engine) was "more confusing" than Text 1 (Human translation). There were keywords that were "tricky" and she thought they were probably wrong, such as *sostenedor* instead of *atril* for *holder*, also she mentioned words that seemed to be completely out of context, such as *hechizo* for spell. Regarding Text 3 (SMT), the participant said that it was "really, really tricky" and "the language was really difficult" not because of words but because of incorrect grammar, and she stated that sentences were difficult to understand. She commented that "there were times where it came to my mind that these were direct translations from English". Because of the incorrect translations provided by the engine (two English options were translated in the same way in Spanish by the SMT engine), the participant answered two questions incorrectly. Participant 5 mentioned that in Text 2 (SMT, in this case), he noticed grammar mistakes "straight away", and then he realised that "it was translated by a machine" as "almost every sentence had something wrong". He mentioned that, although he had to read the sentences several times to try and make sense of the meaning, the content was not difficult for him. On the other hand, he found Text 3 (NMT, in this case) easier because there were fewer questions to answer, but he also mentioned that Text 3 was machine-translated. He noticed a few grammar errors and inconsistencies. For example, he noticed *Despido sumario* and *Resumen despido* as a translation for *Summary Dismissal*, and *Constructivo Despido* and *Constructivo despido* for *Constructive dismissal*, and this created confusion when he was answering the comprehension questions. He thought that the language was more technical than in the other documents but at the same time that the questions were easier to answer. When asked if he saw any difference between Text 2 and Text 3, he said that he had no reasons to assume a different MT system was used.

Regarding the Simplified Chinese language, P08 stated that Text 2 (SMT in this case) was the most difficult text of the three. According to him, Text 2 "was not fluent", some words were "weird", and he had to guess a lot of the text by the context and the questions. For him, the first two paragraphs, for example, were difficult to understand. Therefore, both contents and language were difficult. Regarding Text 3 (NMT), P08 found that it was "in the middle of the three". The paragraphs were "better" and the questions were "clear". Although, the content was new to the participant, he found the language easier to understand in Text 3 than in Text 2 but worse than in Text 1, as "the words were correct", but the order was wrong, and there were also characters missing. As for P09, she found that the structure of Text 2 (NMT, in this case) was "okay" but she was not familiar with the topic. She thought the language was also "okay"; although there were errors and sometimes the vocabulary was incorrect, she could understand it. In this text, she found the headings difficult to place in the corresponding section. P09 found that Text 3 (SMT in this case) was the most difficult one. She understood that the text was about dismissals, but she found the language "strange", "totally unclear", "the structure was not that good" and it was "hard to understand". She found that Text 2 and Text 3 were stressful, especially Text 3. She commented that she could understand 60 percent of Text 2, but only 20 percent of Text 3.

In summary, the EN participants found Text 2 more cumbersome to resolve than Text 1 and Text 3, and therefore more time was required, but only P02 mentioned that the language was an issue and that it could be improved in Text 2 with regards to commas and double negatives. This is very interesting as it suggests that the difficulties EN participants found in the source could have been translated in the target languages. For ES and ZH, the four participants found Text 1 (human translation) easy in content and language, while they were divided on Text 2 and Text 3. In Simplified Chinese, the texts translated with NMT, regardless of whether they were Text 2 or 3, were viewed as better linguistically than their counterparts translated with SMT, even when the NMT texts had certain terms or grammar turns that were wrong, and this influenced the participants' responses. In Spanish, one of the participants found the NMT option better linguistically, while the other participant found that both options were comparable and possibly came from the same MT system.

5 Conclusions and Future Work

The aim of this pilot experiment was to verify the methodology to measure the impact of the quality of two MT paradigms - NMT and SMT - on the end user. For that, we established three research questions regarding efficiency (goal completion), cognitive effort, and satisfaction.

Regarding RQ1 (Which MT engine offers better efficiency to participants?), results show that participants (Figure 2) in the two target languages - Spanish and Simplified Chinese - were able to complete more tasks successfully when using the NMT translated texts when compared to the SMT translations, as well as when compared to participants who used the original EN texts. Regarding the time spent to complete the texts, again, we noted that when using the NMT translations, participants were faster than when using SMT translations and, moreover, have task completion times closer to participants who used the English text than the results for SMT.

Regarding RQ2 (To what extent are there differences in participants cognitive processes due to different engines?), results for the FD (Figure 4) and FC (Figure 5) show that cognitive effort does not seem to differ much for ES, and presents a bit of mixed results for ZH, were FD are slightly longer for the NMT system, whereas FC are lower. We believe that with a greater number of participants, a clearer tendency would be observed.

Regarding our last research question (RQ3: What is the participants level of satisfaction with SMT and NMT when reading for comprehension?), participants rated NMT higher and also commented that the language in NMT texts was easier to understand in the post-task retrospective interviews. It is also necessary to point out that ES and ZH participants commented on the fact that the language in the human translation (Text 1) was easy to understand, while they struggled in certain sections in both NMT and SMT texts (Texts 2 and 3). This was not the case with EN participants that only made slight remarks on the quality of the English, but they did not mention any misunderstandings of the texts.

We are aware of the limitations of the results presented here since the number of participants was very low, and there were few texts for each MT system. Our next steps are to add more languages, especially those languages which have been showing greater improvement with NMT over the SMT paradigm, as well as gathering more participants. Another consideration to bear in mind is the nature of the texts; we noted that the combination of difficult text with easy questions and vice-versa could cloud the findings.

Furthermore, we believe that this research could benefit from computing more eye-tracking measures, such as visit count, which is the number of visits to an area of interest, as the shifts of attention between the questions and the text may be an indicator of cognitive effort (Castilho et al., 2014).

Acknowledgements: We would like to thank Dag Schmidtke from Microsoft Ireland and Joss Moorkens for invaluable help, and the participants for their support on this pilot experiment. This research was supported by the Edge Research Fellowship programme that has received funding from the European Unions Horizon 2020 and innovation programme under the Marie Sklodowska-Curie grant agreement No. 713567, and by the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106) and co-funded under the European Regional Development Fund.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and

Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. *CoRR*, abs/1608.04631.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August. Association for Computational Linguistics.

Castilho, Sheila and Sharon O'Brien. 2016. Evaluating the impact of light post-editing on usability. In *LREC*, pages 310–316, Portoroz, Slovenia, May.

Castilho, Sheila, Sharon O'Brien, Fabio Alves, and Morgan O'Brien. 2014. Does post-editing increase usability? A study with Brazilian Portuguese as Target Language. In *Proceedings of European Association for Machine Translation (EAMT)*, pages 183–190, Dubrovnik, Croatia.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017a. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Panayota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. 2017b. A Comparative Quality Evaluation of PBSMT and NMT using Professional Translators. In *MT Summit 2017*, Nagoya, Japan.

Castilho Monteiro de Sousa, Sheila. 2016. *Measuring acceptability of machine translated enterprise content*. Ph.D. thesis, Dublin City University.

Doherty, Stephen and Sharon O'Brien. 2012. A user-based usability assessment of raw machine translated technical instructions. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*, pages 1–10, San Diego, California, USA.

Fuji, Masaru, N Hatanaka, E Ito, S Kamei, H Kumai, T Sukehiro, T Yoshimi, and Hitoshi Isahara. 2001. Evaluation method for determining groups of users who find mt useful. In *MT Summit VIII: Machine Translation in the Information Age*, pages 103–108.

Fuji, Masaru. 1999. Evaluation experiment for reading comprehension of machine translation outputs. In *Proceedings of MT Summit VII*, pages 285–289.

Jean, Sébastien, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal Neural Machine Translation Systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal, September.

Jones, Douglas, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds, and Clifford Weinstein. 2005. Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 5, pages v–1009. IEEE.

Jordan-Nez, Kenneth, Mikel L Forcada, and Esteve Clua. 2017. Usefulness of mt output for comprehension an analysis from the point of view of linguistic intercomprehension. volume 1, pages 241–253, September.

Klerke, Sigrid, Sheila Castilho, Maria Barrett, and Anders Søgaard. 2015. Reading metrics for estimating task efficiency with mt output. In *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 6–13.

Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR*, abs/1706.03872.

Roturier, Johann. 2006. *An investigation into the impact of controlled English rules on the comprehensibility, usefulness and acceptability of machine-translated technical documentation for French and German users*. Ph.D. thesis, Dublin City University.

Scarton, Carolina and Lucia Specia. 2016. A reading comprehension corpus for machine translation evaluation. In Chair), Nicoletta Calzolari (Conference, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Stymne, Sara, Henrik Danielsson, Sofia Bremin, Hongzhan Hu, Johanna Karlsson, Anna Prytz Lillkull, and Martin Wester. 2012. Eye tracking as a tool for machine translation error analysis. In *LREC*, pages 1121–1126.

Tomita, Masaru, Masako Shirai, Junya Tsutsumi, Miki Matsumura, and Yuki Yoshikawa. 1993. Evaluation of mt systems by toefl. In *Proceedings of the Theoretical and Methodological Implications of Machine Translation (TMI-93)*, pages 252–265.

Are Automatic Metrics Robust and Reliable in Specific Machine Translation Tasks?

Mara Chinea-Rios **Álvaro Peris** **Francisco Casacuberta**
Pattern Recognition and Human Language Technology Research Center
Universitat Politècnica de València, València, Spain
{machirio, lvapeab, fcn}@prhlt.upv.es

Abstract

We present a comparison of automatic metrics against human evaluations of translation quality in several scenarios which were unexplored up to now. Our experimentation was conducted on translation hypotheses that were problematic for the automatic metrics, as the results greatly diverged from one metric to another. We also compared three different translation technologies.

Our evaluation shows that in most cases, the metrics capture the human criteria. However, we face failures of the automatic metrics when applied to some domains and systems. Interestingly, we find that automatic metrics applied to the neural machine translation hypotheses provide the most reliable results. Finally, we provide some advice when dealing with these problematic domains.

1 Introduction

Machine translation (MT) assessment is an open research question. The most accurate methods require a manual evaluation of the MT system. Unfortunately, this is a difficult and costly process, being unaffordable while developing new MT engines. Therefore, protocols for automatic evaluation of MT are required. The most common approach for evaluating the MT quality is to compare the system hypotheses with one or more reference sentences and compute a quality score.

A significant research effort has been spent on enhancing the automatic metrics. For instance, a shared task is running since 2008, as part of the Conference on Machine Translation (WMT). Although several metrics have been proposed, the literature is nowadays dominated by BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002) and, to a lesser extent, by TER (Translation Edit Rate) (Snover et al., 2006).

Despite their usefulness, those metrics may diverge, sometimes leading to deceiving conclusions. This is the case of unconventional tasks or domains (e.g Chinea-Rios et al. (2017)).

This work aims to shed some light on these behaviors, by conducting a human evaluation of MT outputs that produce inconsistencies in the metrics. More precisely, we study the correlation between human judgment and automatic evaluation on three problematic domains and for three different MT systems. We analyze the strengths and flaws that each automatic metric conveys, giving some advice for future research. The main contributions of this paper are the following:

- We deepen into an unexplored field: the evaluation of MT outputs which present inconsistencies between automatic metrics.
- We conduct a human evaluation of MT hypothesis which produced inconsistent automatic evaluations, following the direct assessment (DA) methodology.
- We compare a large number of state-of-the-art automatic metrics for our tasks at hand.
- We study the correlation of all metrics with human judgments, finding out that automatic metrics capture relatively well the human evaluation criteria in several cases.

This paper is structured as follows: in Section 2, we review relevant literature in the field of MT evaluation. Section 3 provides a brief summary of the metrics under study in this work. Section 4 explains the methods used to evaluate MT. In Section 5, we describe the experimental setup. We show and discuss the results of our evaluation in Section 6. Finally, we conclude in Section 7 by highlighting the main lessons learned from this work.

2 Related work

As stated in the previous section, the automatic evaluation of MT quality is a key element for the effective development of MT. Therefore, it has been studied from long ago (Pierce and Carroll, 1966; White et al., 1994). From here, a large amount of metrics have been proposed. Among them, the most widely used, especially in the academia, is the aforementioned BLEU. Nonetheless it is also widely accepted that BLEU suffers from several limitations when correlating with human judgments (Turian et al., 2003; Tatsumi, 2009) and can be fooled with bad translations (Smith et al., 2016). Other metrics are also common in the literature. This is the case of TER, METEOR (Lavie and Denkowski, 2009), word error rate (WER) (Klakow and Peters, 2002; Morris et al., 2004) or NIST (Doddington, 2002). Despite these efforts, the automatic assessment problem remains open, being organized several evaluation campaigns (Mauro et al., 2017) and shared tasks (Bojar et al., 2017a).

Due to the fragility and ambiguity of the existing metrics, several works attempted to perform a fine-grained evaluation of different MT systems or technologies. With the recent irruption of the neural machine translation (NMT) paradigm, a natural question arises: is NMT better than classical phrase-based statistical machine translation (PB-SMT) systems?

Several works aimed to answer this question. Toral and Sánchez-Cartagena (2017) performed an extensive comparison of NMT and PB-SMT systems, measuring several facets of the translation, such as similarity, fluency or reordering. Error analyses of NMT and PB-SMT have also been reported, either automatic (Bentivogli et al., 2018) or manual (Klubička et al., 2017). The conclusions were alike: NMT handled better verbs and nouns reordering, while the translation of proper nouns was worse.

However, it is still uncertain whether the NMT paradigm works better in situations with scarce data, as pointed out by Koehn and Knowles (2017). A solution to this issue is to add monolingual data. The usage of synthetic data in NMT has reported excellent results in terms of BLEU (Chinea-Rios et al., 2017; Sennrich et al., 2016a); but a study on the importance of adding synthetic data in NMT with respect to the human perception of translation is still missing.

3 Automatic evaluation of machine translation

In the context of this paper, the goal of automatic metrics is to assign scores to MT outputs in a way that they correlate with a human evaluation of the translation quality. In this section we briefly describe the eight metrics compared in this work. These are the most common metrics used for evaluating MT.

3.1 BLEU

BLEU tries to model the correspondence between the output from a MT system and the one produced by a human. The BLEU score is based on the n-gram precision. It counts the number of n-grams from the hypothesis that appear in the reference, dividing this count by the number of n-grams in the hypothesis. This count is clipped to the maximum number of counts that the n-gram has in any sentence of the reference document. BLEU also features a brevity penalty for short translations.

The final BLEU score is computed as a geometric mean of the n-gram precision, modified by the brevity penalty. The maximum order of the n-grams involved in the computation of BLEU is set to 4, as this provides the highest correlation with human evaluation, according to the original experimentation (Papineni et al., 2002).

3.2 METEOR

BLEU only considers n-gram precision, ignoring the recall component. Moreover, it lacks an explicit word matching. METEOR aims to mitigate these issues. METEOR is an alignment-based metric, which computes all valid alignments between the hypothesis and the references. For computing these alignments, it makes use of a stemmer and a synonym database. Therefore, this is a language-dependent metric.

Once the set of alignments is computed, the METEOR metric is a harmonic mean of the unigram precision and unigram recall, modified by an alignment penalty.

3.3 TER

The TER is defined as the minimum number of word edit operations that must be made in order to transform the hypothesis into the reference. The edit operations considered are insertion, substitution, deletion and swapping groups of words. The number of edit operations is normalized by the number of words in the reference sentence. The minimum number of edit operations is obtained by dynamic programming. Note that, unlike BLEU and METEOR, this is an error-based metric. Hence, the lower, the better.

3.4 WER

Metric based on the Levenshtein distance, working at word level. WER is based on the calculation of the number of words that differ between a piece of machine translated text and a reference translation. WER is similar to TER but ignoring the swapping operation. It was originally used for measuring the performance of speech recognition systems, but was also used in the evaluation of machine translation. As TER, the lower the WER, the better.

3.5 PER

Position independent word Error Rate (PER) (Tillmann et al., 1997) is similar to TER and WER but comparing the words in the two sentences without considering the word order. The PER score is always lower than or equal to WER. On the other hand, a shortcoming of the PER is that the word order may be important in some cases. Therefore the best solution is to calculate both word error rates.

3.6 NIST

NIST was designed to improve BLEU by rewarding the translation of infrequently used words. This was intended to prevent the inflation of MT evaluation scores by focusing on common words and high confidence translations. As a result, the NIST metric assigns larger weights to infrequent words. Similarly to BLEU, the final NIST score is computed according to the arithmetic mean of the weighted n-gram matches between the MT outputs and the reference translations. A brevity penalty is also included. The reliability and quality of the NIST metric has been shown to be superior to BLEU in several cases.

3.7 BEER

BEtter Evaluation as Ranking (BEER) (Stanojević and Sima'an, 2014a,b, 2017) is a trained evaluation metric with a linear model that combines subword feature indicators (character n-grams) and global word order features (skip bi-grams) to get a language agnostic and fast to compute evaluation metric. This metric obtained very high correlation values with human evaluations in the last evaluation campaigns (e.g. Bojar et al. (2017a)).

3.8 CHRF

Character n-gram F-score (CHRF) (Popović, 2015) computes the F_β-score on the character n-gram precision and recall. According to Popović (2015), using an F_3-score correlated best with human judgment. Its popularity is increasing, as it has shown to be a reliable metric for NMT systems.

4 Methodology

In this section we describe the human evaluation protocol applied in our work. We also describe how we computed the correlation across metrics.

4.1 Direct Assessment

Following the metrics shared task from WMT'17 (Bojar et al., 2017a), we used the monolingual DA model for evaluating the translation adequacy (Graham et al., 2017).

To obtain a correct measure of the translation quality is difficult to achieve, and the DA setup simplifies this task: unlike classical translation assessment protocols (typically bilingual), this is a simpler framework. In DA, the translation adequacy is structured as a monolingual assessment of semantic similarity, in which the reference translation and the MT hypothesis are displayed to the human evaluator. Assessors rate a translation by scoring how adequately it expresses the meaning of the reference translation. The evaluation scale ranges from 0 (worst) to 100 (perfect).

In order to avoid the skew from the different evaluators, we standardized all the scores. The standard score z of a raw score x is computed as:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the average and standard deviation of the scores population, respectively.

4.2 Computing metric correlations

For computing the correlation between two metrics, we applied the widely used Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^{n}\left(h_i-\bar{h}\right)\left(m_i-\bar{m}\right)}{\sqrt{\sum_{i=1}^{n}\left(h_i-\bar{h}\right)^2}\sqrt{\sum_{i=1}^{n}\left(m_i-\bar{m}\right)^2}} \tag{2}$$

where n is the number of samples, h_i is the human assessment score of the i-th translation hypothesis and m_i is the corresponding scores to that hypothesis given by an automatic metric. $\bar{h}$ and $\bar{m}$ are the human and automatic mean scores, respectively.

The r coefficient ranges from $+1$ to -1, where $+1$ means total positive correlation and -1 denotes total inverse correlation. A value of 0 means that there is no linear correlation between both variables.

In this work, we compute statistical significance tests, computing a confidence level of α as:

$$\alpha = 2p_t(|t|, n-2) \tag{3}$$

where p_t denotes the cumulative density function of t and the t value is obtained computed following:

$$t = r\frac{\sqrt{n-2}}{\sqrt{1-r^2}} \tag{4}$$

5 Experimental setup

Our experimental framework related a domain adaptation task, in the English to Spanish language direction. In our setup, we trained a PB-SMT and a NMT system on the same data, from a general corpus extracted from websites (Common Crawl). We applied these systems to three different domains: printer manuals (XRCE) (Barrachina et al., 2009), information technology[1] (IT) and Electronic Commerce (E-Com). We adapted the NMT system to these domains via synthetic data, as proposed by Chinea-Rios et al. (2017). This method consists in, for each domain, selecting related samples from a large monolingual pool, back-translating them and fine-tuning the general NMT system with these data. Table 1 show the main figures of these datasets. It is worth noting the differences existing between the domains, in terms of sentence length: The Common Crawl and IT domains featured long sentences (with around 20 words per sentence); while the XRCE and E-Com domains had much shorter sentences. This shows that the first two domains contained sentences with much more context than the two latter.

	Corpus		$\lvert S\rvert$	$\lvert W\rvert$	$\lvert V\rvert$	$\overline{\lvert W\rvert}$
Training	Common Crawl	En	1.5M	30M	456k	20.0
		Es		31M	522k	20.0
	IT – Syn	En	150k	2.5M	76k	16.7
		Es		3.0M	78k	20.0
	XRCE – Syn	En	180k	2.2M	54k	9.4
		Es		1.7M	58k	12.2
	E-Com – Syn	En	300k	3.2M	100k	10.6
		Es		4.1M	100k	13.6
Test	IT	En	857	15.6k	2.1k	18.2
		Es		17.4k	2.4k	20.3
	XRCE	En	1.1k	8.4k	1.6k	7.6
		Es		10.1k	1.7k	9.2
	E-Com	En	886	7.3k	874	8.2
		Es		8.6k	973	9.7

Table 1: Corpora main figures, in terms of number of sentences ($|S|$), number of words ($|W|$), vocabulary size ($|V|$) and average sentence length ($\overline{|W|}$). Syn indicates synthetic data used for fine-tuning the NMT system. M and k denote millions and thousands, respectively.

5.1 Machine translation systems

We built an attentional recurrent encoder–decoder NMT system, using the NMT-Keras[2] toolkit. The encoder and decoder were made of long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997). Following Britz et al. (2017), the LSTM, word embedding and attention model dimensions were 512 each. We applied joint byte-pair encoding (Sennrich et al., 2016b), with $32,000$ merge operations. We used Adam (Kingma and Ba, 2014) with a learning rate of 0.0002. For obtaining the translations, we used a beam search with a beam size of 6. The fine-tuning of the systems via synthetic data (denoted by NMT+Syn) was made using vanilla SGD with a learning rate of 0.05.

Our PB-SMT system was built using the standard configuration of Moses (Koehn et al., 2007). The language model was a 5-gram with modified Kneser-Ney smoothing (Kneser and Ney, 1995). The phrase table was generated employing symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The weights of the log-linear model were tuned using MERT (Minimum Error Rate Training) (Och, 2003).

The metrics were computed using the scripts provided at the WMT metrics shared task (Bojar

[1] `http://metashare.metanet4u.eu/qtleapcorpus`

[2] `https://github.com/lvapeab/nmt-keras`

et al., 2017b). For all metrics, we used a single reference.

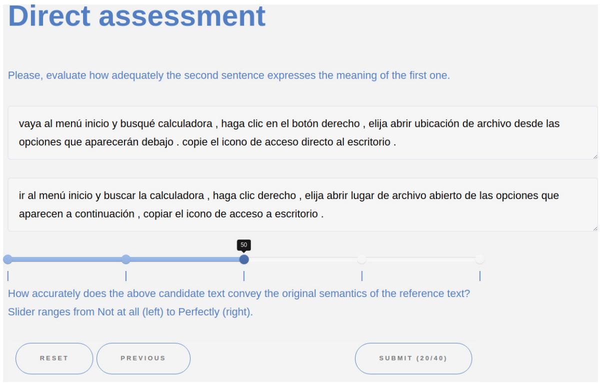
5.2 Human evaluation experiments

For each domain and MT system, we randomly sampled several translation hypotheses. The samples were arranged in 8 non-overlapping blocks of 40 sentences each. Each block was evaluated by two users. Therefore, each sentence was assessed twice. Table 2 show figures of the distribution of evaluated sentences according to each system and domain. 16 human evaluators participated in our study, all native speakers of the target langauge (Spanish). None of them was a professional translator. Note that, as we are using the DA framework, the evaluators do no require any knowledge of the source language.

$\vert S\vert$	Domain	MT system		
		Moses	NMT	NMT+Syn
320	IT	40	24	24
	XRCE	40	32	40
	E-Com	32	48	40

Table 2: Figures of the evaluated samples. We show the total number of sentences ($|S|$) and the distribution of sentences from each domain and MT system.

We developed a web page[3] to follow the DA methodology (see Fig. 1 for an example of the front-end). The users were asked to assess *how accurately does the candidate text convey the original semantics of the reference text?*. The ratings ranged from 0 (worst) to 100 (perfect).



Direct assessment

Please, evaluate how adequately the second sentence expresses the meaning of the first one.

vaya al menú inicio y busqué calculadora , haga clic en el botón derecho , elija abrir ubicación de archivo desde las opciones que aparecerán debajo . copie el icono de acceso directo al escritorio .

ir al menú inicio y buscar la calculadora , haga clic derecho , elija abrir lugar de archivo abierto de las opciones que aparecen a continuación , copiar el icono de acceso a escritorio .

50

How accurately does the above candidate text convey the original semantics of the reference text?
Slider ranges from Not at all (left) to Perfectly (right).

Figure 1: Front-end of the webpage developed for performing the DA protocol. The users were asked to assess how accurately does the candidate text convey the original semantics of the reference text.

6 Results and discussion

In this section, we present and discuss the results obtained from our experimentation. We analyze all metrics according to the domain and to the translation technology. Firstly, we show the overall metrics for the three different MT systems in the three different domains. Table 3 shows the BLEU, TER and METEOR scores of our data, as well as the scores given by the human evaluators.

Domain	System	BLEU	TER	METEOR	HUMAN
IT	Moses	33.2	45.8	60.6	58.4
	NMT	34.1	52.8	53.3	64.7
	NMT+Syn	32.2	47.3	58.3	66.3
XRCE	Moses	23.6	61.8	47.5	51.2
	NMT	22.3	78.3	44.7	47.9
	NMT+Syn	23.1	62.0	43.5	47.4
E-Com	Moses	26.2	51.8	46.8	59.7
	NMT	25.5	84.7	45.5	40.7
	NMT+Syn	30.3	52.3	48.9	43.3

Table 3: Human and automatic metrics, for all systems and domains. BLEU, METEOR and HUMAN scores range from 0 to 100, being the higher values, the better. On the other hand, the lower the TER values, the better.

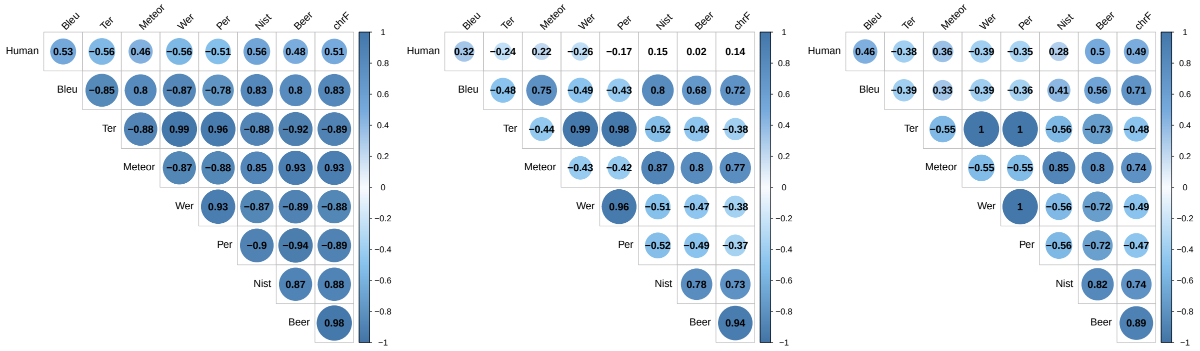
This table reflects the large differences that automatic metrics may produce: for the E-Com task, the NMT system is 0.7 BLEU points worse than Moses, but its TER is more than 30 points worse than Moses. Other inconsistencies in the metrics can be found in this table. We now deepen in these results, performing a fine-grained analysis.

6.1 Evaluating the domains

We compared each domain, regardless the translation technology applied to obtain the translations. Fig. 2 presents the correlation matrix of all metrics, for each domain. Moreover, it also shows whether the correlation of a metric with respect another is statistically significant at $\alpha \leq 0.05$ (dotted cells) or not (white cells).

It is interesting to observe the large difference in terms of correlation existing between the different domains. The IT domain correlated much better with the human judgments that the other two domains (XRCE and E-Com). The reasons of such differences were found in the different corpora features: IT was a more complex corpus than the other two, featuring longer sentences with more complex syntactic structures. Moreover, in this domain, all automatic metric exhibited a significant correlation with respect to the human judgment. TER, WER and METEOR achieved

[3] The evaluation platform, scores and all data used in this work can be found at: `http://lvapeab.github.io/mt_evaluation.html`.



(a) IT domain. **(b)** XRCE domain. **(c)** E-Com domain.

Figure 2: Correlation of the metrics across different domains (IT, XRCE, E-Com). Blue circles denote a statistically significant correlation with respect to the human assessment for the metric; while white cells denote a non-statistically significant correlation (at $\alpha \leq 0.05$).

the highest correlation with the human evaluation, although the differences were statistically non-significant. Therefore, we have not enough evidence to conclude which metric is better for evaluating this domain. The results for the E-Com domain were alike, having all metrics a significant correlation with humans.

On the other hand, for the XRCE domain the correlation of automatic metrics with humans were considerably lower than in the previous task. Moreover, several metrics (NIST, BEER and PER) are unable to properly correlate with humans.

Another interesting result is shown at Fig. 3. We computed a heatmap cluster of the correlation across all metrics. For all domains, the figures are divided in two main clusters. The first one, refers to n-gram-based metrics, such as BLEU, NIST, BEER, METEOR and CHRF. In the second cluster, we find error-based metrics, TER, WER and TER.

This indicates that the n-gram based metrics and error-based metrics assess different aspects of the translation quality. Provided that, both n-gram based and error-based metrics were able to correlate well with human criteria, we therefore recommend to always provide at least one metric from each family, when reporting results of translation quality.

6.2 Evaluating the translation technology

We are interested not only in the correlation across domains, but also in the behaviors of the different MT systems. We deepen in our analysis, studying each system separately. Fig. 4 shows the correlation results for all metrics according to each domain and MT system.

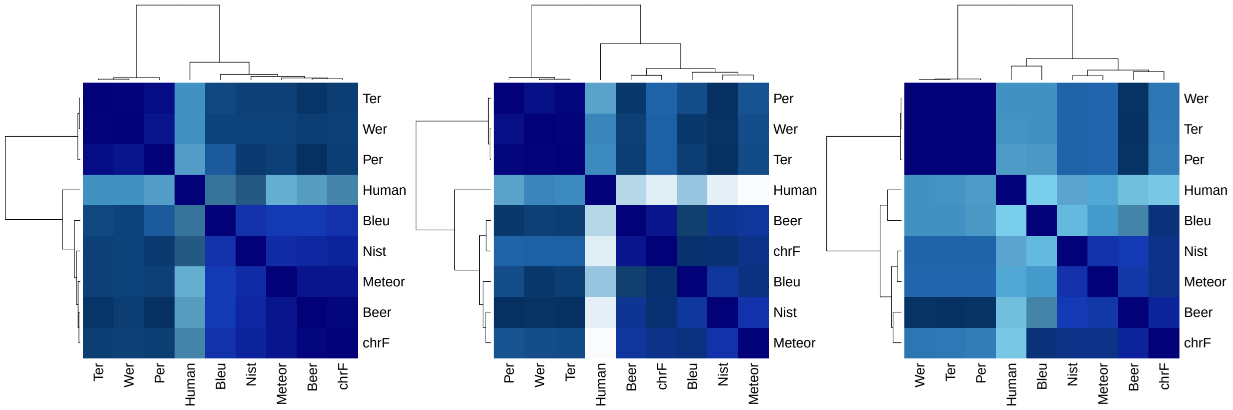
As in the previous section, we found the most reliable behavior in the IT domain. Most automatic metrics were able to properly correlate with the human criteria. However, the correlations of neural-based system were higher than those obtained by Moses. In this case, the highest correlation values were found in the NMT+Syn system, in all cases greater than 0.6.

The XRCE domain presented bad results. In this case, all the metrics failed to measure the human criteria. Only BLEU for the NMT system, with and without synthetic data, was able to properly correlate with the human assessment.

In the E-Com domain, we observed mixed results. The automatic metrics were able to correctly assess the NMT outputs, but failed with Moses. In this latter case, BLEU was the only metric that correlated well with the human evaluation.

These results suggest that automatic evaluations of NMT systems (either including synthetic data or not) were systematically more reliable than the evaluation of Moses. These differences were especially dramatic as the domain contained more sentences without large contexts nor complex syntactic structures (i.e. XRCE and E-Com). The metrics provided more reliable results for the neural systems; although they can also diverge from the human criteria.

With the addition of synthetic data to NMT systems, the correlation of metrics with respect to the human assessment slightly decreased, especially in the E-Com domain. Note that this domain was



(a) IT domain.

(b) XRCE domain.

(c) E-Com domain.

Figure 3: Clustered metrics according to their correlation for the different domains (IT, XRCE, E-Com). Blue cells denote statistically significant correlation between two metrics ($\alpha \leq 0.05$).

greatly benefited from the addition of synthetic data (Table 3). While it seems that including synthetic data effectively improves the systems, these increases should be taken with caution.

Finally, it should be noted that BLEU was the metric that correlated best with human criteria in cases involving short and simple sentences. However, in domains containing sentences with more complex syntactic structures and longer contexts, BLEU is surpassed by several metrics, like TER or CHRF .

7 Conclusions

In this work, we studied the behavior of automatic metrics in several translation systems for different domains. Since the metrics provided contradictory results, we conducted a human evaluation, based on the DA protocol. Next, we computed the correlation of the automatic metrics with respect to the human criteria.

Our findings were that automatic metrics were closer to the human as more structured and contextual the task was. When evaluating tasks with short sentences (e.g. samples from a printer manual), the correlation of the automatic metrics with respect to the human greatly fell. We also found that the automatic metrics evaluated surprisingly well NMT systems, while failing in the evaluation of classical PB-SMT systems.

Finally, we also found that the metrics were clustered, even in these specific domains, according to their nature, n-gram-based or error-based. Therefore, we recommend to always give error-based and n-gram-based metrics when reporting results on MT quality.

As future work, we intend to develop a metric capable to complement the existing ones, especially when dealing with the aforementioned short and simple corpora.

Acknowledgements: We acknowledge the users who performed the human assessment in an altruistic way. The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEO/2018/004. We also acknowledge NVIDIA for the donation of a GPU used in this work.

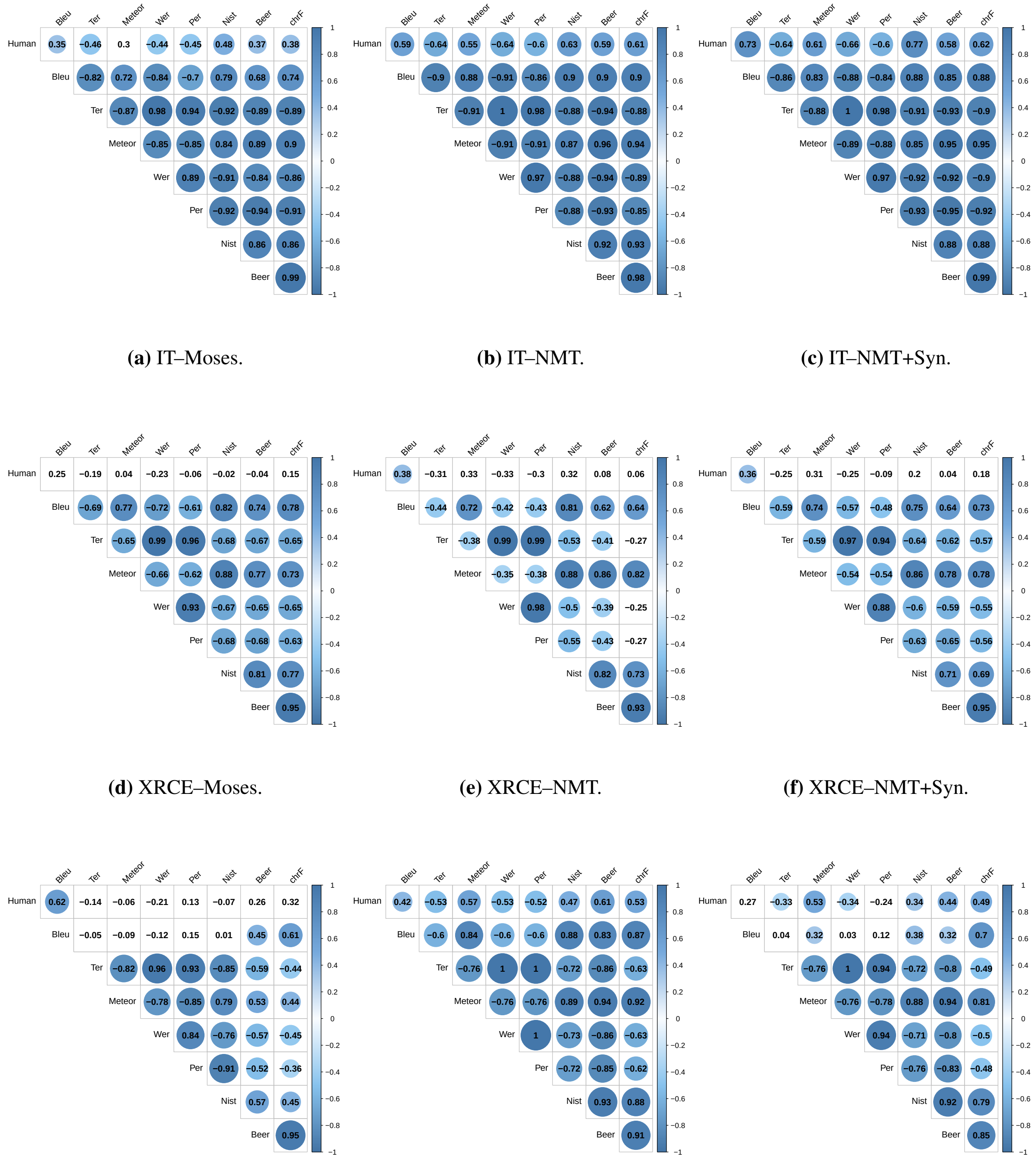
References

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., and Vilar, J.-M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2018). Neural versus phrase-based mt quality: An in-depth analysis on english–german and english–french. *Computer Speech & Language*, 49:52–70.

Bojar, O., Graham, Y., and Kamran, A. (2017a). Results of the wmt17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 489–513.

Bojar, O., Graham, Y., and Kamran, A. (2017b). Results of the wmt17 metrics shared task. In *Proceedings of the Second Conference on Ma-*



(a) IT–Moses.

(b) IT–NMT.

(c) IT–NMT+Syn.

(d) XRCE–Moses.

(e) XRCE–NMT.

(f) XRCE–NMT+Syn.

(g) E-Com–Moses.

(h) E-Com–NMT.

(i) E-Com–NMT+Syn.

Figure 4: Metric correlations for each system (Moses, NMT, NMT+Syn), for all domains (IT, XRCE, E-Com).

chine Translation, Volume 2: Shared Task Papers, pages 489–513.

Britz, D., Goldie, A., Luong, T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv:1703.03906*.

Chinea-Rios, M., Peris, Á., and Casacuberta, F. (2017). Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the International Conference on Human Language Technology Research*, pages 138–145.

Graham, Y., Baldwin, T., Moffat, A., and Zobel, J. (2017). Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Klakow, D. and Peters, J. (2002). Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1-2):19–28.

Klubička, F., Toral, A., and Sánchez-Cartagena, V. M. (2017). Fine-grained human evaluation of neural versus phrase-based machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):121–132.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Lavie, A. and Denkowski, M. J. (2009). The METEOR metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115.

Mauro, Cettolo, F., Luisa, B., Jan, N., Sebastian, S., Katsuitho, S., Koichiro, Y., and Christian, F. (2017). Overview of the IWSLT 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Morris, A. C., Maier, V., and Green, P. (2004). From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2765–2768.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Pierce, J. R. and Carroll, J. B. (1966). Language and machines: Computers in translation and linguistics.

Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Smith, A., Hardmeier, C., and Tiedemann, J. (2016). Climbing mount BLEU: The strange world of reachable high-BLEU translations. *Baltic Journal of Modern Computing*, 4(2):269.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231.

Stanojević, M. and Sima'an, K. (2014a). Evaluating word order recursively over permutation-forests. In *Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 138–147.

Stanojević, M. and Sima'an, K. (2014b). Fitting sentence level translation evaluation with many dense features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206.

Stanojević, M. and Sima'an, K. (2017). Alternative objective functions for training mt evaluation metrics. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 20–25.

Tatsumi, M. (2009). Correlation between automatic evaluation metric scores, post-editing speed, and some other factors. In *Proceedings of the Machine Translation Summit*, pages 332–339.

Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., and Sawaf, H. (1997). Accelerated dp based search for statistical translation. In *Fifth European Conference on Speech Communication and Technology*.

Toral, A. and Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1063–1073.

Turian, J. P., Shea, L., and Melamed, I. D. (2003). Evaluation of machine translation and its evaluation. In *Proceedings of the Machine Translation Summit*, pages 386–393.

White, J., OConnell, T., and OMara, F. (1994). The arpa mt evaluation methodologies: evolution, lessons, and future approaches. In *Proceedings of the 1994 Conference, Association for Machine Translation in the Americas*, pages 193–205.

Creating the best development corpus for Statistical Machine Translation systems

Mara Chinea-Rios[1] **Germán Sanchis-Trilles**[2] **Francisco Casacuberta**[1]

[1] Pattern Recognition and Human Language Technology Research Center
Universitat Politècnica de València, València, Spain
{machirio, fcn}@prhlt.upv.es
[2] Sciling, València, Spain
sanchis@sciling.es

Abstract

We propose and study three different novel approaches for tackling the problem of development set selection in Statistical Machine Translation. We focus on a scenario where a machine translation system is leveraged for translating a specific test set, without further data from the domain at hand. Such test set stems from a real application of machine translation, where the texts of a specific e-commerce were to be translated. For developing our development-set selection techniques, we first conducted experiments in a controlled scenario, where labelled data from different domains was available, and evaluated the techniques both with classification and translation quality metrics. Then, the best-performing techniques were evaluated on the e-commerce data at hand, yielding consistent improvements across two language directions.

1 Introduction

Tuning is a critical step in every system that presents a weighted combination of features. By adjusting the weights so that they best fit the target distribution, this process typically yields important improvements on the performance of the system developed. However, selecting an appropriate development set is key for this process to reach its goal.

In Statistical Machine Translation (SMT), the tuning step implies optimizing the log-linear weights $\{\lambda_1 \dots \lambda_m \dots \lambda_M\}$ of a discriminative model that implements a weighted combination of features $\{h_1 \dots h_m \dots h_M\}$, considered relevant in the translation process:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x}, \mathbf{y}) \qquad (1)$$

with $\mathbf{x}$ and $\mathbf{y}$ being the source and target sentences.

Such optimization has become de-facto standard in SMT, thanks to the wide-spread adoption of tuning algorithms such as Minimum Error Rate Training (MERT) (Och, 2003) or the Margin Infused Relaxed Algorithm (MIRA) (Cherry and Foster, 2012). The purpose of these algorithms is to adjust the log-linear weights such that the model distribution best fits the target distribution, or the target metric by which the system is evaluated.

Given that the amount of weights λ_m is typically around 10 or 20, the size of the development corpus required for tuning is typically in the range of hundreds or a few thousands of sentences. However, such corpus is typically required to be disjoint from the training corpus, used to estimate the features h_m, and its selection is critical, having an important impact on the system's performance if the development set of choice is too different from the test set at hand (Koehn, 2010).

The Data Selection (DS) task is stated as the problem of selecting the best sub-corpus of sentences from an available pool of sentences, with which to train a machine learning system. This paper deals with DS, but here the aim is to select, out of an available pool of sentences, the best development corpus for a given test set, for the purpose of log-linear weight optimization in SMT.

We study our development DS techniques in two different tasks. In the first case, the purpose is to

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 99–108
Alacant, Spain, May 2018.

analyse the behaviour of our techniques in a controlled scenario where the data is labelled according to domain. The goal is to study our methods' capacity of correctly predicting the domain labels, in addition to the translation quality achieved. In the second scenario, we evaluate the techniques presented in a real task, where a specific test set belonging to the texts of a real e-commerce site is provided, without domain labels.

The main contributions of this paper involve the necessary steps required to assess our novel development set selection techniques:

- We propose three different development DS (DDS) techniques: LD-DDS computes the Levenshtein Distance between the candidate sentences and the test sentences (Section 3); TF-DDS is based on the *t*erm frequency – inverse document frequency, which can be seen as a way of computing a numeric representation for a sentence (Section 4.1); lastly, CVR-DDS leverages a vector-space representation of sentences, relying on word the embeddings by (Mikolov et al., 2013) (Section 4.2).
- We study our DDS techniques in a controlled scenario, where domain labels are available (Section 5.2).
- We validate our DDS techniques in a real e-commerce translation task, with results that improve over random selection (Section 5.3).

This paper is structured as follows. Sections 3 and 4 present our different DDS methods. Section 5 presents the experiments: in Section 5.2, we present the analysis derived from the controlled experiment; Section 5.3 presents the results achieved with the real e-commerce task. Section 2 summarises related work. Conclusions and future work are discussed in Section 6.

2 Related works

The work presented here is close in concept to the domain adaptation scenario. Domain adaptation in SMT systems received considerable attention from the research community. Different domain adaptation techniques, including data selection, mixture models, etc., have been developed for different scenarios. A wide variety of data selection methods have been used over the years, where the main principle is to measure the similarity of sentences from the out-of-domain corpus to some in-domain corpus, either the development or the (source side of the) test set. Such similarity is often based on information theory metrics, like perplexity or cross entropy. In the last years, perplexity-based, or cross-entropy based, methods have become more common (Moore and Lewis, 2010; Axelrod et al., 2011; Rousseau, 2013; Schwenk et al., 2012; Mansour et al., 2011). Cross-entropy difference is a typical and well-established ranking function. Techniques based on information retrieval have also been widely used for data selection (Hildebrand et al., 2005; Lü et al., 2007). Furthermore, (Duh et al., 2013) leveraged neural language models to perform DS, reporting substantial gains over conventional n-gram language model-based DS. Finally, many researchers have used convolutional neural networks (CNN) in the domain adaptation field (Chen and Huang, 2016; Chen et al., 2016; Peris et al., 2016).

All the above DS approaches assume that the selection corpus is used to train or combine the SMT models. However, previous research on selecting the appropriate development corpus also exists. Such research can be split into two categories: in the first category, a development set is chosen, from among several "closed" development sets, based on the test set at hand (transductive learning) (Li et al., 2010; Zheng et al., 2010; Liu et al., 2012). The second category deals with the problem without knowing the test set beforehand, but knowing the domain of the test set (inductive learning). Previous work on development data selection for unknown test sets includes (Hui et al., 2010; Song et al., 2014). Note that the work presented here has an important difference with both transductive and inductive learning: even though it is closer to the transductive learning setting, all these works are based on selecting the most adequate development corpus from a collection of "closed" development corpora, with the purpose of choosing the one that belongs to the test set domain. In our case, we want to construct a specific development corpus for a given test corpus, without knowing the domain of the test set.

3 Levenshtein Distance DDS

The first DDS technique proposed involves computing the edit distance (Levenshtein Distance) between a candidate sentence and the closest sentence in the test set. Here, the intuition is to consider that a given sentence to be included in the development set D is a good candidate if it is not too far away from the sentences in the test set T, as measured by the Levenshtein Distance. We will refer to this technique as LD-DDS.

The Levenshtein Distance (LD) (Levenshtein, 1966) is a string metric for measuring the difference between two sequences (words or sentences). The LD between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to make them match.

Algorithm 1 shows the procedure. Here, P is the pool of sentences available, $[\mathbf{x}_p, \mathbf{y}_p]$ is an out-of-domain sentence pair ($[\mathbf{x}_p, \mathbf{y}_p] \in P$), and $|P|$ is the number of sentences in P. Then, our objective is to select data from P such that it is the most suitable for translating data belonging to the test corpus T (composed only of source sentences).

```
Data: pool $P$; test data $T$; threshold $\tau$
Result: Development corpus $D$
forall t in $T$ do
    forall $[\mathbf{x}_p, \mathbf{y}_p]$ in $P$ do
        if $LD(\mathbf{t}, \mathbf{x}_p) \leq \tau$ then
            if $[\mathbf{x}_p, \mathbf{y}_p] \notin D$ then
                add $[\mathbf{x}_p, \mathbf{y}_p]$ to $D$
            end
        end
    end
end
```

Algorithm 1: Pseudo-code for LD-DDS.

Algorithm 1 introduces the $LD(\cdot,\cdot)$ function, which computes the LD between two given sentences. Note that threshold τ establishes the size of the development corpus, and will need to be fixed empirically (Section 5.2).

4 DDS with vector-space representations

Here, we present two other DDS selection techniques, where the common point is that they both leverage a continuous vector-space representation of the sentences involved. First, we will describe our technique in abstract terms, and then we will present two different candidates for obtaining a continuous vector-space representation $F(\mathbf{x})$ (or $F_{\mathbf{x}}$ for short) of a given sentence $\mathbf{x}$.

Here, the intuition is to select as candidate sentences those whose vector-space representation is similar to those in the test set, assuming that similar sentences will have similar vectors.

The advantage of having a continuous vector-space representation of the test sentences is that a centroid can be computed, which can be assumed to be a sort of prototype of the sentences present in the test set. Note it was not possible to compute such centroid in the case of LD-DDS (Section 3).

Perhaps the best way to explaining this intuition is graphically, as shown in Figure1. This figure is a graphical example of the idea that we follow in this section, where sentences are represented in a two-dimensional vector-space. Here, blue points are the representation of the test sentences and red points represent the vectors of the sentences of the available pool of sentences, from which the development set is to be selected. Assuming that similar sentences will have a similar vector-space representation, the vectors of the test corpus will be very closer to each other, but the vectors for the general pool of sentences will be more disperse. The idea in our method is to draw a circle boundary, containing all test-sentences within it, and (hopefully) only a few of the sentences in the candidate pool. The radius of this circumference (or hyper-sphere in a multi-dimensional vector-space) is established as the distance between the centroid of the test set, and the furthest of the test sentences.

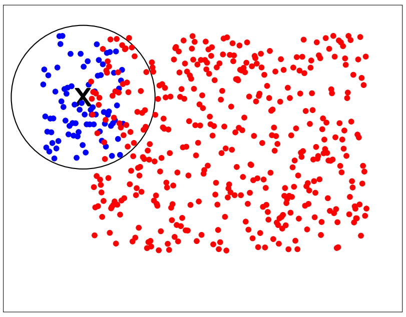


Figure 1: Graphical representation of the intuition behind our vector-space selection techniques. Red points represent the development sentence vectors, blue points represent the test sentence vectors. X is the centroid for the test vectors and the circumference represents the boundary obtained.

Algorithm 2 shows the procedure. Here, P is the pool of candidate sentences, $[\mathbf{x}_p, \mathbf{y}_p]$ is a candidate sentence pair, with $[\mathbf{x}_p, \mathbf{y}_p] \in P$, $F_{\mathbf{x}}$ is the vector-space representation of $\mathbf{x}$, and $|P|$ is the number of sentences in P. Then, our objective is to select data from P such that it is the most suitable for translating data belonging to the source test data T. For this purpose, we define $F_{\mathbf{t}}$ as the vector-space representation of a sentence $\mathbf{t} \in T$.

Algorithm 2 introduces several functions:

- *centroid*$(\cdot)$: calculates centroid $F_T = \{F_{T1} \dots F_{Tz} \dots F_{TZ}\}$ for test corpus T, as-

```
Data: Pool $P$; test data $T$
Result: Development corpus $D$
$F_T = centroid(T)$; $\rho = inf$
forall $\mathbf{t}$ in $T$ do
    if $cos(F_{\mathbf{t}}, F_T) \leq \rho$ then
        $\rho = cos(F_{\mathbf{t}}, F_T)$
    end
end
forall $[\mathbf{x}_p, \mathbf{y}_p]$ in $P$ do
    if $cos(F_{\mathbf{x}_p}, F_T) \geq \rho$ then
        add $[\mathbf{x}_p, \mathbf{y}_p]$ to $D$
    end
end
```

Algorithm 2: Pseudo-code for DDS leveraging vector-space representations of sentences.

suming a Z-dimensional vector-space:

$$F_{Tz} = \frac{1}{|T|} \sum_{t}^{|T|} F_{tz} \tag{2}$$

- $cos(\cdot, \cdot)$: the cosine similarity between two different vectors, e.g.:

$$cos(F_{\mathbf{t}}, F_T) = \frac{F_{\mathbf{t}} \cdot F_T}{\|F_{\mathbf{t}}\| \cdot \|F_T\|} \tag{3}$$

In addition, ρ represents the radius of the circumference, which is computed in lines 2 to 6 (the first **forall** loop) in Algorithm 2.

Once the selection algorithm has been established, we need to define how to represent sentences in a Z-dimensional space. Using vector-space representation for textual data (word, sentence or document) is not a new idea and has been widely employed in a variety of NLP applications. These representations have recently demonstrated promising results across a variety of tasks.

In this paper, we used two different approaches for representing sentences in a continuous vector-space: the popular *term frequency – inverse document frequency* (TF-IDF), and sentence embeddings (Mikolov et al., 2013). The basic idea is to represent a sentence $\mathbf{x}$ with a real-valued vector of some fixed dimension Z, i.e., $F(\mathbf{x}) \in R^Z$ that is able to capture similarity (lexical, semantic or syntactic) between a given pair of sentences.

4.1 TF-IDF representation

The TF-IDF (Term Frequency and Inverse Document Frequency) values can be used to create vector representations of sentence or documents. Using this kind of representation in a common vector-space is called vector space model (Salton et al., 1975), which is not only used in information retrieval but also in a variety of other research fields like machine learning (i.e. clustering, classification, information retrieval).

Each sentence $\mathbf{x} \in P$ is represented as a vector $F_{\mathbf{x}} = (F_{\mathbf{x}1}, \ldots, F_{\mathbf{x}k}, \ldots, F_{\mathbf{x}|V|})$, where $|V|$ is the size of the vocabulary V. Then, each $F_{\mathbf{x}k}$ is calculated as follows:

$$F_{\mathbf{x}k} = tf_{\mathbf{x}k} \cdot log(idf_k) \tag{4}$$

where $tf_{\mathbf{x}k}$ is the Term Frequency (TF), computed as the raw frequency of word $\mathbf{x}_k$ in a sentence, i.e. the number of times that word $\mathbf{x}_k$ occurs in sentence $\mathbf{x}$. idf_k is the Inverse Document Frequency (IDF), which is a measure of how much information word $\mathbf{x}_k$ provides, i.e., whether the term is common or rare across corpus P, computed as:

$$idf_k = \frac{|P|}{|\{\mathbf{x} \in P : \mathbf{x}_k \in \mathbf{x}\}|} \tag{5}$$

where $|P|$ is the number of sentences in corpus P, and $|\{\mathbf{x} \in P : \mathbf{x}_k \in \mathbf{x}\}|$ is number of sentences of P where the word $\mathbf{x}_k$ appears.

We will refer to the DDS technique that derives from using TF-IDF in Algorithm 2 as TF-DDS.

4.2 Continuous vector-space representation

The idea of representing words or sentence in a continuous vector-space employing neuronal networks was initially proposed by (Hinton, 1986; Elman, 1990). Continuous vector-space representations (CVR) of words or sentences have been widely leveraged in a variety of natural language applications and demonstrated promising results across a variety of tasks, such as speech recognition, part-of-speech tagging, sentiment classification and identification and machine translation just to name a few; (Schwenk et al., 2012; Glorot et al., 2011; Socher et al., 2011; Cho et al., 2014; Chinea-Rios et al., 2016).

In this paper, we use a sophisticated CVR for obtaining the representation of the sentences dealt with in our DDS method. Specifically, in (Le and Mikolov, 2014), the authors presented a CVR sentence approach. The authors adapted the continuous Skip-Gram model (Mikolov et al., 2013) to generate representative vectors of sentences or documents. *Document vectors* follow the Skip-Gram architecture to train a particular vector $F_{\mathbf{x}}$

representing the sentence or document. This work leverages the propose by (Le and Mikolov, 2014). We will refer to this representation by CVR[1], and to the DDS technique derived from using CVR in Algorithm 2 as CVR-DDS.

5 Experiments

In this section, we describe the experimental framework employed to assess the performance of the DDS methods described in Sections 3 and 4. For this purpose, we studied their behaviour in two separate tasks: a controlled scenario with labelled data, and a real e-commerce translation task. We will first detail the experimental setup employed, which is common to both tasks, and then we will report on each one of the tasks and their results.

5.1 Experimental setup

All experiments were carried out using the open-source phrase-based SMT toolkit Moses (Koehn et al., 2007). The language model used was a 5-gram with modified Kneser-Ney smoothing (Kneser and Ney, 1995), built with the SRILM toolkit (Stolcke, 2002). The phrase table was generated employing symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The log-lineal combination weights λ were optimized using MERT (Minimum Error Rate Training) (Och, 2003). Since MERT requires a random initialisation of λ that often leads to different local optima being reached, every result in this paper constitutes the average of 10 repetitions.

To study to which extent weight optimization could yield improvements in translation quality, and hence obtain an upper bound for the performance of our DDS techniques, we will also report results with a so-called *oracle*, in which tuning was performed directly using the test set. Note that this setting is not realistic, but is useful to understand how much room for improvement there is by only choosing the development set wisely.

In addition to *oracle*, two more comparative results will also be provided: *baseline*, that is obtained by a translation system where tuning was performed on the original out-of-domain data; and *in-domain*, where tuning is performed using an in-domain development set, and is hence a good reference for comparison purposes if we were to assume that such development set is not available.

Translation quality will be measured as:

- BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002) measures n-gram precision with respect to a reference set, with a penalty for sentences that are too short.
- TER (Translation Error Rate) (Snover et al., 2006) is an error metric that computes the minimum number of edits (including swaps) required to modify the system hypotheses so that they match the reference.
- METEOR (Lavie, 2014) is a precision metric that includes stemmed and synonym matches when measuring the similarity between the system's hypotheses and the references.

For the case of CVR-DDS (Section 4.2), two meta-parameters need to be fixed: $Z = 200$, the dimension of the vector-space, and $n_c = 1$, the minimum number of times a given word needs to appear in the training data for its corresponding vector to be built. These values were fixed according to preliminary research, and maintained for all the experiments reported in this paper.

5.2 Controlled scenario results

First, we conducted an assessment of our DDS methods (LD-DDS, TF-DDS, and CVR-DDS) by analyzing their performance in a controlled scenario, where domain labels were readily available. The purpose was to study to which extent the DSS techniques proposed were able to correctly classify development sentences according to some common feature, as for instance domain, by providing a test set belonging to that specific domain.

We resorted to the domain adaptation task from the Johns Hopkins Summer Workshop 2012 (Carpuat et al., 2012), where the task was to adapt French→English models. The training corpus provided originated in the parliamentary domain (Canadian Hansards). Development and test corpora included the medical domain (referred to as EMEA), the general news domain (NEWS), the press domain (PRESS), and the subtitle domain (SUBS). Statistics are provided in Table 1.

In this scenario, the development data extracted by our DDS techniques was obtained from a set where all four domain-specific development sets were merged. The *baseline* system was tuned on the Hansards development data, and the *in-domain* system was tuned on the domain-specific development data of each domain, respectively.

[1] http://radimrehurek.com/gensim/doc2vec

Table 1: Corpora used in the controlled scenario. (Dev-in) is the in-domain development set, (Test) is the evaluation set, (Training) is the training corpus and (Dev-bsln) is the baseline development set. M stands for millions and k thousands of elements; $|S|$ stands for number of sentences and $|V|$ for vocabulary size.

		EMEA		**NEWS**		**PRESS**		**SUBS**			**HANSARD**	
		$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$		$\|S\|$	$\|V\|$
EN FR	Dev-in	2022	2285 2563	2043	3682 3828	1990	4232 4583	2972	1755 1879	Training	8.1M	186.6k 191.5k
EN FR	Test	2045	2061 2274	2489	4404 4759	1982	4259 4551	3306	1980 2032	Dev-bsln	1367	24.1k 24.9k

Table 2: Precision, recall and F_1 scores for LD-DDS, TF-DDS and CVR-DDS in the controlled scenario.

		EN-FR			**FR-EN**		
Domain	System	Precision	Recall	F_1	Precision	Recall	F_1
EMEA	LD-DDS	0.35	0.33	0.34	0.37	0.32	0.34
	TF-DDS	0.16	0.32	0.21	0.16	0.32	0.21
	CVR-DDS	**0.64**	**0.47**	**0.54**	**0.74**	**0.45**	**0.56**
NEWS	LD-DDS	0.10	0.12	0.11	0.08	0.12	0.09
	TF-DDS	0.24	0.28	0.25	**0.25**	**0.60**	**0.35**
	CVR-DDS	**0.16**	**0.53**	**0.25**	0.17	0.54	0.25
PRESS	LD-DDS	0.01	0.01	0.01	0.01	0.02	0.02
	TF-DDS	0.32	0.46	0.38	0.21	0.60	0.31
	CVR-DDS	**0.38**	**0.52**	**0.47**	**0.36**	**0.47**	**0.41**
SUBS	LD-DDS	0.77	0.39	0.51	**0.81**	**0.43**	**0.56**
	TF-DDS	0.74	0.38	0.50	0.38	0.43	0.39
	CVR-DDS	**0.79**	**0.39**	**0.52**	0.74	0.39	0.51
Total	LD-DDS	0.24	0.46	0.31	0.26	0.27	0.27
	TF-DDS	0.35	0.32	0.33	0.24	0.46	0.32
	CVR-DDS	**0.37**	**0.46**	**0.41**	**0.37**	**0.45**	**0.40**

5.2.1 Precision, Recall and F_1-score

We analysed the ability of our DDS methods to recover the domain labels by providing the corresponding test set. We measured precision, recall and the F_1 measureResults are shown in Table 2, where the last row, *total*, shows precision, recall and F_1 across all domains in a 4-class confusion matrix (i.e., not the average). Several things should be noted:

- Selecting sentences using CVR-DDS obtained significantly better results than TF-DDS and LD-DDS, except for SUBS, where all methods obtained very similar results.
- The best classification quality was obtained in SUBS domain. We understand that this is because this domain has the largest test corpus, and hence yields better estimations.
- In the case of NEWS, our DDS methods obtained the worst values of precision and recall, which implies that they were not able to retrieve the correct development sentences. This seems to signal that it is not an adequate corpus for research on adaptation, as already observed in related work (Haddow and Koehn, 2012; Irvine et al., 2013).
- Finally, the results obtained for the three different methods are coherent across different language pairs (EN-FR and FR-EN).

Note that the result of LD-DDS depends on threshold τ. In Table 2 we only reported the best results obtained, which might slightly bias the results in favour of LD-DDS. However, given that LD-DDS is even so not the best DDS technique (neither in terms of classification metrics, nor in terms of translation quality), we report these results for the sake of assessing its potential.

5.2.2 SMT results

Once the quality of the selected development corpus was analysed, we now pursue establish to which extent classification metrics relate to translation quality, measuring the performance of the DDS methods in terms of BLEU (Table 3). Results with METEOR and TER presented similar

Table 3: Translation results in the controlled scenario. $|S|$ denotes number of sentences.

		EMEA		**NEWS**		**PRESS**		**SUBS**	
	System	$\|S\|$	BLEU	$\|S\|$	BLEU	$\|S\|$	BLEU	$\|S\|$	BLEU
EN-FR	*baseline*	1367	22.9	1367	21.4	1367	21.9	1367	16.6
	in-domain	1784	24.8	1467	23.9	1255	23.9	2940	18.3
	LD-DDS	1657	24.0	1772	22.5	2225	20.9	1568	18.2
	TF-DDS	1778	22.9	1718	23.5	1832	21.6	1543	18.0
	CVR-DDS	1295	24.8	3592	23.7	1724	23.8	1436	18.4
	oracle	1842	26.7	1782	24.7	1227	24.6	3281	19.1
FR-EN	*baseline*	1367	22.6	1367	21.5	1367	20.8	1367	12.3
	in-domain	1784	23.8	1467	23.0	1255	21.1	2940	18.9
	LD-DDS	1532	20.2	2418	20.6	2218	17.1	1549	18.5
	TF-DDS	3550	23.9	3563	22.6	3589	20.2	3496	14.9
	CVR-DDS	1067	24.4	4254	22.7	3754	20.9	1543	18.6
	oracle	1842	26.1	1782	23.6	1227	22.0	3281	19.5

conclusions and are omitted in this case for clarity purposes. Several conclusions can be drawn:

- All DDS methods are mostly able to improve over *baseline* across the different domains and language pairs. This seems reasonable, given that the *baseline* results were obtained using an out-of-domain development corpus for tuning purposes.
- CVR-DDS yields better translation quality than LD-DDS and TF-DDS. This seems to signal that CVR-DDS achieves a better representation of the sentences involved. However, results involving the SUBS domain yield very similar results across all three DDS methods.
- Lastly, translation quality results between CVR-DDS and *in-domain* are not significantly different. We understand that this is important since it proves the utility of our development DS method, which is able to recover a development set which is at least as well-suited for the task as the development set originally designed for that task.

5.3 Real scenario results

After analyzing the behaviour of our DDS techniques in a controlled scenario, we pursued to evaluate them in a real-world task, where no development set was readily available. For this purpose, we confronted the system with a set of sentences obtained from a real e-commerce.

For this purpose, we gathered the data from one of our customers, *Cachitos de Plata*[2], where no appropriate development set was readily available. As training data, we explored the use of three different corpora available in the Workshop on Statistical Machine Translation [3] (WMT): *1)* The Europarl (EURO) corpus, which is composed of translations of the proceedings of the European parliament; *2)* The United Nations (UN) corpus, which consists of official records and other documents of the United Nations belonging to the public domain; *3)* The Common Crawl corpus (COMMON) which was collected from web sources. Statistics of these corpora are provided in Table 4. In this case, our DDS methods were set to sample from the pool of development data available from the different years of the WMT task (*Dev* row in Table 4), and the *baseline* system was tuned according to the 2015 development data (*Dev-bsln*).

In this case, and given that no in-domain development set is available, we also considered random sampling a set of sentences from the available pool of data, in addition to *baseline* and *oracle*. We will refer to this baseline as *random*. Here, 2500 sentences were randomly sampled from the available pool of development data, without repetition. The results reported show the average of 5 repetitions of the sampling, where confidence intervals were never greater than 0.2 points (in the corresponding translation quality metric).

Results in Table 5 show the results in terms of BLEU, METEOR and TER, and development set size. In this case, we omitted both LD-DDS and TF-DDS for clarity purposes and because the results were consistent with those reported in Section 5.2. We also omitted the results obtained when using Europarl as training set, given that BLEU

[2] `http://www.cachitosdeplata.com`. In case of acceptance, this data set will be published free for research purposes, for the purpose of replicability and further research.

[3] `http://www.statmt.org/wmt16`

Table 4: Corpora main figures for real e-Commerce task. (Dev) is the pool development set, (Test) is the evaluation data, (Training) is the training corpus and (Dev-bsln) is the development corpus. Same abbreviations as in Table1.

		e-Commerce			**EURO**		**UN**		**COMMON**	
		$\|S\|$	$\|V\|$		$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$
EN ES	Test	886	874 976	Training	1.5M	88.2k 133.7k	11.2M	1.7M 893.2k	1.8M	1.9M 613.8k
EN ES	Dev	16.4k	26.0k 31.7k	Dev-bsln	2600	3691 3925	2600	3691 3925	2600	3691 3925

scores with this corpus were around 9.00 points. Several conclusion can be drawn:

- CVR-DDS achieves consistent improvements over the *baseline* translation quality, in all three metrics considered.
- CVR-DDS achieves consistent improvements over the *random* translation quality, in all three metrics, across both language pairs, and with much fewer sentences. Note that it is typically assumed that such random baseline is very tough to beat in DS and active learning research (Ananthakrishnan et al., 2010; Ambati et al., 2010), and, furthermore, improvements are statistically significant.
- Training with UN and COMMON leads to very different results. We assume this is because COMMON, even though being a smaller corpus, is more related to the domain at hand: the Commoncrawl data is crawled from the web, and in this case we are dealing with web data.

6 Conclusions

In this paper, we have presented different techniques for building a test-specific development corpus, leveraged for optimizing the log-linear weights of the SMT system. We proposed three new development data selection methods: LD-DDS, TF-DDS, and CVR-DDS. We analysed the performance of these methods in a controlled scenario, where domain labels are available, and evaluated the methods in a real translation task where e-commerce data was to be translated, without a development set being readily available. The empirical results show that CVR-DDS, which leverages a continuous vector-space representation of the sentences, is able to improve over baseline translation quality, and provide a development set that leads to similar translation quality as than the one obtained whenever an in-domain development set is readily available. In addition, the results obtained with CVR-DDS consistently and significantly improve over those obtained with a random sampling baseline, across different languages.

In the future, we will further investigate the selection of development corpus, since there is more room for improvements, as reported by the *oracle* setting. We also intend to test our methods on other domains and test data so as to establish their robustness. Finally, we are providing the e-commerce corpus *Cachitos de Plata*, used as test data, free for research purposes.

Acknowledgements: The research leading to these results were partially supported by projects CoMUN-HaT-TIN2015-70924-C2-1-R (MINECO/FEDER) and PROMETEO/2018/004.

References

Ambati, V., Vogel, S., and Carbonell, J. G. (2010). Active learning and crowd-sourcing for machine translation. In *Proc. of the LREC*, pages 2169–2174.

Ananthakrishnan, S., Prasad, R., Stallard, D., and Natarajan, P. (2010). A semi-supervised batch-mode active learning strategy for improved statistical machine translation. In *Proc. of the CoNLL*, pages 126–134.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proc. of the EMNLP*, pages 355–362.

Carpuat, M., Daumé III, H., Fraser, A., Quirk, C., Braune, F., Clifton, A., et al. (2012). Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins summer workshop final report*.

Chen, B. and Huang, F. (2016). Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proc. of the CoNLL*, pages 314–324.

Chen, B., Kuhn, R., Foster, G., Cherry, C., and Huang, F. (2016). Bilingual methods for adap-

Table 5: Translation results for real e-commerce scenario.

		EN-ES				**ES-EN**			
Training	System	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
UN	bsln	2600	13.8	42.2	67.3	2600	17.4	24.9	60.8
	random	2500	12.5	40.9	64.7	2500	18.2	27.4	60.9
	CVR-DDS	1681	15.5	42.8	64.4	1750	18.6	27.9	60.2
	oracle	886	19.3	45.6	58.3	886	21.0	28.8	58.0
COMMON	bsln	2600	21.3	49.1	57.0	2600	24.1	32.9	52.7
	random	2500	21.9	49.7	56.6	2500	22.1	33.1	52.2
	CVR-DDS	2346	22.8	50.6	56.5	1704	25.6	34.4	51.3
	oracle	886	31.1	55.7	53.3	886	33.0	37.4	43.5

tive training data selection for machine translation. In *Proc. of the AMTA*, pages 93–103.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proc. of the NAACL*, pages 427–436.

Chinea-Rios, M., Sanchis-Trilles, G., and Casacuberta, F. (2016). Bilingual data selection using a continuous vector-space representation. In *Proc. of the SPR+SSPR*, pages 95–106.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv e-prints*.

Duh, K., Neubig, G., Sudoh, K., and Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In *Proc. of the ACL*, pages 678–683.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. of the ICML*, pages 513–520.

Haddow, B. and Koehn, P. (2012). Analysing the effect of out-of-domain data on smt systems. In *Proc. of the WMT*, pages 422–432.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proc. of the EAMT*, pages 133–142.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proc. of the CogSci*, pages 12–24.

Hui, C., Zhao, H., Song, Y., and Lu, B.-L. (2010). An empirical study on development set selection strategy for machine translation learning. In *Proc. of the WMT*, pages 67–71.

Irvine, A., Morgan, J., Carpuat, M., Daumé III, H., and Munteanu, D. (2013). Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proc. of the ICASSP*, pages 181–184.

Koehn, P. (2010). *Statistical machine translation.* Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proc. of the ACL*, pages 177–180.

Lavie, M. D. A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proc. of the ACL*, page 376.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. `arXiv:1405.4053`.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(20-29):707–710.

Li, M., Zhao, Y., Zhang, D., and Zhou, M. (2010). Adaptive development data selection for log-linear model in statistical machine translation. In *Proc. of the ACL*, pages 662–670.

Liu, L., Cao, H., Watanabe, T., Zhao, T., Yu, M., and Zhu, C. (2012). Locally training the log-linear model for smt. In *Proc. of the EMNLP*, pages 402–411.

Lü, Y., Huang, J., and Liu, Q. (2007). Improving statistical machine translation performance by training data selection and optimization. In *Proc. of the EMNLP*, pages 343–350.

Mansour, S., Wuebker, J., and Ney, H. (2011). Combining translation and language model scoring for domain-specific data filtering. In *Proc. of the IWSLT*, pages 222–229.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. `arXiv:1301.3781`.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proc. of the ACL*, pages 220–224.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. of the ACL*, pages 160–167.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proc. of the ACL*, pages 311–318.

Peris, Á., Chinea-Rios, M., and Casacuberta, F. (2016). Neural networks classifier for data selection in statistical machine translation. `arXiv:1612.05555`.

Rousseau, A. (2013). Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schwenk, H., Rousseau, A., and Attik, M. (2012). Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proc. of the NAACL-HLT*, pages 11–19.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proc. of the AMTA*, pages 223–231.

Socher, R., Lin, C. C., Manning, C., and Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proc. of the ICML*, pages 129–136.

Song, X., Specia, L., and Cohn, T. (2014). Data selection for discriminative training in statistical machine translation. In *Proc. of the EAMT*, pages 45–53.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proc. of the ICSLP*, pages 901–904.

Zheng, Z., He, Z., Meng, Y., and Yu, H. (2010). Domain adaptation for statistical machine translation in development corpus selection. In *Proc. of the IUCS*, pages 2–7.

Training Deployable General Domain MT for a Low Resource Language Pair: English–Bangla

Sandipan Dandapat and William Lewis
Microsoft AI & Research
{sadandap,wilewis}@microsoft.com

Abstract

A large percentage of the world's population speaks a language of the Indian subcontinent, what we will call here *Indic* languages, comprising languages from both Indo-European (e.g., Hindi, Bangla, Gujarati, etc.) and Dravidian (e.g., Tamil, Telugu, Malayalam, etc.) families, upwards of 1.5 Billion people. A universal characteristic of Indic languages is their complex morphology, which, when combined with the general lack of sufficient quantities of high quality parallel data, can make developing machine translation (MT) for these languages difficult. In this paper, we describe our efforts towards developing general domain English–Bangla MT systems which are deployable to the Web. We initially developed and deployed SMT-based systems, but over time migrated to NMT-based systems. Our initial SMT-based systems had reasonably good BLEU scores, however, using NMT systems, we have gained significant improvement over SMT baselines. This is achieved using a number of ideas to boost the data store and counter data sparsity: crowd translation of intelligently selected monolingual data (throughput enhanced by an IME (Input Method Editor) designed specifically for QWERTY keyboard entry for Devanagari scripted languages), back-translation, different regularization techniques, dataset augmentation and early stopping.

1 Introduction

Today, machine translation (MT) is largely dominated by neural (NMT) and statistical MT (SMT), with NMT, by far, becoming the most prevalent among the two (Bahdanau et al., 2014; Bojar et al., 2017). The performance of the corpus-based approaches to MT primarily depends on the availability of corpora to train them, specifically sufficient quantities of parallel data in a given language pair. This problem is exacerbated by NMT, which generally needs larger quantities of parallel data, and has stricter requirements as to the cleanliness of that data. Unfortunately, large amounts of readily available parallel resources exist only for a small number of languages, e.g., OPUS (Tiedemann and Nygaard, 2004) and Europarl (Koehn, 2005), with very few sources of Indic language data.

While Indian languages are widely spoken (in terms of native speakers), most of these languages have very little or no parallel resources available to build a general domain MT system (Khan et al., 2017; Singh et al., 2017). In the absence of readily available parallel corpora, comparable resources are often used to extract good quality parallel data from the web (Irvine and Callison-Burch, 2013; Wołk et al., 2015). In this direction also, Indic languages have a very few comparable resources. A clear indication can be found by examining the number of Wikipedia pages available for Indic languages. We found only 57k pages are available for Bangla (no Indic Language has more than 125k pages), while a large number of European languages have more than 1 million pages. Furthermore, due to the usage of multiple fonts and encodings, a significant portion of the web data is not usable to extract useful parallel content.

One of the major problems with training an

NMT system on little data, especially when training an engine for general usage (i.e., not domain specific), is the problem of overfitting. Deep neural networks have large parameter spaces and need ample amounts of data in order to generalize adequately; with small amounts of data they tend not to generalize well. We address this overfitting issue by learning the optimizer over a smaller number of steps. Of course, adding more data always help, which is one of the benefits of synthetic data.

In this paper, we describe our English (En)–Bangla (Bn) general purpose, production quality MT systems. Bangla is the seventh most commonly spoken language in the world with an estimated reach of 215 million people in Bangladesh and the Indian subcontinent. First, we describe the SMT-based system trained on approximately 1 million parallel sentences. Bangla is a morphologically rich language, and as such, suffers from a high out-of-vocabulary (OOV) rate in a low data scenario. We address the data sparsity issue through aggressive word segmentation technique. Secondly, we build NMT models using the same parallel resources used for the SMT systems. Furthermore, we augmented a lot of synthetic training data (Sennrich et al., 2015) generated using reverse translation engine to improve the NMT systems.

The primary focus of this work is to develop general purpose MT systems for relatively low resource languages. The focuses of this work is summarized below.

- We describe our effort towards achieving a reasonably good amount of parallel data from scratch and building publicly deployed En–Bn MT systems using the same.
- We propose a novel word segmentation technique to handle the OOV words of the baseline SMT models for a morphological rich source language.
- We demonstrate how data augmentation and early stopping can be used to build a usefully deployed NMT system with less resource.
- The use of back-translated data, data filtering and controlled learning duration can effectively build deployable[1] NMT system using low resource.

The rest of the paper is organized as follows. Section 2 describes the data sets used to build the system. In Section 3, we describe the SMT and NMT models and their components. Section 4 highlights the experimental setup and results. Concluding remarks are made in Section 5.

2 Data Set

The training data used to build our systems includes both true parallel data and synthetically generated parallel data using back-translation (Sennrich et al., 2015). We use true parallel data to train both SMT and NMT systems. However, the synthetic parallel data is used to train only the NMT systems. In this section we focus on the true parallel data and describe the generation of synthetic data in Section 3.2. Altogether, we have used 1M true parallel sentences along with larger synthetic data (approximately 2.8M and 8.2M for En→Bn and Bn→En, respectively).

Data from the Web: Often many web pages are available in multiple languages. Some of these pages are sentence or paragraph aligned (less-noisy) parallel data (eg. TED talks' transcription) and some articles are comparable or noisy-parallel corpus in nature (eg. interlingually linked Wikipedia documents). We have extracted several parallel and comparable web articles for Bangla and English pair from the Web. These articles for the most are not sentence aligned. Once the potential parallel pages are extracted from the web, the sentence aligner is used to extract sentence aligned parallel text from the data. We extracted the data from the relevant file formats, and used a modified Moore Sentence Aligner to align the data (Moore, 2002).

Crowd Sourced Data: We have used Amazon's Mechanical Turk (MTurk) for crowdsourcing the English to Bangla parallel data creation task. This was primarily motivated from the work described in (Post et al., 2012). In MTurk, every task is divided into a set of Human Intelligence Task (HIT). In particular to our translation task, each HIT consists of translating 10 sentences. The two key properties of our HITS are reward amount ($0.50) and assignment duration (3 hours). Furthermore, we incorporated automated quality checking into the HITs for identifying incorrect entries made by turkers. This prunes some of the fraudulent entries and essentially reduces the manual approval time.

[1] The term deployable refers to general domain MT system that produce acceptable translation by human judges and requires low-latency.

The automatic check takes care of the following:

- The translated text should be in UTF (for Bangla)
- No sentence can be left un-translated while submitting the HIT
- The text can not have three same consecutive character other than numbers

One key issue with MTurk is to identify a set of trusted users for the desired task as a lot of turkers provide bad data, e.g., by providing nonsense content, or most frequently, unedited MT'd content. We published 2 test HITs (translate English into Bangla and Bangla into English) to find our trusted turkers based on the test HITs. The turkers whose work has been approved manually were considered as trusted turkers. We had altogether a set of 24 trusted turkers from a total of 65 submissions. Note, we integrated the Indic Language Input Tool (ILIT) into the English into Bengali HIT interface so that the turkers can easily enter Bangla text in the translation text box using a QWERTY keyboard.

Due to the small size of the trusted crowd for Bangla, it was time consuming to generate a large amount of parallel sentences using MTurk. Thus, we needed a careful selection process to choose the sentences which we wanted to translate to ensure maximum vocabulary saturation (Lewis and Eetemadi, 2013). We selected novel data based on the frequency distribution of the words in the existing parallel corpora. We ranked all the sentences in the un-translated source text based on the Equation (1) and selected the top candidates (higher score) for manual translation.

$$score(s_j) = \frac{1}{n} \sum_{\forall w_i : f_{w_i} < 10} 1 - \frac{f_{w_i}}{10} \qquad (1)$$

Here, s_j $(= \{w_i\}_1^n)$ is a candidate source sentence in the entire monolingual data, n is the total number of words in s_j. f_{w_i} is the unigram frequency of word w_i in the existing parallel corpora. We used a frequency threshold of 10 assuming that the word have occurred in a significant number of different context when it has observed frequency $(f_{w_i}) \geq 10$.

2.1 Test Data

We created 2 different test sets to evaluate our systems. Our first test set was created by selecting sentences from news articles. We took the source sentences from a Hindi newspaper (http://hindi.webdunia.com/) and translated across multiple Indian languages including Bangla and English.[2] All the test data are manually created and validated twice by human experts. We shall refer this testset as **Webdunia**.

Our second testset was created using a subset of sentences from the standard WMT2009 for English–French. 1000 English sentences were randomly selected and manually translated into Bangla by human experts. We call this test set **WMT2009**). Table 1 summarizes the different data used for training and testing.

Parallel Data	#sentences	#En	#Bn
Train	976,634	13.8	12.5
Webdunia (test set)	5,000	14.4	13.0
WMT (test set)	1,000	22.8	20.2
Dev	3,500	16.6	15.2
Monoligual Data			
English	14m	15.1	–
Bangla	13m	–	13.7

Table 1: Data set used: #En = average English sentence length, #Bn=average Bangla sentence length

3 Models

3.1 SMT Model

We have used vanilla **phrasal** (Koehn et al., 2003) and **treelet** (Quirk et al., 2005; Bach et al., 2009) translation model for Bn→En and En→Bn systems, respectively. The treelet translation uses a source-language dependency parser to extract syntactic information on the source side. The dependency parse structure is projected onto the target sentence using an unsupervised alignment of the parallel data to extract a dependency treelet[3] translation pairs (source and target treelet with word-level alignment). These dependency treelet pairs are used to train a tree-based reordering model. We use a hand-built rule-based parser for English (Heidorn, 2000). Note, that due to unavailability of a Bangla parser we do not use treelet translation system in Bn→En direction (that system is strictly phrasal).

[2] We selected Hindi as the source as we are creating the same testset across multiple Indian languages (results for the other languages are not discussed in this paper).

[3] Which is an arbitrary connected subgraph from the dependency parse tree

For both phrasal and treelet systems, word alignment is done using GIZA++ (Och and Ney, 2003) in both directions. We use the target side of the parallel corpus along with additional monolingual target language data (cf. Table 1) to train a 5-gram language model using modified Kneser–Ney smoothing (Kneser and Ney, 1995). Finally, we use MERT (Och, 2003) to estimate the lambda parameters using the held out *Dev* data with a single reference translation.

With the baseline phrasal system for Bn→En, we found 4.9% words are untranslated. We categorized these OOV words into 3 broader categories: these include unseen inflected surface forms or compounds (˜46% of the OOVs), unseen foreign words (˜40%) and numbers (˜4%). Remaining ˜9% OOVs are due to incorrect spelling of the word. We developed a *word breaker* to handle the first 46% of OOVs and use a transliteration module to transliterate foreign words. In Bangla, foreign words are often inflected with case markers (eg. accusative, locative and negative). The word breaker module also splits the suffixes from the inflected foreign words and subsequently the transliteration module will transliterate unknown foreign words. Finally, Bangla numbers (in digits) are also often inflected with specificity and/or with an intensifier. We remove these markers from the number and directly convert them into English numerals. Table 2 shows some examples of each of the aforementioned OOVs.

word	affix	type
minArgulo	*-gulo*	inflectional
bhAShAi	*-i*	clitic
rachanAkAla	*-kAla*	compounding
bhumikendrika	*-kendrika*	derivational
negalijensa	-	foreign word
lakera	*-era*	inflectional foreign word
507ti	*-ti*	inflectional
5i	*-i*	clitic

Table 2: Example OOVs

Word Breaker: We develop an aggressive suffix splitter to handle OOVs resulting from the morphological richness of Bangla. This is motivated by the work reported in (Koehn and Knight, 2003). Koehn and Knight (2003) used monolingual and parallel corpora to identify the potential splitting options of a word. In contrast, we use linguistic suffix list to find the candidate splits and use parallel corpora to rank these candidate splits based on the frequency of the non-affix part. This frequency is the raw frequency estimated from the surface form words in the parallel data. Algorithm 1 shows the detail of the word breaker.

Algorithm 1 wordbreaker(w, V, S)

```
In: input word $w$,
parallel corpus vocabulary with frequency $V$=$\{< v_i, f_i >\}_1^m$,
list of suffixes $S$=$\{s_i\}_1^n$
Out: best split $b$
1:  $C = \{(w, \phi)\}$ {candidate split}
2:  $mw = 2$ {minimum word length}
3:  for $i := length(w) - 1$ to $mw$ do
4:      split $w$ into $w_r$ and $s$ at position $i$
5:      if inVoc$(w_r, V)$ and isComposable$(s, S)$ then
6:          $C = C \cup (w_r, s)$
7:      end if
8:  end for
9:  sort $C$ based on frequency $f(w_r)$ {based on the vocabulary V}
10: $(w'_r, s') \leftarrow top(C)$
11: {suff} $\leftarrow$ decompose$(s', S)$
12: $b \leftarrow (w_r, \{suff\})$
```

Line 3-6 split the surface word recursively into potential subwords and affixes. The main intuition behind the split is to chop the word until a known subword is found from the parallel data with a set of valid suffixes. Line 5 of the algorithm finds if the subword (w_r) lies in the vocabulary of the parallel corpus to ensure after split we will be able to translate the w_r part. The *isComposable*() function checks if the suffix s is a concatenation of multiple suffixes which is further decomposed into multiple suffixes in line 11 using *decompose*() function. We have used 55 different suffixes (S) and 152K surface words with their frequency (V). The suffix list includes common affixes (both inflectional and derivational) like *'gulo'*, *'bhAbe'*, *'ke'* and also some very productive compounding cases like *'kAla'*, *'samAja'* etc. We use the word breaker during training (parallel data) and decoding time (test sentence). Note that one of the candidate split includes the surface form (line 1 of the Algorithm) of the word. This ensures that the already observed (in the parallel data) surface forms may not required a split unless we found one of its potential split (w_r) with higher occurrence in the data.

3.2 NMT Model

Our NMT model is developed based on the architecture described in (Devlin, 2017). The encoder uses a 3-layer bi-directional RNN (consists of 512 LSTM units). The decoder uses an LSTM layer in the bottom to capture the context and the attention. The LSTM layer is then followed by 5 fully-connected layers applied in each timestamp using a ResNet-style skip connection (He et al., 2016). The details of the model and equations are described in (Devlin, 2017). The model pre-computes part-of the first hidden layer offline. Additionally, the embedding layer (Devlin et al., 2014) is fed into multiple hidden layers (Devlin et al., 2015) to pre-compute all of them independently. These multiple hidden layers are placed next to each other to avoid stacked network and used for lateral element combination. This is the best known model to balance the trade-off between latency and accuracy of NMT system.

Due to very small amount of training data (approximately 1M parallel sentences), the vanilla NMT model does not find any improvement over the SMT model described in the previous section. We use synthetic data (2.8M and 8.2M for En→Bn and Bn→En, respectively), byte pair encoding and early stopping (lesser number of epochs) to significantly surpass the SMT accuracy.

All of our NMT systems use early stopping. Early stopping is done to reduce the number of training steps by monitoring the performance on the validation set. We select the model which has the lowest perplexity on the validation set. All the models are trained using ADAM optimizer (Kinga and Adam, 2015) with a dropout rate of 0.25. The optimizer uses 100K and 500K steps with a batch size of 1024 for En→Bn and Bn→En baseline NMT systems, respectively.

Synthetic data: We create synthetic parallel data by pairing monolingual (target side) data with back-translated data, which is created using a reverse translation engine. For this, we used our initial baseline NMT systems for back-translation.[4]

This is an effective way of increasing parallel content for an NMT system. While SMT system uses a separate language model using monolingual corpora, the back-translation technique has shown effective means to improve quality as compared to other techniques of incorporating monolingual data into NMT models (eg. deep fusion, null source) (Gulcehre et al., 2015). For example, we have used En→Bn baseline NMT system to translate English monolingual corpus into Bangla. The back-translated Bangla and original English sentence pairs are then used as synthetic parallel data into the Bn→En NMT system. This essentially ensures that the decoder observes error free target side data (from monolingual corpus) while the input can have errors caused by the reverse MT system. Similarly, we also create synthetic data for En→Bn NMT system using the Bangla monolingual corpus.

We found that the back-translation quality varies widely across sentences. Thus, we filter poor quality back-translated sentences using a pseudo fuzzy match (PFS) score (He et al., 2010) to rank all the back-translated output. First, the reverse translation engine (e.g., En→Bn) to translate monolingual target sentence (t) into a back-translated source (s). Then the back-translated s is further translated into t' using the forward (eg. Bn→En) baseline translation engine which we are trying to improve through back-translation. Equation 2 computes the PFS between t and t'.

$$PFS = \frac{EditDistance(t, t')}{max(|t|, t'|)} \quad (2)$$

We have selected all back-translation pairs with $PFS \leq 0.3$. Table 3 summarizes the detail of the synthetic data used to train the NMT systems.

Corpus	#sentences	#En	#Bn
En_{synth}, Bn_{mono}	2.8m	11.9	12.4
Bn_{synth}, En_{mono}	8.2m	15.7	12.9

Table 3: Synthetic data

After adding synthetic data, we train the ADAM optimizer with 200k steps with a batch size of 4096.

In the case of Bn→En NMT system, source-side Bangla sentences are represented using byte-pair encoding (BPE) (Sennrich et al., 2015) to reduce the data sparsity problem, which uses 50,000 merging operations. In addition, we use a list of 15,000 Bangla names which are not converted into a subword representation.

[4] Although the baseline SMT system has higher BLEU score but we have found that the relatively lower accuracy baseline NMT system performs better when used to generate back-translated data.

4 Experiment and Results

First we conduct different experiments with the SMT systems and compare the same with online (**Online-A**) En–Bn systems. The baseline SMT experiments uses vanilla **phrasal** and **treelet** systems for Bn→En and En→Bn, respectively. Furthermore, we conduct two different experiments using a word breaker (**+wordbreak**) and transliteration (**+trans**) in Bn→En direction. Note, we have not used transliteration in En→Bn direction. We used BLEU (Papineni et al., 2002) for automatic evaluation of our MT systems. Table 4 compares the different SMT systems with respect to baseline and Online-A system.

	Bn→En		En→Bn	
	Webdunia	WMT	Webdunia	WMT
Phrasal	13.62	14.57	–	–
Treelet	–	–	7.41	6.32
+trans	13.54	14.29	–	–
+wordbreak	16.56	16.16	–	–
Online-A	23.31	22.26	8.61	7.29

Table 4: SMT system comparison

We found that the use of transliteration does not improve BLEU score although it prevents information loss. However, the use of word breaker significantly improve the BLEU score and also reduces the number of OOV words which were all transliterated previously. We found an absolute improvement of 2.91 and 1.59 BLEU points over the baseline phrasal system, respectively, for Webdunia and WMT testsets. Figure 1 shows the reduction in OOVs using word breaker.

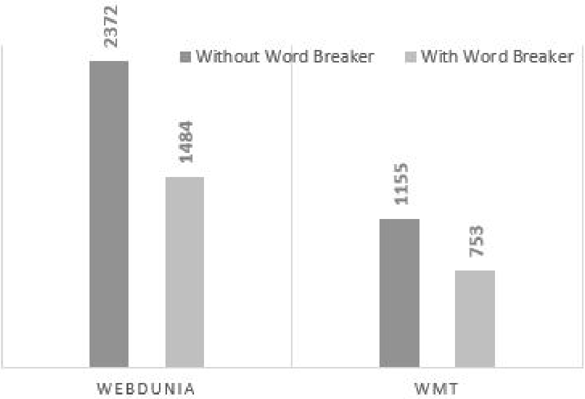


Figure 1: OOV reduction through word breaker

In our second set of experiments, we conducted different experiments using an NMT system. We conduct three different experiments with a neural system: (1) Baseline **NMT** system with early stopping; (2) synthetic data augmentation (**+Synth**) using back-translated data; and (3) using sub-word representation (**+BPE**).

	Bn→En		En→Bn	
	Webdunia	WMT	Webdunia	WMT
Final SMT	16.56	16.16	7.41	6.32
Online-A	23.31	22.26	8.61	7.29
NMT	14.51	13.46	7.24	7.16
+Synth	20.23	19.12	9.73	9.22
+BPE	19.87	20.64	9.51	9.80
Δ_{SMT}	**+3.31**	**+4.48**	**+2.1**	**+3.48**
$\Delta_{Online-A}$	-3.44	-1.62	**+0.9**	**+2.51**

Table 5: NMT System comparison. Δ_x indicates the change in BLEU score of the +BPE system with respect to x.

Table 5 shows the detail accuracies of different NMT systems. We found that the baseline NMT systems in general has lower accuracy (except WMT testset in En→Bn direction) compared to our SMT systems. In some cases (in WMT testset for Bn→En and in Webdunia for En→Bn translation) NMT system has lower accuracy than vanilla SMT systems. However, the use of synthetic data improves the systems significantly ($p < 0.05$)[5] across all testsets. We found that the use of synthetic data (+synth) has 5.72 and 5.66 absolute BLEU points improvement for Webdunia and WMT testsets in Bn→En translation over the base line NMT systems, respectively. In En→Bn direction, the use of synthetic data gives an improvement of 2.49 and 2.06 absolute BLEU points over the baseline NMT, respectively for Webdunia and WMT testsets.

The use of synthetic data also shows improvement over our final SMT systems. We found an absolute improvement of 3.67 and 2.96 BLEU points over the baseline phrasal Bn→En system, respectively for Webdunia and WMT testsets. Similarly, we found an absolute improvement of 2.32 and 2.9 BLEU points over the baseline in En→Bn direction, respectively for Webdunia and WMT testsets. The use of BPE improves the performance with WMT testset, where there is little drop in BLEU score with Webdunia test set. This is due to the fact that the percentage of unknown word in WMT testset is much higher compared to Web-

[5] Statistical significance tests were performed using paired-bootstrap resampling (Koehn, 2004).

dunia. Finally, our system shows 0.9 and 2.51 absolute BLEU point improvement over the Online-A system in En→Bn direction.

4.1 Example

Figure 2 shows some cherry picked example in the Bn→En direction. Example (a) shows better word order and lexical choice in NMT compared to SMT. In example (b), the negation (*not*) is missing in the SMT output which changes the meaning completely. In example (c), NMT system accurately convey the meaning whereas the SMT system does not produces either a grammatically or a meaningful correct translation.

4.2 Human Evaluation

In addition to the above automatic evaluations, we performed a manual evaluation of the MT output to understand the translation quality from a human perspective. While manually evaluating the MT systems, we assign values from four-point scale (1 through 4, 4 is the best) representing the absolute quality of the translation. The scoring was done according to the guideline (Brockett et al., 2002) mentioned in Table 6.

1≡**Unacceptable**	Absolutely not comprehensible and/or little or no information transferred accurately
2≡**Possibly Acceptable**	Possibly comprehensible (given enough context and/or time to work it out); some information transferred accurately
3≡**Acceptable**	Not perfect (stylistically or grammatically odd), but definitely comprehensible, AND with accurate transfer of all important information
4≡**Ideal**	Not necessarily a perfect translation, but grammatically correct, and with all information accurately transferred

Table 6: Human evaluation scale

Five independent evaluators were asked to evaluate 100 randomly drawn output from both final SMT (phrasal+wordbreak for Bn→En and treelet for Bn→En) and final NMT systems (+BPE for Bn→En and +Synth for Bn→En as shown in Table 5) from both the testsets. Table 7 shows the average absolute translation quality of the two approaches in both directions. The human evaluation shows statistically significant ($p = 0.0012$) improvement of 0.2 in the absolute scale for Bn→En compared to the SMT system. Though there is no improvement in human score in En→Bn direction, but the translation produced by NMT system is much more fluent which is reflected by the improvement in the BLEU score over the SMT-based system. Overall, our human evaluation scores lies in the possibly acceptable to acceptable range for a general domain MT system developed using a small parallel data.

System	Bn → En	En → Bn
SMT	2.1	2.9
NMT	2.3	2.9

Table 7: Human evaluation score.

5 Conclusion

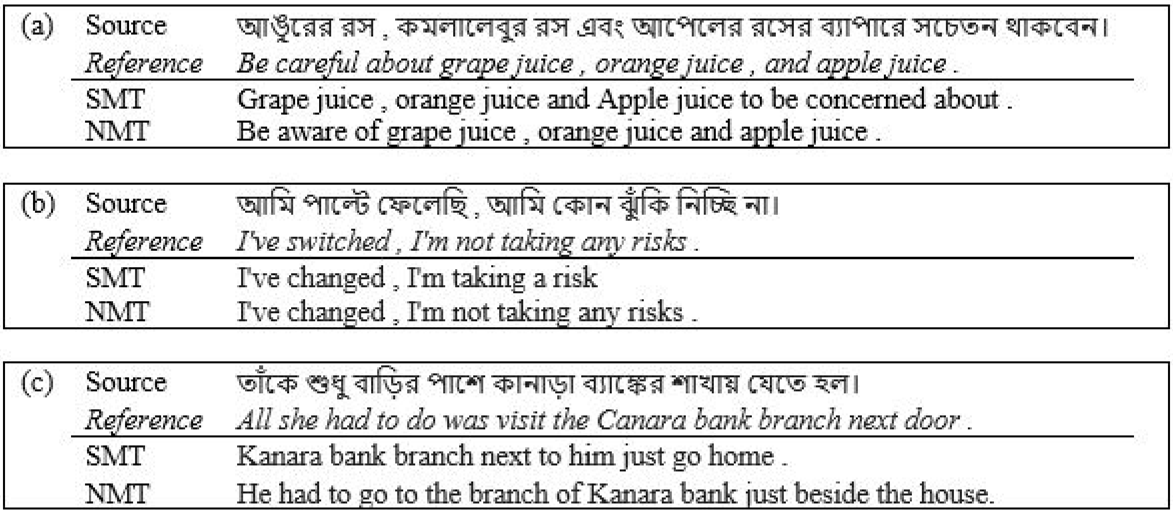
In this paper we presented En–Bn SMT and NMT systems, all of which were trained over a relatively small parallel corpus. The morphological richness of Bangla exacerbates the problem of data sparsity, and we counter this problem through a variety of techniques and tools: developing a word breaker for Bangla, generating synthetic parallel data, applying byte pair encoding (BPE) or morphological decomposition, and even crowd translating content based on vocabulary saturation data selection. Additionally, we used early stopping to prevent overfitting. The MT systems and APIs are publicly available in `https://www.bing.com/translator`. For future work, we plan to look into the integration of a word breaker into the NMT models (augmenting or replacing BPE). Also, given the success we had with data selection, specifically, vocabulary saturation for the selection of content to manually translate, we plan to explore similar or related methods of data selection to improve the quality of synthetic data that we're translating (*a la* (Junczys-Dowmunt and Birch, 2016), specifically applying (Moore and Lewis, 2010)).

References

Bach, N., Gao, Q., and Vogel, S. (2009). Source-side dependency tree reordering models with sub-tree movements and constraints. *Proceedings of the MTSummit-XII, Ottawa, Canada, August. International Association for Machine Translation.*

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M.,



(a)	Source	আঙুরের রস , কমলালেবুর রস এবং আপেলের রসের ব্যাপারে সচেতন থাকবেন।
	Reference	*Be careful about grape juice , orange juice , and apple juice .*
	SMT	Grape juice , orange juice and Apple juice to be concerned about .
	NMT	Be aware of grape juice , orange juice and apple juice .
(b)	Source	আমি পাল্টে ফেলেছি , আমি কোন ঝুঁকি নিচ্ছি না।
	Reference	*I've switched , I'm not taking any risks .*
	SMT	I've changed , I'm taking a risk
	NMT	I've changed , I'm not taking any risks .
(c)	Source	তাঁকে শুধু বাড়ির পাশে কানাড়া ব্যাঙ্কের শাখায় যেতে হল।
	Reference	*All she had to do was visit the Canara bank branch next door .*
	SMT	Kanara bank branch next to him just go home .
	NMT	He had to go to the branch of Kanara bank just beside the house.

Figure 2: Examples of $Bn \rightarrow En$ translation using SMT and NMT.

Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the 2nd Conference on Machine Translation: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Brockett, C., Aikawa, T., Aue, A., Menezes, A., Quirk, C., and Suzuki, H. (2002). English-japanese example-based machine translation using abstract linguistic representations. In *Proceedings of the 2002 COLING workshop on Machine translation in Asia-Volume 16*, pages 1–7. Association for Computational Linguistics.

Devlin, J. (2017). Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. *arXiv preprint arXiv:1705.01991*.

Devlin, J., Quirk, C., and Menezes, A. (2015). Pre-computable multi-layer neural network language models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 256–260.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1370–1380.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010). Integrating n-best smt outputs into a tm system. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 374–382. Association for Computational Linguistics.

Heidorn, G. (2000). Intelligent writing assistance. *Handbook of natural language processing*, pages 181–207.

Irvine, A. and Callison-Burch, C. (2013). Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Junczys-Dowmunt, M. and Birch, A. (2016). The university of edinburghs systems submission to the mt task at iwslt. In *Proceedings of the First Conference on Machine Translation, Seattle, USA*.

Khan, N. J., Anwar, W., and Durrani, N. (2017). Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*.

Kinga, D. and Adam, J. B. (2015). A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Lewis, W. and Eetemadi, S. (2013). Dramatically reducing training data size through vocabulary saturation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 281–291.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Post, M., Callison-Burch, C., and Osborne, M. (2012). Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Singh, S., Panjwani, R., Kunchukuttan, A., and Bhattacharyya, P. (2017). Comparing recurrent and convolutional architectures for english-hindi neural machine translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 167–170.

Tiedemann, J. and Nygaard, L. (2004). The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*.

Wołk, K., Rejmund, E., and Marasek, K. (2015). Harvesting comparable corpora and mining them for equivalent bilingual sentences using statistical classification and analogy-based heuristics. In *International Symposium on Methodologies for Intelligent Systems*, pages 433–441. Springer.

Deep Neural Machine Translation with Weakly-Recurrent Units

Mattia A. Di Gangi
FBK, Trento, Italy
University of Trento, Italy
digangi@fbk.eu

Marcello Federico
MMT Srl, Trento, Italy
FBK, Trento, Italy
federico@fbk.eu

Abstract

Recurrent neural networks (RNNs) have represented for years the state of the art in neural machine translation. Recently, new architectures have been proposed, which can leverage parallel computation on GPUs better than classical RNNs. Faster training and inference combined with different sequence-to-sequence modeling also lead to performance improvements. While the new models completely depart from the original recurrent architecture, we decided to investigate how to make RNNs more efficient. In this work, we propose a new recurrent NMT architecture, called Simple Recurrent NMT, built on a class of fast and weakly-recurrent units that use layer normalization and multiple attentions. Our experiments on the WMT14 English-to-German and WMT16 English-Romanian benchmarks show that our model represents a valid alternative to LSTMs, as it can achieve better results at a significantly lower computational cost.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015) is a sequence-to-sequence problem that requires generating a sentence in a target language from a corresponding sentence in a source language. Similarly to other language processing task, NMT has mostly employed recurrent neural networks (RNNs) (Sennrich et al., 2016b; Sennrich et al., 2017b; Luong and Manning, 2015), in both their LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014) variants, which can model long-range dependencies. Besides their simplicity, the choice of RNNs is also due to their expressive power, which has been proven equivalent to Turing Machines (Siegelmann and Sontag, 1995). RNNs have represented so far the state of the art of machine translation, and have constantly been enhanced to improve their performance. Nonetheless, their explicit time dependencies make training of deep RNNs computationally very expensive (Wu et al., 2016; Barone et al., 2017).

Recent works have proposed new NMT architectures, not based on RNNs, that obtained significant improvements both in training speed and translation quality: the so-called convolutional sequence-to-sequence (Gehring et al., 2017) and the self-attentive or transformer (Vaswani et al., 2017) models. Speed improvements by these models mainly come from the possibility of parallelizing computations over word sequences, as both models do not have time dependencies. On the other hand, performance improvements appear to be due to the path lengths needed by the networks to connect distant words in a sentence: linear for RNNs, logarithmic for convolutional models, and constant for the transformer.

In this paper we propose a neural architecture that shares some properties with the abovementioned models, while maintaining a recurrent design. Our hypothesis is that current RNNs for NMT have not been designed to take full advantage of deep structures and that better design could lead to improved performance and efficiency. Contemporary to this work, Chen et al. (2018) have shown that RNN can still outperform the transformer model when using better hyper-parameters.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 119–128
Alacant, Spain, May 2018.

We start by discussing previous efforts that proposed simplified and theoretically grounded versions of the LSTM RNN, which very recently lead to the so-called Simple Recurrent Unit (SRU). Then, we introduce our NMT architecture based on weakly-recurrent units, which we name Simple Recurrent NMT (SR-NMT). We present machine translation results on two public benchmark, WMT14 English-German and WMT16 English-Romanian, and compare the results of our architecture against LSTM and SRU based NMT, using similar settings for all of them. Results show that SR-NMT trains faster than LSTM NMT and outperforms both LSTM and SRU NMT. In particular, SR-NMT with 8-layers even outperforms Google's NMT 8-layer LSTM architecture (Wu et al., 2016). Moreover, training our model took the equivalent of 12 days on a single K80 GPU against the 6 days on 96 K80 GPUs reported by (Wu et al., 2016). Finally, the NMT architecture presented in this paper was developed in OpenNMT-py (Klein et al., 2017) and the code is publicly available on Github[1].

2 Related works

RNNs are an important tool for NMT, and have ranked at the top of the WMT news translation shared tasks (Bojar et al., 2017) in the last three years (Luong and Manning, 2015; Sennrich et al., 2016b; Sennrich et al., 2017b). Recurrent NMT is also the first approach that outperformed phrase-based statistical MT (Bentivogli et al., 2016). Despite the important results, training of RNNs remains inefficient because of an intrinsic lack of parallelism and the necessity of redundant parameters in its LSTMs and GRUs (Ravanelli et al., 2018; Zhou et al., 2016) variants. Sennrich et al. (2017b) reduce training time in two different ways: by reducing the network size with tied embeddings (Press and Wolf, 2017) and by adding layer normalization to their architecture (Ba et al., 2016). In fact, the reduction of the covariate shift produced by this mechanism shows to significantly speed up convergence of the training algorithm. Of course, it does not alleviate the lack of parallelism.

Pascanu et al. (2014) studied RNNs and found that the classical stacked RNN architecture does not have a clear notion of *depth*. In fact, when performing back-propagation through time, the gradient is sent backward in both the horizontal and vertical dimensions, thus having a double notion of depth, which also hurts the optimization procedure. They propose as a solution the notions of *deep transition*, from one hidden state to the following hidden state, and the notion of *deep output*, from the last RNN layer to the network output layer. The winning model in WMT17 actually implemented both of them (Sennrich et al., 2017b; Sennrich et al., 2017a).

Balduzzi and Ghifary (2016) proposed strongly-typed RNNs, which are variants of vanilla RNN, GRU and LSTM that respect some constraints and are theoretically grounded on the concept of *strongly-typed quasi-linear algebra*. A strongly-typed quasi-linear algebra imposes constraints on the allowed operations for an RNN. In particular, in this framework there is a constraint inspired from the type system from physics, and one inspired by functional programming. The idea of types forbids the sum of vectors generated from different branches of computation. In the case of RNNs, this means that it is not possible to sum among them the previous hidden state and the current input, as they are produced by different computation branches. The second constraint aims to simulate the distinction among *pure functions* and functions with side effects, typical of functional programming. In fact, as RNNs own a state, they can approximate *algorithms* and also produce "side effects". According to the authors, side effects manifest when the horizontal (time-dimension) connections are altered, and are the reason behind the poor behavior of techniques such as dropout (Srivastava et al., 2014) or batch normalization (Ioffe and Szegedy, 2015) when they are applied to the horizontal direction straightforwardly (Laurent et al., 2016; Zaremba et al., 2014). Thus, the side effects should be confined to a part of the network that cannot hinder the learning process. The solution they propose consists in using learnable parameters only in stateless equations (*learnware*), while the states are combined in parameterless equations (*firmware*). The combination is achieved through the use of *dynamic average pooling* (or peephole connections), which allows the network to use equations with parameters to compute the states and the gates, and then use the gate vectors to propagate forward horizontally the hidden state. The authors show theoretically that strongly-typed RNNs have generalization capabilities similar to their classical coun-

[1] https://github.com/mattiadg/SR-NMT

terparts, and confirm it with an empirical investigation over several tasks, where the strongly-typed RNNs achieve results not worse than their classical counterparts while training for less time. In addition, the absence of parameters in the state combination cancels the problem of depth introduced by Pascanu and colleagues, as these models need only the classical back-propagation and not back-propagation through time.

Quasi-recurrent neural networks (Bradbury et al., 2017) are an extension of the previous work that use gated convolutions in order to not compute functions of isolated input tokens, but always consider the context given by a convolutional window.

SRUs (Lei et al., 2017b), are a development of the units proposed by Balduzzi and Ghifary designed for training speed efficiency. The equations can be easily CUDA optimized, while a good task performance is obtained by stacking many layers in a deep network. SRUs use *highway connections* (Srivastava et al., 2015) to enable the training of deep networks. Moreover, SRUs can parallelize the computation over the time steps also in the decoder. In fact during training the words of the whole sequence are known and there is no dependency on the output of the previous time step. As for strongly-typed RNNs, the information from the context is propagated with dynamic average pooling, which is much faster to compute than matrix multiplications. SRUs were tested on a number of tasks, including machine translation, and showed performance similar to LSTMs, but with significantly lower training time. However to obtain results comparable to a weak LSTM-based NMT, SRUs require many more layers of computation. The results show that a single SRU has a significantly lower representation capability than a single LSTM. In addition, every layer adds little overhead in terms of training time per epoch, but also the results show little improvement.

In this work we further develop the idea of SRUs, and propose an NMT architecture that can outperform LSTM-based NMT.

3 Simple Recurrent NMT

We propose a sequence-to-sequence architecture that uses an enhanced version of SRUs (see Figure 1) to improve the training process, in particular with many layers, and increase the representation capability. In fact, although Lei et al. (2017b) show that they can train networks with up to 10 layers of SRUs, both in encoder and decoder, without overfitting, their results are far from the state of the art of recurrent NMT. Our design goals are addressed in a way similar to (Gehring et al., 2017; Vaswani et al., 2017). We add an attention layer within every decoder unit, and make the training more stable by adding a layer normalization layer (Ba et al., 2016) after every matrix multiplication with a parameter matrix. The layer normalization reduces the covariate shift (Ioffe and Szegedy, 2015), thus it makes easier the training of deep networks. In addition to layer normalization, our units use highway connections (Srivastava et al., 2015), which enable the training of deep networks. Our SR-NMT architecture is shown in Figure 2

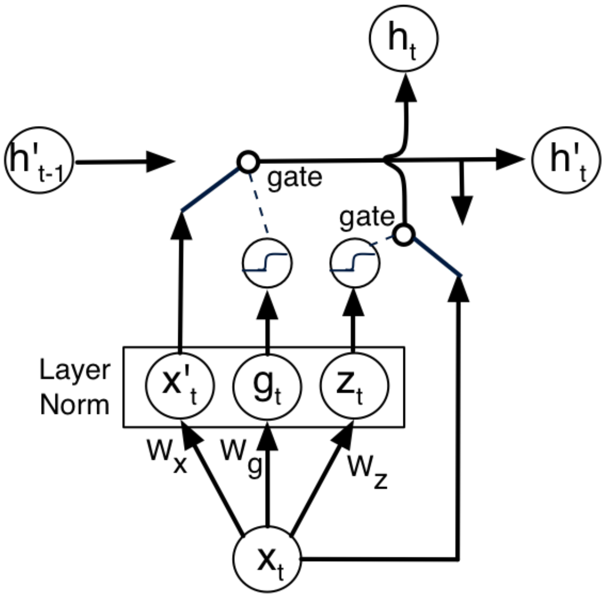


Figure 1: Core weakly-recurrent unit used in the SR-NMT architecture. Layer normalization is performed only once for all the transformations. At the end of the unit, the gate $\mathbf{z}_t$ is used for the highway connection.

Our weakly-recurrent units used in the encoder and decoder both separate *learnware* and *firmware*, although not being strongly typed (Balduzzi and Ghifary, 2016) as they include highway connections summing vectors of different types. In the following, we introduce in detail the encoder and decoder networks of our simple recurrent NMT architecture.

3.1 Encoder

Our encoder uses bidirectional weakly-recurrent units with layer normalization. We use two candidate hidden states ($\overrightarrow{\mathbf{h}}_i$, $\overleftarrow{\mathbf{h}}_i$) and two recursion

Model	train speed
LSTM 2L	3700 tok/s
SRU 3L	4600 tok/s
SR-NMT 1L	7900 tok/s
SR-NMT 2L	5500 tok/s
SR-NMT 3L	4300 tok/s
SR-NMT 4L	3600 tok/s

Table 1: Training speed comparison of our architectures with LSTM and SRU baselines on WMT14 En-De. Timings are performed on a single Nvidia Gtx 1080 GPU with CUDA 8.0 and pytorch 0.2.

gates ($\overrightarrow{\mathbf{g}}_i$, $\overleftarrow{\mathbf{g}}_i$) for the two directions. The candidate hidden state for every time step is computed as a weighted average among the current input and the previous hidden state, controlled by the two gates (peephole connections). We apply a single normalization (*LN*) for each layer to improve training convergence and impose a soft constraint among the parameters. Finally, the input of each layer is combined with its output through highway connections. Formally, our encoder layer is defined by the following equations:

$$\mathbf{x_i} \in R^d; \quad \mathbf{W} \in R^{d\times(4\frac{d}{2}+d)}$$

$$[\overrightarrow{\mathbf{x}}_i, \overleftarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{g}}_i, \overleftarrow{\mathbf{g}}_i, \mathbf{z_i}] \;=\; LN(\mathbf{x_i}\mathbf{W})$$

$$\begin{aligned}
\overrightarrow{\mathbf{h}}_i &= (1-\sigma(\overrightarrow{\mathbf{g}}_i)) \odot \overrightarrow{\mathbf{h}}_{i-1} + \sigma(\overrightarrow{\mathbf{g}}_i)) \odot \overrightarrow{\mathbf{x}}_i \\
\overleftarrow{\mathbf{h}}_i &= (1-\sigma(\overleftarrow{\mathbf{g}}_i)) \odot \overleftarrow{\mathbf{h}}_{i+1} + \sigma(\overleftarrow{\mathbf{g}}_i) \odot \overleftarrow{\mathbf{x}}_i \\
\mathbf{h}_i &= (1-\sigma(\mathbf{z}_i)) \odot [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] + \sigma(\mathbf{z}_i) \odot \mathbf{x}_i
\end{aligned}$$

3.2 Decoder

The decoder employs unidirectional units, with layer normalization (*LN*) after every matrix multiplication similarly to the encoder units, and has an attention mechanism in every layer. The attention output is combined with the layer's hidden state in a way similar to the *deep output* (Pascanu et al., 2014) used by Luong (2015). The highway connection is applied only at the end of the unit. The presence of multiple attention models connected to the last encoder layer produces a high gradient for the encoder output, thus we scale the gradient dividing the attention output by $\sqrt{d}$. This kind of scaling has been proposed in (Vaswani et al., 2017) inside the transformer model, but we observed empirically that this version works better for our model. Formally:

$$\mathbf{y_i} \in R^d; \;\; \mathbf{W} \in R^{d\times 3d}; \;\; \mathbf{W_s}, \mathbf{W_c} \in R^{d\times d}$$

$$\begin{aligned}
[\tilde{\mathbf{y}}_\mathbf{i}, \mathbf{g_i}, \mathbf{z_i}] &= LN(\mathbf{y_i}\mathbf{W}) \\
\tilde{\mathbf{s}}_\mathbf{i} &= (1-\sigma(\mathbf{g_i})) \odot \tilde{\mathbf{s}}_\mathbf{i-1} + \sigma(\mathbf{g_i}) \odot \tilde{\mathbf{y}}_\mathbf{i} \\
\mathbf{c_i} &= attn(\tilde{\mathbf{s}}_\mathbf{i}, \mathbf{H})(1/\sqrt{d}) \\
\mathbf{o_i} &= \tanh(LN(\tilde{\mathbf{s}}_\mathbf{i}\mathbf{W_s}) + LN(\mathbf{c_i}\mathbf{W_c})) \\
\mathbf{s_i} &= (1-\sigma(\mathbf{z_i})) \odot \mathbf{o_i} + \sigma(\mathbf{z_i}) \odot \mathbf{y_i}
\end{aligned}$$

The decoder includes a standard *softmax* layer over the target vocabulary which is omitted from this description. For our architecture, we opted for a layer-normalized version of the MLP global attention (Bahdanau et al., 2015), which showed to perform better than the dot attention model (Luong et al., 2015):

$$\begin{aligned}
\tilde{\alpha}_\mathbf{ij} &= \mathbf{v}_\alpha \tanh(LN(\tilde{\mathbf{s}}_\mathbf{i}\mathbf{W_{as}}) + LN(\mathbf{h_j}\mathbf{W_{ah}})) \\
\boldsymbol{\alpha_i} &= \text{softmax}(\tilde{\boldsymbol{\alpha}}_i) \\
\mathbf{c_i} &= \sum_{i=0}^{L} \alpha_{ij}\mathbf{h_j}
\end{aligned}$$

Our SR-NMT architecture stacks several layers both on the encoder and decoder sides, as shown in Figure 2. The natural structure we consider is one having the same number of layers on both sides, although different topologies could be considered, too.

4 Experiments

We implemented our architecture in PyTorch (Paszke et al., 2017) inside the OpenNMT-py toolkit (Klein et al., 2017). All the tested models have been trained with the Adam (Kingma and Ba, 2015) optimizer until convergence, using the typical initial learning rate of 0.0003, and default values for β_1 and β_2. At convergence, the models were further trained until new convergence with learning rate 0.00015 (Bahar et al., 2017). The model used to restart the training is selected according to the perplexity on the validation set. We applied dropout of 0.1 before every multiplication by a parameter matrix, and in the case of LSTM it is applied only to vertical connections in order to use the LSTM version optimized in CUDA. The batch size is 64 for all the experiments and all the layers for all the models have an output size of 500.

4.1 Datasets

We used as benchmarks the WMT14 English to German and the WMT16 English to Romanian

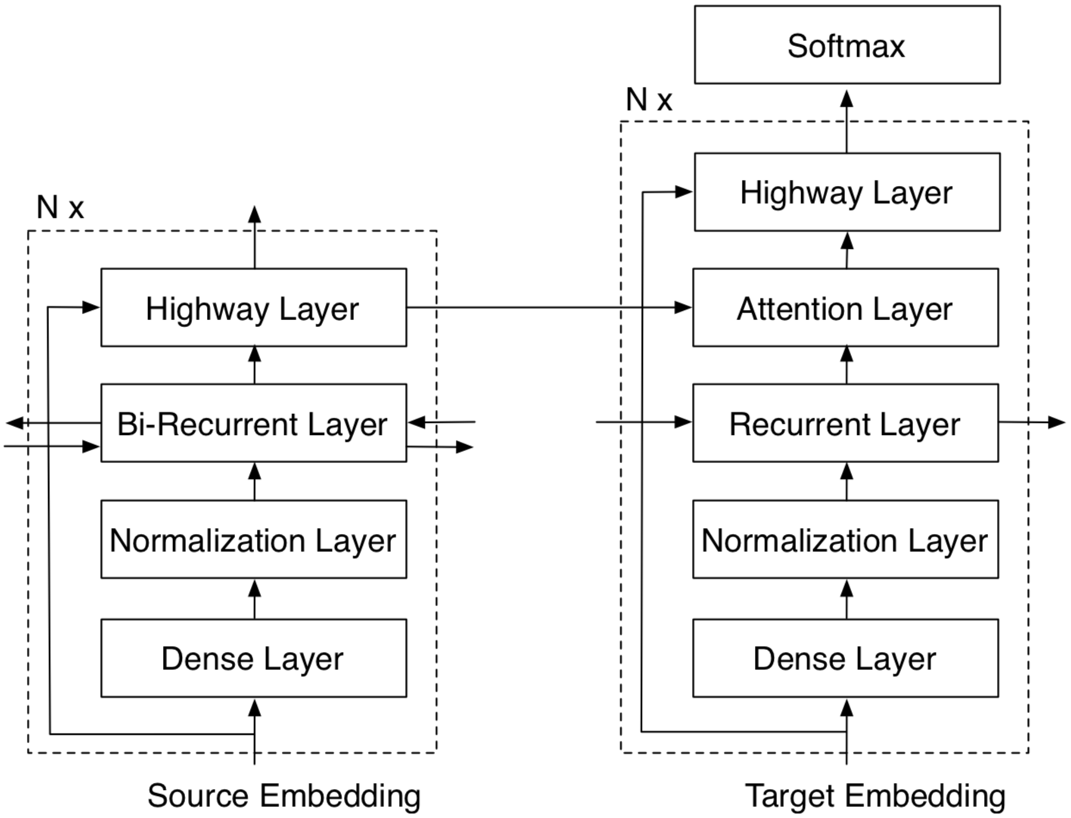


Figure 2: SR-NMT encoder-decoder architecture. On the left, a single encoder block to repeat N times. The output of the last layer is used as input for the decoder's attention layers. On the right, a decoder block to repeat N times. The first three sub-layers are the same in the encoder and the decoder, but the latter has an attention layer before the highway connection.

datasets.

In the case of WMT14 En-De, the training set consists of the concatenation of all the training data that were available for the 2014 shared task, the validation set is the concatenation of newstest2012 and 2013, and newstest2014 is our test set. Then, it was preprocessed with tokenization, punctuation normalization and de-escaping of the special characters. Furthermore, we applied BPE segmentation (Sennrich et al., 2016a) with 32,000 merge rules. We removed from the training data all the sentence pairs where the length of at least one sentence exceeded 50 tokens, resulting on a training set of $3.9M$ sentence pairs. Furthermore, we cleaned the training set by removing sentences in a wrong language and poorly aligned sentence pairs. For the cleaning process we used the automatic pipeline developed by the ModernMT project[2].

In the case of WMT16 En-Ro, we have used the same data and preprocessing used by Sennrich et al. (2016b) and Gehring et al. (2017). The back-translations to replicate the experiments are available[3] and we applied the same preprocessing[4], which involves punctuation normalization, tokenization, truecasing and BPE with $40K$ merge rules.

5 Evaluation

In this section, we describe the evaluation of our models with the two benchmarks. As our main goal is to prove that SR-NMT represent a valid alternative to LSTMs, we have put more effort on WMT14 En-De, which is widely used as a benchmark dataset. The experiments on WMT16 En-Ro are aimed to verify the effectiveness of our models in a different language pair with a different data size.

5.1 WMT14 English to German

The results for WMT14 En-De are evaluated on cased output, tokenized with the tokenizer script from the Moses toolkit (Koehn et al., 2007), and the BLEU score is computed using multi-bleu.pl

[2] https://github.com/ModernMT/MMT

[3] http://data.statmt.org/rsennrich/wmt16_backtranslations/en-ro .

[4] https://github.com/rsennrich/wmt16-scripts/blob/80e21e5/sample/preprocess.sh

from the same toolkit. With this procedure the results are comparable with the results reported from the other publications[5].

We compare our models with the results reported in (Lei et al., 2017b), and also reproduce some of their experiments. We train 4 baseline models following the same procedure used for SR-NMT. Three baselines are LSTM-based NMT models as provided by OpenNMT-py, with 2, 3 and 5 layers in both encoder and decoder. The other is an SRU model with 3 layers that we re-implemented in PyTorch, in order to perform a more fair comparison with our model. For the baselines we use dropout after every layer and MLP attention (Luong et al., 2015), both resulting in better results than the default implementation. Furthermore, we compare our results with Google's NMT system (Wu et al., 2016), Convolutional S2S model (Gehring et al., 2017), and the Transformer (Vaswani et al., 2017).

5.2 WMT16 English to Romanian

In the case of English to Romanian, we trained our models with the same hyper-parameters used for English to German, despite the difference in the amount of data. The BLEU score is computed using the official script of the shared task[6], which runs on cased and detokenized output.

We did not implement baselines for this language pair, and we compare our results with the winning submission of the WMT16 shared task (Sennrich et al., 2016b), with the Convolutional S2S model (Gehring et al., 2017) and the Transformer (Gu et al., 2018).

WMT14 En-De	**BLEU**	**# par**
LSTM 2L	21.82	62M
LSTM 3L	22.26	65M
LSTM 5L	22.72	72M
SRU 3L	20.88	59M
SR-NMT 1L	18.33	56M
SR-NMT 2L	21.82	58M
SR-NMT 3L	22.35	61M
SR-NMT 4L	23.32	63M
SR-NMT 5L	24.11	66M
SR-NMT 6L	23.93	68M
SR-NMT 7L	24.34	71M
SR-NMT 8L	24.87	73M
SR-NMT 9L	25.04	76M
SR-NMT 10L	24.98	78M
Setting of (Lei et al., 2017b)		
LSTM 2L	19.67	84M
LSTM 5L	20.45	96M
SRU 3L	18.89	81M
SRU 10L	20.70	91M
GNMT (Wu et al., 2016)		
LSMT 8L	24.61	-
Ensemble	26.30	-
Convolutional (Gehring et al., 2017)		
ConvS2S 15L	25.16	-
Ensemble	26.43	-
Transformer (Vaswani et al., 2017)		
Base 6L	27.30	65M
Big 6L	**28.40**	213M

Table 2: Experiments with cleaned data on WMT14 En-De both for our architectures and the baselines, and comparison with the state of the art.

6 Results

In this section, we discuss the performance in terms of training speed and translation quality of our architecture.

6.1 WMT14 En-De

In the first part of Table 2 we list the results of SR-NMT using from 1 up to 10 layers and our baselines. The training speeds are reported in Table 1.

SR-NMT with 3 layers has a number of parameters comparable to the LSTM baseline with 2 layers, but its training speed is 14% faster (4300 tok/s vs 3700 tok/s), and the BLEU score is 0.5 points higher. Moreover, the implementation of the LSTM is optimized at CUDA level, while our architecture is fully implemented in PyTorch and could be made faster following the optimizations of Lei et al. (2017b). Furthermore, also the layer normalization can be implemented faster in CUDA[7]. By increasing the number of LSTM layers from 2 to 5, the improvement in terms of BLEU score is only 0.9 points, and it is worse than SR-NMT with 4 layers.

The comparison with NMT based on SRUs is in favor of our architecture, which achieves higher translation quality with less layers. In particular, SR-NMT with 2 layers outperforms SRU NMT with 3 layers by 1 BLEU point and also trains

[5] https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utils/get_ende_bleu.sh

[6] mteval-v13a.pl

[7] https://github.com/MycChiu/fast-LayerNorm-TF

WMT16 En-Ro	BLEU
SR-NMT 1L	24.74
SR-NMT 2L	26.41
SR-NMT 4L	28.81
SR-NMT 6L	29.04
SR-NMT 8L	28.70
GRU (Sennrich et al., 2016b)	
GRU 1L+2L	28.1
Ensemble	28.2
Convolutional (Gehring et al., 2017)	
ConvS2S 15L	30.02
Transformer (Gu et al., 2018)	
NAT	29.79
Transformer	**31.91**

Table 3: Results on the test set of WMT16 En-Ro and comparison with the state of the art.

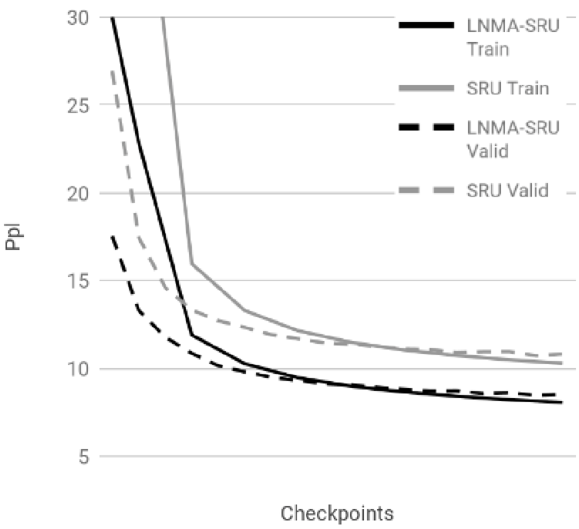


Figure 3: Perplexity against time for SR-NMT and SRU-based NMT with 3 layers and the same optimization policy. The convergence is achieved after a comparable number of iterations, but SR-NMT achieves a better convergence point.

faster (Table 1). However, this comparison is performed with implementations that are not optimized for fast execution in GPU. A speed comparison with optimized implementations could lead to different results.

In the second part of Table 2 we report some results from (Lei et al., 2017b) on the same benchmark. The different number of parameters is probably due to a different size of the vocabulary, in fact the number of merge rules used is not reported in the paper. Our LSTM baseline performs clearly better then the one cited because of the straightforward improvements we implemented, i.e. the use of *input feeding* (Luong et al., 2015), MLP attention instead of general or dot attention, and dropout in every layer. With this improvements, our baseline with 2 layers obtains 1.4 BLEU scores more than its counterpart using 5 layers. Moreover, it also outperforms SRU with 10 layers by more than 1 point. This result shows that our additions are fundamental to have a competitive architecture based on weakly recurrent units.

Figure 3 shows a comparison of the learning curves of SR-NMT and SRU NMT both with 3 layers. We can easily observe that the convergence of SR-NMT occurs at comparable speed but to a better point and the validation perplexities of the two models are very close to the training perplexities.

When we compare SR-NMT to GNMT (Table 2), we can observe that SR-NMT with 8 layers performs slightly better than GNMT, which in turn uses many more parameters, as it uses 8 LSTM layers with size 1024. Moreover, GNMT was trained for 6 days on 96 Nvidia K80 GPUs, while our model took the equivalent[8] of 12 days on a single K80 GPU.

Our best BLEU score, 25.04, is obtained with 9 layers. This is only 0.12 BLEU points below the convolutional model that used 15 layers in both encoder and decoder, and hidden sizes of at least 512. Finally, we notice that SR-NMT's best performance is still below that of the transformer model. Future work will be devoted to deeper explore the hyper-parameter space of our architecture and enhance it along the recent developments in (Chen et al., 2018).

6.2 WMT16 En-Ro

The results for WMT16 En-Ro are listed in Table 3. We obtain the highest score for this dataset with 6 layers, which can be due to the smaller dimension of the dataset, for which we did not add any form of regularization.

Our best SR-NMT system, which obtained a BLEU score of 29.04, is 1 BLEU point lower than ConvS2S, and almost 3 BLEU points lower than the state of the art. Nonetheless, this score is almost 1 BLEU point better than the score obtained by the winning system in WMT16 (Bojar et al., 2016), showing that SR-NMT represent a viable alternative to more complex RNNs.

[8] As our training currently works on single GPU, we could only fit models up to 7 layers into a K80, hence the estimate. Actually, models above 7 layers were trained on a V100 GPU.

Model	BLEU	Δ
LNMA-SRU 4L	22.99	0
- LayerNorm	21.97	-1.02
- Multi Attention	21.57	-1.42
- Highway	20.85	-2.14
- Ln & MA	20.51	-2.48
- LN & highway	/	- /
- MA & highway	19.54	-3.45
- LN, MA & highway	18.39	-4.6

Table 4: Ablation experiments on SR-NMT with 4 layers. BLEU scores are computed after one training stage. While removing multi attention we still keep one attention model in the last layer. The system without layer normalization and highway connections failed to converge.

7 Ablation experiments

In this section, we evaluate the importance of our enhancements to the original SRU unit, namely multi-attention and layer normalization, and of the highway connections, which were already present in the original formulation of SRUs.

We take our SR-NMT model with 4 layers and remove from it one component or a set of components. All the combinations are reported. Results refer to the WMT14 En-De task after performing only one training stage. In other words, we did not restart training after convergence as we did for the systems reported in Table 2. As our previous experiments already proved the superiority of SR-NMT to LSTMs, the goal of this section is to understand whether all the proposed additions are important and to quantify their contributions.

From Table 4 we can observe that the removal of highway connections causes the highest drop in performance (-2.14 BLEU points), followed by multi attention and then layer normalization. Another important observation is the additivity of the contributions from all the components, in fact when two or three components are removed at once, the drop in performance is roughly the sum of the drops caused by the single components. Finally, the removal of layer normalization and highway connections, while keeping multi attention, causes a gradient explosion that prevents the SR-NMT system from converging.

8 Conclusions

In this paper we have presented a simple recurrent NMT architecture that enhances previous SRUs (Lei et al., 2017b) by adding elements of other architectures, namely layer normalization and multiple attentions. Our goal was to explore the possibility to make weakly-recurrent units competitive with LSTMs for NMT. We have shown that our SR-NMT architecture is able to outperform more complex LSTM NMT models on two public benchmarks. In particular, SR-NMT performed even better than the GNMT system, while using a simpler optimization policy, a vanilla beam search and a fraction of its computational resources for training. Future work will be in the direction of further enhancing SR-NMT by integrating core components that seem to particularly boost performance of the best non recurrent NMT architectures.

Acknowledgements

We would like to thank Nicola Bertoldi for his technical support and the anonymous reviewers for their useful comments. We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research.

References

Ba, Jimmy Lei, Jamie Ryan Kiros Geoffrey E. Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Bahar, Parnia, Tamer Alkhouli, Jan-Thorsten Peter, Christopher J. S. Brix, and Hermann Ney. 2017. Empirical investigation of optimization algorithms in neural machine translation. *The Prague Bulletin of Mathematical Linguistics, 108(1)*, 13-25.

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate *Proceedings of the 3rd International Conference on Learning Representations,* San Diego, USA.

Balduzzi, David and Muhammad Ghifary. 2016. Strongly-Typed Recurrent Neural Networks. *International Conference on Machine Learning*, 1292–1300.

Barone, Antonio Valerio Miceli, Jindrich Helcl, Rico Sennrich, Barry Haddow and Alexandra Birch. 2017. Deep architectures for Neural Machine Translation. *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. 99-107.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, November 1-5, 2016, Austin, Texas, USA. 257–267

Bradbury, James, Stephen Merity, Caiming Xiong and Richard Socher. 2017. Quasi-Recurrent Neural Networks. *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation (wmt16). *Proceedings of the First Conference on Machine Translation*, Berlin, Germany. 131–198.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. 169–214.

Cettolo, Mauro, Christian Girardi and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, Trento, Italy.

Chen, Mia Xu, et al. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. *arXiv preprint arXiv:1804.09849.*

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches *Syntax, Semantics and Structure in Statistical Translation (2014): 103.*

Gal, Yarin and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *In Advances in neural information processing systems*, 1019–1027.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *Proceedings of the 34th International Conference on Machine Learning*, Sidney, Australia. 1243–1252.

Gu, Jiatao, James Bradbury, Caiming Xiong, Victor O.K. Li, Richard Socher. 2018. Non-Autoregressive Neural Machine Translation. *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory *Neural computation.* MIT Press 1735–1780.

Ioffe, Sergey and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Kalchbrenner, Nal, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves and Koray Kavukcuoglu. 2016 Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099.*

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, San Diego, USA.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart and Alexander M. Rush. 2017. *OpenNMT: Open-Source Toolkit for Neural Machine Translation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, System Demonstrations, 67-72.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej Bojar, Alexandra Constantin, Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of ACL on interactive poster and demonstration sessions*, 177–180.

Laurent, César, Gabriel Pereyra, Philémon Brakel, Ying Zhang and Yoshua Bengio. 2016. Batch normalized recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2657–2661.

Lei, Tao and Wengong Jin, Regina Barzilay and Tommi Jaakkola. 2017. Deriving Neural Architectures from Sequence and Graph Kernels. *Proceedings of the 34th International Conference on Machine Learning*, Sidney, Australia. 2024–2033.

Lei, Tao, Yu Zhang and Yoav Artzi. 2017. Training RNNs as Fast as CNNs. *arXiv preprint arXiv:1709.02755.*

Luong, Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* 1412–1421.

Luong, Minh-Thang and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. *Proceedings of the 12th International Workshop on Spoken Language Translation*, Da Nang, Vietnam. 76–79.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).* 311–318.

Pascanu, Razvan, Tomas Mikolov and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, USA. 1310–1318.

Pascanu, Razvan, Caglar Gulcehre, Kyunghyun Cho and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga and Adam Lerer. 2017. Automatic differentiation in PyTorch. *NIPS 2017 Autodiff Workshop*, Long Beach, USA.

Press, Ofir and Lior Wolf. 2017. Using the Output Embedding to Improve Language Models. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 2, Short Papers (Vol. 2, pp. 157-163).

Ravanelli, Mirco, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. 2018. Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence 2*, no. 2 (2018): 92-102.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany. 1715–1725.

Sennrich, Rico, Barry Haddow and Alexandra Birch. 2016. Edinburgh Neural Machine Translation Systems for WMT 16. *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, Vol. 2, 371–376.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Lubli, Antonio Valerio Miceli Barone, Jozef Mokry, Maria Ndejde. 2017. Nematus: a Toolkit for Neural Machine Translation. *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain. 65–68.

Sennrich, Rico, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. 389–399.

Siegelmann, Hava T. and Eduardo D. Sontag. 1995. On the computational power of neural nets. *Journal of computer and system sciences*, 50(1), 132-150.

Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. ICML 2015 Deep Learning Workshop.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. 2014 Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, Vol. 15, n. 1, 1929–1958.

Sutskever, Ilya, Oriol Vinyals and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems.* 3104–3112.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jacob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems (pp. 6000-6010).

Wu, Yonghui,Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes and Jeffrey Dean. 2016. Googles Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint arXiv:1609.08144*

Zaremba, Wojciech, Ilya Sutskever and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*

Zhou, Guo-Bing, Jianxin Wu, Chen-Lin Zhang, Zhi-Hua Zhou. 2016. Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing 13.3(2016)*: 226-234.

Spelling Normalization of Historical Documents by Using a Machine Translation Approach

Miguel Domingo
PRHLT Research Center
Universitat Politècnica de València
midobal@prhlt.upv.es

Francisco Casacuberta
PRHLT Research Center
Universitat Politècnica de València
fcn@prhlt.upv.es

Abstract

The lack of a spelling convention in historical documents makes their orthography to change depending on the author and the time period in which each document was written. This represents a problem for the preservation of the cultural heritage, which strives to create a digital text version of a historical document. With the aim of solving this problem, we propose three approaches—based on statistical, neural and character-based machine translation—to adapt the document's spelling to modern standards. We tested these approaches in different scenarios, obtaining very encouraging results.

1 Introduction

With the aim of preserving the cultural heritage, there is an increased need for the digitalization of historical documents, a procedure which strives for creating digital text which can be searched and automatically processed (Piotrowski, 2012). However, the linguistic properties of historical documents create an additional difficulty. On the one hand, human language evolves with the passage of time. On the other hand, the lack of a spelling convention makes orthography to change depending on the author and the time period in which a given document was written. This makes historical documents harder to read, and makes it even more difficult to search for certain information in a collection of documents, or any other process that must be applied to them.

Spelling normalization aims to resolve these problems. Its goal is to adapt the document's spelling to modern standards, increasing documents' readability and achieving an orthography consistency. Some approaches to spelling normalization include creating an interactive tool that includes spell checking techniques to assist the user in detecting spelling variations (Baron and Rayson, 2008). Porta et al. (2013) made use of a weighted finite-state transducer, combined with a modern lexicon, a phonological transcriber and a set of rules. Scherrer and Erjavec (2013) combined a list of historical words, a list of modern words and character-based Statistical Machine Translation (SMT). Bollmann and Søgaard (2016) took a multi-task learning approach using a deep bi-LSTM applied at a character level. Ljubešic et al. (2016) applied a token/segment-level character-based SMT approach to normalize historical and user-created words. Domingo et al. (2017) applied a SMT approach combined with the use of data selection techniques. Finally, Korchagina (2017) made use of rule-based MT, character-based SMT and character-based NMT.

In this work, we propose three approaches to tackle spelling normalization: a method based on SMT; another method based on Neural Machine Translation (NMT); and another method based on Character-Based Machine Translation (CBMT). Our main contribution are the followings:

- First use (to the best of our knowledge) of word-based and subword-based NMT—character-based NMT was already used by Korchagina (2017)—for spelling normalization.
- Comparison of different approaches based on SMT and NMT.
- Experimented with four historical corpora

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 129–137
Alacant, Spain, May 2018.

from three different time periods, in two different languages and with three distinct alphabets.

The rest of this document is structured as follows: In Section 2, we introduce the machine translation approaches used in our work. Section 3 presents the different approaches taken to achieve spelling normalization. Then, in Section 4, we describe the experiments conducted in order to assess our proposal. After that, in Section 5, we present and discuss the results of those experiments. Finally, in Section 6, conclusion are drawn.

2 Machine Translation Approaches

In this section, we present the machine translation approaches used in our work.

2.1 Statistical Machine Translation

The goal of SMT is to find, given a source sentence $\mathbf{x}$, its best translation $\hat{\mathbf{y}}$ (Brown et al., 1993):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} Pr(\mathbf{y} \mid \mathbf{x}) \tag{1}$$

For years, phrase-based models (Koehn, 2010) have been the prevailing approach to compute this expression. These models rely on a log-linear combination of different models (Och and Ney, 2002): namely, phrase-based alignment models, reordering models and language models; among others (Zens et al., 2002; Koehn et al., 2003). However, more recently, this approach has shifted into neural models (see Section 2.2).

2.2 Neural Machine Translation

NMT is the neural approach to compute Eq. (1). Frequently, it relies on a Recurrent Neural Network (RNN) encoder-decoder framework. At the encoding step, the source sentence is projected into a distributed representation. Then, at the decoding step, the decoder generates its translation word by word (Sutskever et al., 2014).

The system's input is a sequence of words in the source language. Each source word is linearly projected to a fixed-sized real-valued vector through an embedding matrix. These word embeddings are feed into a bidirectional (Schuster and Paliwal, 1997) Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) network, resulting in a sequence of annotations produced by concatenating the hidden states from the forward and backward layers.

The model features an attention mechanism (Bahdanau et al., 2015), which allows the decoder to focus on parts of the input sequence, computing a weighted mean of annotations sequence. These weights are computed by a soft alignment model, which weights each annotation with the previous decoding state.

The decoder is another LSTM network, conditioned by the representation computed by the attention model and the last word generated. Finally, a deep output layer (Pascanu et al., 2013) computes a distribution over the target language vocabulary.

The model is trained by means of stochastic gradient descend, applied jointly to maximize the log-likelihood over a bilingual parallel corpus. At decoding time, the model approximates the most likely target sentence with beam-search (Sutskever et al., 2014).

2.3 Character-based Machine Translation

CBMT comes as a solution to reduce the training vocabulary by dividing words into a sequence of characters, and treating each character as if it were a word. Moreover, it also strikes for being a solution of not having a perfect segmentation algorithm—which should be able to segment a given sentence in any language, into a sequence of lexemes and morphemes (Chung et al., 2016).

Although CBMT was already being researched in SMT (Tiedemann, 2009; Nakov and Tiedemann, 2012), its interest has increased with NMT. Some approaches to character-based NMT consist in using hierarchical NMT (Ling et al., 2015), a character level decoder (Chung et al., 2016), a character level encoder (Costa-Jussà and Fonollosa, 2016) or, for alphabets in which words are composed by fewer characters, by constructing an NMT system that takes advantage of that alphabet (Costa-Jussà et al., 2017).

3 Spelling Normalization

In this section, we propose different approaches to adapt the spelling of historical documents to modern standards.

Our first approach is based on SMT. Considering the document's language as the source language and its normalized version of that language as the target language, we propose to use SMT to adapt the document's spelling to modern standards.

In our second approach, we wanted to assess how well NMT works for normalizing the spelling of a historical document. Therefore, similarly as to with the previous approach, considering the document's language as the source language and its normalized version of that language as the target language, we propose to use NMT to adapt the document's spelling to modern standards.

Finally, since in spelling normalization changes frequently occur at a character level, it seemed fitting to use a character-based strategy. Therefore, our third approach is based on CBMT. Similarly as to with the previous approaches, considering the document's language as the source language and its normalized version of that language as the target language, we propose to use CBMT to adapt the document's spelling to modern standards.

As a starting point and to have the same conditions in both SMT and NMT, in this work we chose to use the simplest character-based approach: to split words into characters and, then, apply conventional SMT/NMT.

4 Experiments

In this section, we describe the experiments conducted in order to assess our proposal. Additionally, we present the corpora and metrics.

4.1 Corpora

To conduct our experiments, we made use of the following corpora:

Entremeses y Comedias (F. Jehle, 2001): A collection of comedies by Miguel de Cervantes, written in 17th century Spanish.

Quijote (F. Jehle, 2001): The 17th century Spanish novel by Miguel de Cervantes.

Bohorič (Ljubešić et al., 2016): A collection of 18th century Slovene texts written in the Bohorič alphabet.

Gaj (Ljubešić et al., 2016): A collection of 19th century Slovene texts written in the Gaj alphabet.

The first two corpora are Spanish literary works, written across the 17th century. The first corpus is composed of 16 plays—8 of which have a very short length—while the second corpus is a two-volumes novel. The last two corpora are a collection of texts extracted from Slovene books. The first one is made up of texts from the 18th century and it is written in the old Bohorič alphabet, and the second one is made up of texts from the 19th century and written in the contemporary Gaj alphabet. Table 1 shows the corpora statistics.

4.2 Metrics

In order to asses our proposal, we made use of the following well-known metrics:

BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002): computes the geometric average of the modified n-gram precision, multiplied by a brevity factor that penalizes short sentences.

Translation Error Rate (TER) (Snover et al., 2006): computes the number of word edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation.

Character Error Rate (CER): computes the number of character edit operations (insertion, substitution and deletion), normalized by the number of characters in the final translation.

Confidence intervals ($p = 0.05$) are computed for all metrics by means of bootstrap resampling (Koehn, 2004).

4.3 Systems

SMT systems were trained with the `Moses` toolkit (Koehn et al., 2007), following the standard procedure: we optimized the weights of the log-lineal model with MERT (Och, 2003), and used `SRILM` (Stolcke, 2002) to estimate a 5-gram language model, smoothed with the improved Kneser-Ney method (Chen and Goodman, 1996). Moreover, since source and target have the same linguistic structures—the only changes between source and target are orthographic—we used monotonous reordering. Finally, the corpora were lowercased and tokenized using the standard scripts, and the translated text was truecased with `Moses`' truecaser.

NMT systems were trained with `OpenNMT` (Klein et al., 2017), as described in Section 2.2. Following the findings from Britz et al. (2017), we used LSTM units. The size of the LSTM and word embedding were set according to the results of the development set. We used *Adam* (Kingma and Ba, 2014) with a learning rate of 0.0002 (Wu et al., 2016). The beam size was set to 6. Finally, the corpora were

		Entremeses y Comedias	**Quijote**	**Bohorič**	**Gaj**
Train	$\|S\|$	35.6K	48.0K	3.6K	13.0K
	$\|T\|$	250.0/244.0K	436.0/428.0K	61.2/61.0K	198.2/197.6K
	$\|V\|$	19.0/18.0K	24.4/23.3K	14.3/10.9K	34.5/30.7K
	$\|W\|$	52.4K	97.5K	33.0K	32.7K
Development	$\|S\|$	2.0K	2.0K	447	1.6K
	$\|T\|$	13.7/13.6K	19.0/18.0K	7.1/7.1K	25.7/25.6K
	$\|V\|$	3.0/3.0K	3.2/3.2K	2.9/2.5K	8.2/7.7K
	$\|W\|$	1.9K	4.5K	3.8K	4.5K
Test	$\|S\|$	2.0K	2.0K	448	1.6K
	$\|T\|$	15.0/13.3K	18.0/18.0K	7.3/7.3K	26.3/26.2K
	$\|V\|$	2.7/2.6K	3.2/3.2K	3.0/2.6K	8.4/8.0K
	$\|W\|$	3.3K	3.8K	3.8K	4.8K

Table 1: Corpora statistics. $|S|$ stands for number of sentences, $|T|$ for number of tokens, $|V|$ for size of the vocabulary and $|W|$ for the number of words whose spelling does not match modern standards. K denotes thousand.

lowercased and tokenized—and, later, truecased and detokenized—using `OpenNMT`'s tools.

CBMT systems were trained in the same way as conventional SMT/NMT systems. The only difference is that the corpora's words were previously split into characters. Then, after translating the document, words were restored.

To reduce the vocabulary, we used Byte Pair Encoding (BPE) (Sennrich et al., 2016). These systems were trained in the same way as conventional SMT/NMT systems. The only difference is that the corpora were previously encoded using BPE, and the translated text was decoded afterwards. BPE encoding was learned and applied using the scripts kindly provided by Sennrich et al. (2016). In learning the encoding, we used the default values for the number of symbols to create and the minimum frequency to create a new symbol.

Finally, in order to assess our proposal, we considered as a baseline the quality of the original document with respect to its ground truth version, in which the spelling has already been updated to match modern standards. Nonetheless, as a second baseline, we implemented a statistical dictionary. Using `mgiza` (Gao and Vogel, 2008), we computed *IBM's model 1* (Och and Ney, 2003) to obtain word alignments from source and target of the training set. Then, for each source word, we selected as its translation the target word which had the highest alignment probability with that source word. Finally, at translation time, we translated each source word with the translation that appeared in the dictionary. If a given word did not appear in the dictionary, then we left it untranslated.

5 Results

In this section, we present and discuss the experiments conducted in order to assess our proposal. Table 2 presents the experimental results.

The Slovene language had a big restructuring in the 18$^{\text{th}}$ century. For this reason, *Bohorič*—whose documents were written during this period—is the corpus whose orthography differs the most compared to modern standards. Evaluating the document's spelling differences with respect to modern orthography results in a low BLEU value, a high TER value and a fairly high CER value. However, just by applying a statistical dictionary we achieved great improvements: BLEU and TER improved highly, and CER decreased significantly.

With our first approach, we achieved even greater improvements for all metrics. Furthermore, when using BPE to reduce the vocabulary, we achieved new improvements. These improvements were more notorious when evaluating with CER and BLEU, although they were significant with TER as well.

Our second approach achieved less satisfying results. The document's spelling differences were significantly reduced when measuring with BLEU and TER. However, the results were significantly worse than the ones obtained using a statistical dictionary. Furthermore, using CER to measure the spelling differences resulted in the document having more differences than the original document. Using BPE to reduce the vocabulary did not help. In fact, results were significantly worse. Most likely, this was due to the properties of the corpus: being the smallest of the corpora (less than four thousand sentences and with a vocabulary of around ten thou-

System	Entremeses y Comedias			Quijote			Bohorič			Gaj		
	BLEU	TER	CER	BLEU	TER	CER	BLEU	TER	CER	BLEU	TER	CER
Baseline	46.1 ± 1.4	31.7 ± 1.2	12.0 ± 0.4	59.6 ± 1.2	19.4 ± 0.7	7.4 ± 0.3	16.4 ± 1.6	49.0 ± 1.5	21.7 ± 0.6	68.1 ± 1.1	12.3 ± 0.5	3.5 ± 0.1
SD	80.8 ± 1.2	8.3 ± 0.5	4.0 ± 0.3	89.7 ± 0.8	5.3 ± 0.5	3.4 ± 0.3	52.5 ± 2.0	20.7 ± 1.2	17.2 ± 0.7	75.1 ± 0.8	8.8 ± 0.4	8.7 ± 0.3
SMT	82.1 ± 1.1	8.0 ± 0.5	6.7 ± 0.2	91.1 ± 0.7	4.5 ± 0.4	5.3 ± 0.3	63.0 ± 2.1	15.1 ± 1.1	9.0 ± 0.5	**82.6 ± 0.7**	5.2 ± 0.3	2.8 ± 0.1
SMT_{BPE}	83.6 ± 1.1	7.2 ± 0.5	6.2 ± 0.2	**94.6 ± 0.6**	**2.8 ± 0.3**	4.3 ± 0.2	70.4 ± 2.0	11.7 ± 1.0	5.3 ± 0.3	**83.7 ± 0.7**	**1.8 ± 0.3**	2.7 ± 0.1
NMT	72.2 ± 1.4	15.2 ± 0.9	18.0 ± 0.8	84.4 ± 0.9	8.1 ± 0.5	10.2 ± 2.4	36.7 ± 2.0	33.9 ± 2.1	41.4 ± 1.4	50.4 ± 1.4	28.3 ± 3.3	36.0 ± 2.7
NMT_{BPE}	76.7 ± 1.3	12.4 ± 0.8	8.1 ± 0.5	92.0 ± 0.7	4.6 ± 0.4	3.8 ± 0.3	31.6 ± 2.2	43.5 ± 6.1	48.6 ± 3.6	68.0 ± 1.5	23.7 ± 3.7	19.8 ± 2.6
CBSMT	**91.4 ± 0.9**	**3.7 ± 0.4**	**1.2 ± 0.1**	**94.7 ± 0.6**	**2.8 ± 0.3**	**2.0 ± 0.2**	**75.5 ± 1.8**	**8.7 ± 0.9**	**2.4 ± 0.2**	**83.2 ± 0.7**	5.0 ± 0.3	**1.3 ± 0.1**
CBNMT	81.3 ± 1.3	8.3 ± 0.8	3.0 ± 0.6	91.0 ± 0.7	4.6 ± 0.4	2.9 ± 0.3	27.6 ± 2.4	85.2 ± 6.7	68.2 ± 4.5	40.2 ± 1.9	62.7 ± 2.9	52.5 ± 2.1

Table 2: Experimental results. Baseline system corresponds to considering the original document as the document to which the spelling has been normalized to match modern standards. SD is the statistical dictionary. SMT is the standard SMT system. SMT_{BPE} is the SMT system trained after encoding the corpora using BPE. NMT is the standard NMT system. NMT_{BPE} is the NMT system trained after encoding the corpora using BPE. CBSMT is the character-based SMT system. CBNMT is the character-based NMT system. Best results are denoted in **bold**.

sand words), it was not big enough for NMT to learn properly how to update the document's orthography.

Finally, our third approach was both the most and least satisfying. While character-based SMT achieved the best results for all metrics–all of them were significantly better than the results achieved by SMT with BPE—character-based NMT achieved the worst results. Once more, this was most likely due to the corpus being too small for the neural systems.

Entremeses y Comedias, the oldest of the corpora, is the next corpus with greater orthographic difference. Nonetheless, the quality of the original document shows fairly good BLEU value, a considerable good TER value, and a low CER value. In spite of this, the statistical dictionary achieved significant improvements, the most noteworthy being the increase of BLEU, which was the metric that showed the lowest quality.

The SMT approach reduced significantly the spelling differences from the original document. However, in this case, results were not significantly different to the ones obtained by the statistical dictionary, except when evaluating with CER, which results were slightly (around two CER points) worse. Moreover, reducing the vocabulary with BPE did not achieved a significant difference with using the full vocabulary.

The NMT system behave in a similar fashion as with the previous corpus: BLEU and TER showed a significant reduction of the spelling difference from the original document, but smaller than the reduction achieved by the statistical dictionary. In this case, however, the differences with the statistical dictionary were smaller (around eight points of BLEU and TER). Moreover, although CER still showed more spelling differences than in the original document, its value was not as bad as with *Bohorič* (around six points of CER). Furthermore, despite still being worse than the statistical dictionary, BPE helped to improve results. It is worth noting the improvement in CER (around ten points), which represents an improvement with respect to the spelling differences in the original document.

Once more, the character-based approach yielded the best results. Character-based NMT was the neural approach which yielded the best results, although these results were not significantly different to the ones obtained by the statistical dictionary. However, character-based SMT did significantly improved the statistical dictionary.

Similarly to what happened with the other corpora, the statistical dictionary significantly reduced the spelling differences in the *Quijote* corpus. It is worth noting, however, that these differences are considerable smaller in this corpus: measuring the spelling differences in the original document shows a fairly good BLEU and TER values, and fairly small CER values.

In this case, the SMT approach did not yield results as satisfactorily as with the previous corpora. Results showed a significant improvement with respect to the original document. However, this improvement was not significantly different than the one achieved by the statistical dictionary—except when measuring with CER, whose value was significantly worse. Nonetheless, BPE improved the results, and the generated document was significantly better (for all metrics except for CER) than the document generated by the statistical dictionary.

The results yielded by the NMT system showed a significant improvement with respect to the spelling differences from the original document (except when measuring with CER), but this improvement was significantly worse than the one achieved by the statistical dictionary. Reducing the vocabulary with BPE helped to improve the results—specially when measuring with CER, whose results were now significantly better than the original document—but they were similar to the statistical dictionary's results.

Finally, the character-based approached achieved, once more, the best results. However, while using CER to measure the document's spelling differences with respect to modern standards yielded a significant improvement (for character-based SMT), measuring with BLEU and TER yielded similar results to using the SMT approach combined with BPE. Similarly, character-based NMT achieved a significant improvement in terms of CER, but similar BLEU and TER results to the NMT-BPE approached.

Being the newest corpus, *Gaj* contains fewer spelling differences with respect to modern orthography. In fact, measuring the spelling differences from the original document already yielded satisfactory BLEU and TER values, and a low CER value. Nonetheless, the statistical dictionary managed to improve BLEU and TER results, although yielded a worse CER value.

The SMT approach managed to significantly improve results for all metrics. However, reducing the vocabulary with BPE yielded similar results, except when measuring with TER, whose results were significantly better.

Gaj being a fairly small corpus (thirteen thousand sentences and with a vocabulary of around thirty thousand words), the NMT systems behaved similarly as with *Bohorič*: The generated document had more spelling differences than the original document. Using BPE improved results, but the generated document still contained more spelling differences than the original one.

Character-based SMT yielded the best results when using CER to measure the spelling difference. However, measuring with BLEU and TER yielded similar results to the SMT approach. Character-based NMT, however, was the NMT approach which yielded the worst results—specially when measuring with TER and CER.

In general, except for one exception (*Gaj*, whose best results—when evaluating with TER—were achieved by the approach that combined SMT with BPE), character-based SMT was the approach that yielded the best results for all metrics. It is also worth noting how well—for being such a simplistic approach—using an statistical dictionary behave: except for one exception (*Gaj*, which yielded an increase of spelling differences when evaluating with CER), all results showed a significant reduction of spelling differences with respect to the original document and, in some cases, not too much worse than character-based SMT.

The BLEU and TER from the original document, and how much these values have significantly improved, seem to indicate that the final document is quite different to the original one. However, CER seems to indicate otherwise. Most likely, since spelling differences occur more frequently at a character level (i.e., most orthographic changes consist in a few letters per word), BLEU and TER—which evaluate at a word-level—are being penalized. Nonetheless, all metrics show that the spelling differences have been significantly reduced.

6 Conclusions and Future Work

In this work, we proposed three machine translation approaches to update the spelling of a historical document to match modern standards, increasing the document's readability and helping in the preservation of the cultural heritage.

Additionally, as an extra baseline, we proposed a simplistic approach: Based on the frequency of which, on the training corpora, the spelling of a word is changed, to build a statistical dictionary. Then, on a given document, we checked, word by word, if it was on the dictionary. If the search was positive, we changed that word by the translation that appeared in the dictionary. Otherwise, we left the word as it appeared in the original document.

We tested our proposal with four datasets formed by documents from three different time periods, two different languages and three distinct alphabets, obtaining very encouraging results.

In general, approaches based on SMT yielded better results than those based on NMT. This was specially true for the smallest corpora, in which the neural systems were not able to learn properly and yielded more spelling differences than the ones contained in the original document.

As it was to be expected due to the task characteristics (in spelling normalization, changes frequently occur at a character level), the character-based approaches—both phrased-based and neural—yielded the best results for each kind of system (i.e., character-based was the best SMT approach, and character-based NMT was the best NMT approach). The exception was character-based NMT, which yielded worse results when applied to the smallest corpora.

Finally, it is worth noting how well the statistical dictionary behaves. Although its results were not the best, they were close enough to take this approach into consideration. Being the simplest and fastest to compute, it could be useful in cases in which its worth sacrificing quality to increase speed.

As a future work, we would like to try new character-based approaches. In this work, we tested the simplest approach (to split the words into characters and, then, apply conventional SMT/NMT). However, more complex approaches have been developed in recent years (Chung et al., 2016; Costa-Jussà and Fonollosa, 2016; Costa-Jussà et al., 2017).

Finally, a frequent problem when working with historical documents is the scarce availability of parallel training data (Bollmann and Søgaard, 2016). Therefore, we would like to obtain more diverse corpora to be able to experiment in broader domains: older time periods, documents written by a great variety of authors, etc. Additionally, we would like to explore the generation of synthetic data (Sennrich et al., 2015) to create new training data.

Acknowledgments

The research leading to these results has received funding from the Ministerio de Economía y Competitividad (MINECO) under project CoMUN-HaT (grant agreement TIN2015-70924-C2-1-R), and Generalitat Valenciana (grant agreement PROMETEO/2018/004). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for this research.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (arXiv:1409.0473)*.

Baron, A. and Rayson, P. (2008). VARD2: A tool for dealing with spelling variation in historical corpora. *Postgraduate conference in corpus linguistics*.

Bollmann, M. and Søgaard, A. (2016). Improving historical spelling normalization with bi-directional lstms and multi-task learning. In *Proceedings of the International Conference on the Computational Linguistics*, pages 131–139.

Britz, D., Goldie, A., Luong, T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 310–318.

Chung, J., Cho, K., and Bengio, Y. (2016). A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1693–1703.

Costa-Jussà, M. R., Aldón, D., and Fonollosa, J. A. (2017). Chinese–spanish neural machine translation enhanced with character and word bitmap fonts. *Machine Translation*, 31:35–47.

Costa-Jussà, M. R. and Fonollosa, J. A. (2016). Character-based neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 357–361.

Domingo, M., Chinea-Rios, M., and Casacuberta, F. (2017). Historical documents modernization. *The Prague Bulletin of Mathematical Linguistics*, 108:295–306.

F. Jehle, F. (2001). *Works of Miguel de Cervantes in Old- and Modern-spelling*. Indiana University Purdue University Fort Wayne.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Proceedings of the Association for Computational Linguistics Software Engineering, Testing, and Quality Assurance Workshop*, pages 49–57.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1701.02810*.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Korchagina, N. (2017). Normalizing medieval german texts: from rules to deep learning. In *Proceedings of the Nordic Conference on Computational Linguistics Workshop on Processing Historical Language*, pages 12–17.

Ling, W., Trancoso, I., Dyer, C., and Black, A. W. (2015). Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Ljubešić, N., Zupan, K., Fišer, D., and Erjavec, T. (2016). Dataset of normalised slovene text KonvNormSl 1.0. Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1068.

Ljubešic, N., Zupan, K., Fišer, D., and Erjavec, T. (2016). Normalising slovene data: historical texts vs. user-generated content. In *Proceedings of the Conference on Natural Language Processing*, pages 146–155.

Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 301–305.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 295–302.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistic*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Piotrowski, M. (2012). *Natural Language Processing for Historical Texts*. Number 17 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Porta, J., Sancho, J.-L., and Gómez, J. (2013). Edit transducers for spelling variation in old spanish. In *Proceedings of the workshop on computational historical linguistics*, pages 70–79.

Scherrer, Y. and Erjavec, T. (2013). Modernizing historical slovene words with character-based smt. In *Proceedings of the Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 58–62.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 257–286.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks.

Tiedemann, J. (2009). Character-based PSMT for closely related languages. In *Proceedings of the Annual Conference of the European Association for Machine Translation*, pages 12–19.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zens, R., Och, F. J., and Ney, H. (2002). Phrase-based statistical machine translation. In *Proceedings of the Annual German Conference on Advances in Artificial Intelligence*, volume 2479, pages 18–32.

Neural Machine Translation of Basque

Thierry Etchegoyhen,[1] **Eva Martínez Garcia,**[1] **Andoni Azpeitia,**[1]
Gorka Labaka,[2] **Iñaki Alegria,**[2] **Itziar Cortes Etxabe,**[3] **Amaia Jauregi Carrera,**[3]
Igor Ellakuria Santos,[4] **Maite Martin,**[5] **Eusebi Calonge**[5]
[1]Vicomtech - {tetchegoyhen, emartinez, aazpeitia}@vicomtech.org
[2]IXA taldea, University of the Basque Country - {i.alegria, gorka.labaka}@ehu.eus
[3]Elhuyar - {i.cortes, a.jauregi}@elhuyar.eus
[4]ISEA - isantos@iseamcc.net
[5]Ametzagaiña - maite@adur.com, ecalonge@ametza.com

Abstract

We describe the first experimental results in neural machine translation for Basque. As a synthetic language featuring agglutinative morphology, an extended case system, complex verbal morphology and relatively free word order, Basque presents a large number of challenging characteristics for machine translation in general, and for data-driven approaches such as attention-based encoder-decoder models in particular. We present our results on a large range of experiments in Basque-Spanish translation, comparing several neural machine translation system variants with both rule-based and statistical machine translation systems. We demonstrate that significant gains can be obtained with a neural network approach for this challenging language pair, and describe optimal configurations in terms of word segmentation and decoding parameters, measured against test sets that feature multiple references to account for word order variability.

1 Introduction

Neural machine translation (NMT) is fast becoming the dominant paradigm in both academic research and commercial exploitation, as evidenced in particular by large machine translation providers turning to NMT for their production engines (Crego et al., 2016; Wu et al., 2016) and NMT systems achieving the best results in most cases on standard shared tasks datasets (Bojar, 2016).

Sequence-to-sequence neural networks have proved effective in modelling translation phenomena (Sutskever et al., 2014). In particular, attentional encoder-decoder models (Bahdanau et al., 2015) have become a default NMT architecture, with other architectural variants explored in recent work (Vaswani et al., 2017; Gehring et al., 2016). These models have already been applied to a wide range of languages, initially on the most studied European languages and recently to a larger set of cases that includes morphologically rich languages (Bojar, 2017).

In this article we explore the applicability of neural machine translation to Basque, a language with noteworthy characteristics that may represent a challenge for encoder-decoder approaches with attention mechanisms.

First, Basque is a synthetic language that features agglutinative morphology, i.e. where words can be formed via morphemic sequences, and a large number of case affixes that mark ergativity, datives, different types of locatives and genitives, instrumentality, comitativity or causality, among others. Verbal morphology is also relatively rare, displaying complex forms that include subject, direct object, indirect object and allocutive agreement markers, with number, tense and aspect being marked as well. This kind of rich morphology raises difficulties in terms of word representations and drastically increases data sparseness issues. A detailed description of Basque grammar can be found in (De Rijk and De Coene, 2008).

Secondly, although phrases in this language present a rather fixed inner order, as exemplified for instance by the regular structure of noun phrases,[1] at the sentential level the ordering is rela-

[1] Although regular, the structure of noun phrases may also be challenging, with left-branching relative clauses and affixa-

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 139–148
Alacant, Spain, May 2018.

tively free. Syntactically, order is essentially determined in terms of focus and topic. Although different orderings mostly reflect underlying variations according to these notions, for translation between Basque and languages with more rigid syntax the end-result is higher variability in terms of sentential input and output. Such variations may represent an additional challenge for NMT models that manage input information via learned attentional distributions and generate translations via decoding processes based on the previously generated element and beam searches.

Finally, Basque is a low-resourced language, with few publicly available parallel corpora. This is a third challenge for data-driven approaches in general, and NMT in particular as it usually requires larger training resources than statistical machine translation (Zoph et al., 2016).

To explore these challenges, we built several large neural machine translation models for generic Basque-Spanish translation, and compare their results with those obtained with rule-based and statistical phrase-based systems (Koehn et al., 2003). Our exploration of NMT variants for this language pair mainly focuses on different sub-word representations, obtained via either linguistically-motivated or frequency-based word segmentations, and on different data exploitation methods. We measure the impact of ordering variations partly via manually-created multiple references and also evaluate the impact of tuning the decoding process in terms of length and coverage along the lines of (Wu et al., 2016).

The paper is organised as follows: Section 2 describes related work in machine translation for Basque and other morphologically-rich languages; Section 3 presents the parallel corpora collected for the Basque-Spanish language pair; Section 4 describes the different translation models used for the experiments presented in Section 5; finally, Section 6 draws conclusion from this work.

2 Related work

Morphologically rich languages, a large denomination which includes synthetic languages where words are formed via productive morphological affixation, have been extensively studied in the machine translation literature. In Statistical Machine Translation (SMT) in particular (Brown et al., 1990), the data sparseness issues created by rich morphology have been addressed with a variety of techniques such as the factor-based translation (Koehn and Hoang, 2007). In Neural Machine Translation, the issues raised by rich morphology are even more acute given the vocabulary limitations for typical encoder-decoder neural networks, and recent work has centred on optimal methods to tackle surface variability and data sparseness in a principled manner.

Several approaches address morphological variation via character-based translation (Ling et al., 2015; Lee et al., 2016; Costa-Jussà and Fonollosa, 2016). A case study along these lines for languages with rich morphology is (Escolano et al., 2017), who implement a character-to-character NMT system augmented with a re-scoring model. They report improvements for Finnish-English translations but not for Turkish-English, although the latter result might be due to lack of sufficient training data.

Other approaches tackle this issue via word segmentation into sub-words. Byte Pair Encoding (BPE) (Sennrich et al., 2016) has become a popular segmentation method where infrequent words are segmented according to character pair frequencies. Alternatives include the use of morphological analysers such as MORFESSOR (Virpioja et al., 2013) or CHIPMUNK (Cotterell et al., 2015). Ding et al. (2016) compare the use of these three segmentation alternatives for Turkish-English, obtaining better results with CHIPMUNK and MORFESSOR than with BPE. In (Ataman et al., 2017), both supervised and unsupervised morphological segmentation are shown to outperform BPE for Turkish to English neural machine translation. Morphological decomposition has also been performed with tools tailored for the task, as is the case in (Sánchez-Cartagena and Toral, 2016), who report improvements using the rule-based morphological segmentation provided by OMORFI (Pirinen, 2015) for English-Finnish translation.

Finally, hybrid techniques have also been applied, as in (Luong and Manning, 2016) who built a character/word hybrid NMT system where translation is performed mostly at the word level and the character component is consulted for rare words. Their results for English to Czech demonstrate that their character models can successfully learn to generate well-formed words for a highly-inflected language. This approach has been notably applied

tion of determiners to the rightmost constituent in the noun phrase.

to English-Finnish by (Östling et al., 2017), who also include BPE segmentation in a system that ranked as the top contribution in the WMT2017 shared task for English-Finnish.

The challenges of machine translation of Basque have been addressed in different frameworks. An example-based data-driven system was thus developed by (Stroppa et al., 2006) and a rule-based approach was used to develop the MATXIN system for Spanish to Basque translation (Mayor et al., 2011); both systems are compared in (Labaka et al., 2007). In (Labaka et al., 2014), a hybrid architecture is presented, combining rule-based and phrase-based statistical machine translation approaches. Their hybrid system resulted in significant improvements over both individual approaches. In the next sections, we provide the first description of a large-scale NMT system for the Basque-Spanish language pair.

3 Corpora

To build representative translation models for the Basque-Spanish language pair, parallel corpora were collected and prepared from three different sources: professional translations in different domains, bilingual web pages, and comparable data in the news domain.

3.1 Parallel data

Parallel data for Basque-Spanish are scarce, the largest repository of such data being the professionally translated administrative texts made available in the Open Data Euskadi repository.[2] Amongst these, the largest collection comes from the translation memories of the *Instituto Vasco de Administración Pública* official translation services, with additional data from the *Diputación Foral de Guipúzcoa*. After filtering duplicate segments and dubious segments, we prepared the ADMIN corpus as our main parallel resource.

Additionally, we included four corpora from different domains. Two of them were created from translation memories, namely the SYNOPSIS corpus, a collection of film synopsis, and the IRRIKA corpus, based on content from the Irrika youth magazine. We also included corpora created via automatic alignment of bilingual documents: EIZIE, a corpus of universal literature, and CONSUMER, a collection of articles from Consumer consumption magazine. The EIZIE alignments were also manually revised to ensure a high quality corpus.

CORPUS	SENTENCES	WORDS	
	ES-EU	ES	EU
ADMIN	963,391	23,413,116	17,802,212
SYNOPSIS	229,464	3,501,711	2,824,807
IRRIKA	5,545	99,319	77,574
EIZIE	94,207	2,506,162	1,775,155
CONSUMER	201,353	3,999,156	3,313,798
TOTAL	1,493,960	33,519,464	25,715,972

Table 1: Parallel corpora statistics (unique segments)

CORPUS	SENTENCES	WORDS	
	ES-EU	ES	EU
CRAWL	1,044,581	19,892,360	15,344,336

Table 2: Crawled corpus statistics (unique segments)

The statistics for all parallel corpora are shown in Table 1.

3.2 Crawled data collection

To complement the high quality parallel data described in the previous section, we created a large parallel corpus from crawled data. We used the PACO2 tool (San Vicente and Manterola, 2012), which performs both crawling and alignment to create parallel resources from web corpora.

For this task, the tool was extended with two major optimisations. First, the crawling component was modified in order to retrieve web content dynamically generated via JavaScript. Secondly, both crawling and alignment processes were parallelised, to speed up the overall process.

Both optimisations contributed to the efficient creation of a parallel corpus from a variety of Basque-Spanish web pages. Corpus statistics, after duplicates removal, are shown in Table 2.

CORPUS	SENTENCES	WORDS	
	ES-EU	ES	EU
EITB	807,222	24,046,414	15,592,995

Table 3: Comparable corpus statistics (unique segments)

3.3 Comparable data collection

To further increase the amount of training data and extend domain coverage, we exploited a large strongly comparable corpus in the news domain, facilitated by the Basque public broadcaster EITB.[3] The original data consists of XML documents that contain news independently created by professional journalists in Basque and Spanish, from

[2] http://opendata.euskadi.eus

[3] http://www.eitb.eus/

CORPUS	SENTENCES	WORDS	
	ES-EU	ES	EU
MERGED	3,345,763	76,998,621	56,391,670
MERGED.LGF	3,086,231	61,529,688	47,976,559

Table 4: Final corpora statistics (unique segments)

which a first parallel corpus was created and shared with the research community (Etchegoyhen et al., 2016).

As additional data was made available since the first version of the corpus, we created a second version that included news from 2013 to 2016. For this version, document alignment was performed with DOCAL (Etchegoyhen and Azpeitia, 2016a), an efficient aligner that provided the best results for this language pair. Sentences were then aligned with STACC (Etchegoyhen and Azpeitia, 2016b), a tool to determine parallel sentences in comparable corpora which has proved highly successful for the task (Azpeitia et al., 2017).

After enforcing one-to-one alignments, the corpus resulted in $1,137,463$ segment pairs, each with an associated alignment probability score. Various corpora could then be extracted by selecting different alignment thresholds to filter low-scoring pairs. After training separate SMT models on each of these three corpora, we selected the corpus with alignment threshold 0.15, as it resulted in the best performance overall. Statistics for this corpus are shown in Table 3.

The EITB corpus was also used to prepare tuning and validation sets, as it covers a wide range of topics that includes politics, culture and sports, among others. Thus, $2,000$ segment pairs were selected as held-off development set, and $1,678$ as test set. The alignments for the test set were manually validated and a new set of references was manually created by professionally translating the Spanish source sentences, to account for word order variability in Basque.[4]

3.4 Data filtering & preparation

A unique parallel corpus (hereafter, MERGED) was built by concatenating the previously described corpora and removing duplicates. All sentences were truecased, with truecasing models trained on the monolingual sides of the bitext, and tokenised with adapted versions of the scripts available in the MOSES toolkit (Koehn et al., 2007).

Neural machine translation systems have been shown to be strongly impacted by noisy data (Belinkov and Bisk, 2017). As our gathered corpora comes from potentially noisy sources, as is the case with crawled and comparable data, we prepared an additional filtered version of the corpus. We based our filtering process on length irregularities between source and target sentences, in terms of number of words, with the aim of identifying those pairs where only a subset of a sentence is translated into the other language, a typical case in comparable corpora.

As a simple approach, we computed the modified z-score on the MERGED corpus to filter out segment pairs identified as outliers in terms of length difference between the source and target segments, where the median and standard deviation were computed on the human quality references of the ADMIN corpus. This process discarded $259,532$ segment pairs, as shown in Table 4, where MERGED.LGF refers to the filtered corpus.

4 Models

In the next subsections, we describe the different NMT models for Basque-Spanish that were built using the corpora described in the previous section, as well as the considered baseline systems.

4.1 Baselines

Two kinds of baseline systems were considered: statistical (SMT) and rule-based (RBMT).

All SMT translation models are phrase-based (Koehn et al., 2003), trained using the Moses toolkit (Koehn et al., 2007) with default hyperparameters and phrases of maximum length 5. Word alignment was performed with the FASTALIGN toolkit (Dyer et al., 2013), and the parameters of the log-linear models were tuned with MERT (Och, 2003). All language models are of order 5, trained with the KENLM toolkit (Heafield, 2011).

As an RBMT baseline translation system, we chose the on-line translation service provided by

[4] In what follows, the manually validated test will be referred to as ALNTEST, the human translations by HTTEST and the multi-reference test set as MULTIREF. Note that all test sets will be made available to the research community on our project web page: `http://modela.eus`.

the Basque Government, which is based on a proprietary rule-based system crafted for this language pair to provide general-domain translation.[5]

4.2 NMT

Unless otherwise specified, all NMT systems follow the attention-based encoder-decoder approach (Bahdanau et al., 2015) and were built with the OPENNMT toolkit (Klein et al., 2017). We use 500 dimensional word embeddings for both source and target embeddings. The encoder and the decoder are 4-layered recurrent neural networks (RNN) with 800 LSTM hidden units and a bidirectional RNN encoder. The maximum vocabulary size was set to $50,000$.

The models were trained using Stochastic Gradient Descent with an initial learning rate of 1 and applying a learning decay of 0.7 after epoch 10 or if no improvement is gained on the loss function after a given epoch over the validation set. A mini-batch of 64 sentences was used for training, with a 0.3 dropout probability applied between recurrent layers and a maximum sentence length set to 50 tokens for both source and target side.

The following subsections describe the neural machine translation variants that were prepared, the first three being based on different word segmentations and the last one on fully character-based translation.

4.2.1 Byte Pair Encoding

Byte Pair Encoding (BPE) is a compression algorithm that was adapted to word segmentation for NMT by (Sennrich et al., 2016). It iteratively replaces the most frequent pair of characters in a sequence with an unused symbol, without considering character pairs that cross word boundaries. BPE allows for the representation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences, having the advantage of producing symbol sequences still interpretable as sub-word units.

For our experiments, we trained joint BPE models on both Basque and Spanish data to improve consistency between source and target segmentation. A set of at most $30,000$ BPE merge operations was learned for each training corpus.

4.2.2 FLATCATV2

FLATCATV2 is a system based on MORFESSOR that was developed to implement a linguistically motivated vocabulary reduction for neural machine translation and was originally proposed for Turkish (Ataman et al., 2017). The segmentation process consists of two steps. MORFESSOR is used first to infer the morphology of the considered language in an unsupervised manner, based on an unlabelled monolingual corpus. The learned morphological segmentations are then fit into a fixed-size vocabulary, which amounted to $45,000$ tokens in our case.

Unlike the joint learning method we selected for BPE segmentation, FLATCATV2 segmentation was learned on the monolingual data separately, since this technique is designed to extract a linguistically-sound segmentation of the text.

4.2.3 Morphological analysis

As a third approach to word representation, we opted for a fine-grained morphological analysis and used the IXA-KAT supervised morphological analyser for Basque (Alegria et al., 1996; Otegi et al., 2016). This analyser relies on a lexicon crafted by linguists which includes most of the Basque morphemes and is used to extract all possible segmentations of a word. The hypotheses with the longest lemma are ultimately selected.

Although this linguistically-motivated approach to segmentation does reduce the vocabulary, vocabulary size is not guaranteed to remain within the range necessary for NMT. We therefore followed the two-step approach used in FLATCATV2 and applied BPE after the linguistic segmentation phase, to segment rare tokens and keep the vocabulary within the selected size.

4.2.4 Character-based translation

As an alternative to NMT architectures based on words or sub-words, character-based models provide the means to remove the segmentation problem altogether. These models are based solely on the characters in the sentence on both the input and the target sides, generating translations one character at the time. As previously discussed, this type of approach is particularly interesting for highly inflected languages such as Basque.

To evaluate this approach for Basque-Spanish translation, we built a character-to-character system following (Lee et al., 2016), whose code was publicly available.[6] The system uses convolutional

[5] http://www.itzultzailea.euskadi.eus/traductor/portalExterno/text.do

[6] https://github.com/nyu-dl/dl4mt-c2c

neural networks to generate window representations of fixed length character sequences, set to 5 in our configuration. These representations reduce the length of the input sequence, while enabling the system to identify segment patterns. A bi-directional recurrent neural network is then used to compute the representation of the complete sentence. Finally, translation is generated character by character, using an attention mechanism on the segments computed at the encoder level.

5 Experiments

In this section we first describe the experimental settings and system variants, then present and discuss the results.

5.1 Settings

To compare the different segmentation approaches, a first set of experiments was designed using only the selected EITB corpus. This allowed for a direct comparison between the approaches while also reducing the computational load of training the different variants. From this set of experiments, we selected the overall best approach to segmentation, taking into account the results obtained in both translation directions.

The second set of experiments compares NMT variants, based on the selected segmentation approach, to the SMT system. We also compare the NMT and SMT results with those obtained with the selected rule-based system on the single and multi-reference test sets.

The NMT approach based on the selected segmentation mechanism was trained on the entire corpus, as was the SMT system. Additionally, we evaluated the same NMT architecture and trained a model on the filtered corpus to assess the impact of noisy data on the final system.

We also evaluated the impact of the decoding optimisations proposed in (Wu et al., 2016), which tune the decoder according to length normalisation over the generated beam sequences and to the coverage of the input sequence according to the attention module. We also tuned the decoder with the end of sentence (EOS) penalty available as hyperparameter in OPENNMT. Optimal parameters for these three normalisations were evaluated via grid search, resulting in values of 4 for length, 0 for coverage, and 10 for EOS normalisations in ES-EU, and 10, 0 and 10 respectively in EU-ES.

Finally, we performed a small manual evaluation using the Appraise tool (Federmann, 2012). 28 native speakers of Basque were asked to select their preferred translations for 100 sentences, where the translations were generated by the previously described RBMT system and the NMT system trained on the entire corpus.

5.2 Results

Results in terms of automatic metrics were computed with BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). Tables 5 and 6 show the results for the different approaches to segmentation.[7]

The first noticeable result is the consistency of the scores across all test sets, in both directions. For ES-EU, there was no significant difference between the results obtained with BPE and with the unsegmented words, both achieving the best scores overall. In EU-ES, the optimal approach consistently involved applied linguistically-motivated segmentation first, followed by BPE to restrict the vocabulary size. In both directions, FLATCATV2 performed worse than BPE and character-based translation resulted in the lowest scores overall.

Linguistically-motivated segmentation for Basque was only beneficial on the source side, resulting in degraded results when compared to frequency-based segmentation on the target side. This result may be attributed to the stronger need to disambiguate source-side information in NMT architectures, where weak encoding impacts both sentence representation and the attention mechanism. As conditioned language models, NMT decoders seem to have lesser difficulties in generating correct output on the basis of non-linguistic but consistent segmentation units of the type provided by BPE.

From this first set of results, we selected BPE as our segmentation model for the final systems trained on the entire corpus, as it provided the best results when translating into Basque, was a competitive second ranked system in the other translation direction, and required less resources overall to perform segmentation. The comparative results between, RBMT, SMT and NMT are shown in Tables 7 and 8.[8]

[7] In both tables, † indicates statistical significance between the considered system and BPE, for $p < 0.05$. Significance was computed only for the BLEU metric, via bootstrap resampling (Koehn, 2004).

[8] In both tables, † indicates statistical significance between the considered system and NMT, for $p < 0.05$. Results are given on cased and tokenised output translations, after tokenising the output of the RBMT system for a fair comparison.

SEGMENTATION	VOCABULARY	ALNTEST		HTTEST		MULTIREF	
		BLEU	TER	BLEU	TER	BLEU	TER
WORDS	**50,004 / 50,004**	**19.82**	**64.84**	**18.53†**	**61.51**	**28.72**	**55.71**
BPE	**21,765 / 23,741**	**19.51**	**64.65**	**18.00**	**62.20**	**28.40**	**56.00**
FLATCATV2	38,653 / 29,860	18,23†	65,58	17.43†	62.58	27.13†	56.51
FLATCAT (ES) - MORF (EU)	38,653 / 50,004	16.98†	66.66	16.01†	64.09	25.32†	57.99
BPE (ES) - MORF+BPE (EU)	39,197 / 38,827	18.70†	65.31	17.51†	62.64	27.62†	56.36
CHARNMT	304 / 302	17.17†	67.59	16.23†	64.30	25.04†	59.01

Table 5: Evaluation results of the ES-EU systems using different data segmentation on the EITB corpus

SEGMENTATION	VOCABULARY	ALNTEST		HTTEST	
		BLEU	TER	BLEU	TER
WORDS	50,004 / 50,004	26.40	58.82	33.64†	50.08
BPE	23,741 / 21,765	26.61	58.16	35.71	47.67
FLATCATV2	29,860 / 38,653	24.46†	59.88	32.54†	50.43
MORF (EU) - FLATCAT (ES)	50,004 / 38,653	23.90†	60.80	31.06†	51.78
MORF+BPE (EU) - BPE (ES)	**38,827 / 39,197**	**27.86†**	**56.97**	**37.23†**	**46.33**
CHARNMT	304 / 302	24.58†	64.40	31.59†	57.66

Table 6: Evaluation results for EU-ES systems with different data segmentation on the EITB corpus

In Spanish to Basque, when considering all test sets, the best NMT system outperformed SMT, which in turn provided markedly better results than the RBMT system. Interestingly, the SMT system obtained the best BLEU score on the ALNTEST dataset, and was competitive with the basic NMT system for this metric on the MULTIREF test set as well, while being systematically outperformed on the TER metric by all NMT variants. These results might be due to the known BLEU bias in favour of SMT output, along with other biases (Callison-Burch et al., 2006), and the overall results therefore need to be interpreted by considering both metrics in conjunction. Thus, overall NMT performed markedly better, with gains above 4 BLEU points and 5 TER points on the MULTIREF metric. These constitute significant improvements, indicating that NMT responds better to the challenging properties of Basque than alternative approaches.

For Basque to Spanish translation, the comparative results were similar in terms of systems ranking and in terms of larger differences when considering human translations, used as source for this translation direction. As is usually the case, scores were higher when translating into the language with relatively simpler morphology.

Removing noise from the training corpus, via filtering outliers in terms of length differences, had a significant impact on ES-EU, where the MERGED.LGF model outperformed the non-filtered model by close to 3 BLEU points and 2 TER points on the MULTIREF test set. This confirms the importance of a careful preparation of training data for NMT models. For EU-ES, the filtered corpus gave statistically significant improvements as well, although by a lower margin.

Manual examination of the translations produced by the NMT system indicated that lost-in-NMT-translation phenomena, where the system ignores a significant portion of the input sentence in favour of a fluent but incomplete translation, were notable. The MERGED.LGF.OPT version of the system, where output generation is controlled via the previously described normalisation settings, improved on these grounds, both in terms of metrics and after manual examination of a subset of translations where coverage of the source content seemed to improve.

Another interesting aspect in these results is the impact of multiple references on the interpretation of the results. In most cases, taking into account only the initial test set based on alignments, all validated by human experts as proper translations, would have led to different conclusions than those reached when considering both the additional human translations and multiple references. One could have concluded, for instance, that the gains obtained for ES-EU with NMT over SMT were minor, when the differences were much larger overall when considering all references. The need for multiple references in general, and for this language pair in particular, is made even clearer from the results of these experiments.

Finally, Table 9 shows the results from the comparative human evaluation. Overall, users showed a marked preference for the translations produced by the NMT system, selecting RBMT translations in only 15.14% of the cases. Inter-annotator agree-

SYSTEM	ALNTEST		HTTEST		MULTIREF	
	BLEU	TER	BLEU	TER	BLEU	TER
RBMT	9.08†	79.90	14.01†	66.08	17.17†	66.37
SMT	**23.63**†	65.24	17.40†	61.66	30.43	56.50
NMT	20.46	64.52	23.63	55.39	31.27	53.54
NMT.LGF	22.09†	63.36	23.10†	55.10	34.17†	51.73
NMT.LGF.OPT	22.33†	**63.48**	**23.69**†	**54.47**	**34.65**†	**51.42**

Table 7: Final system evaluation results for ES-EU

SYSTEM	ALNTEST		HTTEST	
	BLEU	TER	BLEU	TER
RBMT	16.76†	69.28	25.06†	58.07
SMT	28.09	60.20	32.46†	52.79
NMT	27.68	55.37	39.21	42.58
NMT.LGF	27.99	55.09	39.73†	42.00
NMT.LGF.OPT	**29.02**†	**54.36**	**40.56**†	**41.26**

Table 8: Final system evaluation results for EU-ES

NMT>RBMT	NMT=RBMT	RBMT>NMT	SKIPPED
67.64%	15.39%	15.14%	1.82%

Table 9: Human evaluation results for ES-EU

ment measures showed fair agreement, with 0.306 and 0.309 for the Krippendorf's Alpha and Fleiss' Kappa metrics, respectively. Although admittedly based on a small sample, these results confirmed the impressions from users of the NMT system, who characterised it as a significant step forward in machine translation of Basque.

6 Conclusions

We presented the first results in neural machine translation for Basque, a synthetic language with an extended case system, complex verbal morphology and relatively free word order. The characteristics of the language made it an interesting test case for NMT and we showed that significant gains could be obtained with a neural network approach, when compared to both rule-based and statistical machine translation systems.

Several variants were prepared in terms of both corpora and models, to determine the optimal configurations for generic machine translation in Basque-Spanish. The impact of noisy datasets when training NMT systems was confirmed in our experiments, and we showed the improvements achievable with a simple filtering of length difference outliers.

Also coming from our results were the gains resulting from fine-grained morphological analysis on the source side, although byte pair encoding was shown to be a robust method overall for this language pair. The presented results were computed on different complementary test set, providing a strong confirmation of the importance of multiple references in general, and for the evaluation of Basque translation in particular.

Neural machine translation has been successfully applied to a large range of languages and scenarios, with recent work centred on languages with rich morphology. This work contributes to this line of research, demonstrating the significant improvements obtained with NMT on a language which features a wide range of properties that represent a challenge for past and current approaches to machine translation.

Acknowledgements: This work was supported by the Department of Economic Development and Competitiveness of the Basque Government via the MODELA project. We would like to thank the three anonymous EAMT reviewers for their helpful comments and suggestions, as well as all participants of the manual evaluation.

References

Alegria, Iñaki, Xabier Artola, Kepa Sarasola, and Miriam Urkia. 1996. Automatic morphological analysis of Basque. *Literary and Linguistic Computing*, 11(4):193–203.

Ataman, Duygu, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, 108:331–342.

Azpeitia, Andoni, Thierry Etchegoyhen, and Eva Martinez Garcia. 2017. Weighted Set-Theoretic Alignment of Comparable Sentences. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 41–45.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Belinkov, Yonatan and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Bojar, Ondřej et al. 2016. Findings of the 2016 conference on machine translation (WMT2016). In *Proceedings of the First Conference on Machine Translation*, WMT2016, pages 131–198, Berlin, Germany.

Bojar, Ondřej et al. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, WMT2017, pages 169–214, Copenhagen, Denmark.

Brown, Peter F, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256.

Costa-Jussà, Marta R. and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 357–361, Berlin,Germany.

Cotterell, Ryan, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 164–174, Beijing, China.

Crego, Josep, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

De Rijk, Rudolf PG and Armand De Coene. 2008. *Standard Basque: A progressive grammar*. MIT Press Cambridge, MA.

Ding, Shuoyang, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. 2016. The JHU machine translation systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation*, WMT2016, pages 272–280, Berlin, Germany.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Escolano, Carlos, Marta R. Costa-jussà, and José A. R. Fonollosa. 2017. The TALP-UPC Neural Machine Translation System for German/Finnish-English Using the Inverse Direction Model in Rescoring. In *Proceedings of the Second Conference on Machine Translation*, WMT2017, pages 283–287, Copenhagen, Denmark.

Etchegoyhen, Thierry and Andoni Azpeitia. 2016a. A Portable Method for Parallel and Comparable Document Alignment. *Baltic Journal of Modern Computing*, 4(2):243–255. *Special Issue: Proceedings of EAMT 2016.*

Etchegoyhen, Thierry and Andoni Azpeitia. 2016b. Set-Theoretic Alignment for Comparable Corpora. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1: Long Papers, pages 2009–2018, Berlin, Germany.

Etchegoyhen, Thierry, Andoni Azpeitia, and Naiara Pérez. 2016. Exploiting a Large Strongly Comparable Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Federmann, Christian. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Gehring, Jonas, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*.

Heafield, Kenneth. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT 2011, pages 187–197, Edinburgh, Scotland.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1701.02810*.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 joint conference on Empirical Methods in Natural Language processing and Computational Natural Language Learning (EMNLP, CoNLL)*, pages 868–876, Prague, Czech Republic.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics, (HLT-NAACL)*, pages 48–54, Edmonton, Canada.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 177–180, Prague, Czech Republic.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain.

Labaka, Gorka, Nicolas Stroppa, Andy Way, and Kepa Sarasola. 2007. Comparing Rule-Based and Data-Driven Approaches to Spanish-to-Basque Machine Translation. In *Proceedings of MT-Summit XI*, pages 297–304.

Labaka, Gorka, Cristina España-Bonet, Lluís Màrquez, and Kepa Sarasola. 2014. A hybrid machine translation architecture guided by syntax. *Machine Translation*, 28:91–125.

Lee, Jason, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.

Ling, Wang, Isabel Trancoso, Chris Dyer, and Alan Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Luong, Minh-Thang and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Mayor, Aingeru, Iñaki Alegria, Arantza Díaz de Ilarraza, Gorka Labaka, Mikel Lersundi, and Kepa Sarasola. 2011. Matxin, an open-source rule-based machine translation system for basque. *Machine Translation*, 25:53–82.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, ACL '03, pages 160–167, Sapporo, Japan.

Östling, Robert, Yves Scherrer, Jörg Tiedemann, Gongbo Tang, and Tommi Nieminen. 2017. The Helsinki Neural Machine Translation System. In *Proceedings of the Second Conference on Machine Translation*, WMT2017, pages 338–347, Copenhagen, Denmark.

Otegi, Arantxa, Nerea Ezeiza, Iakes Goenaga, and Gorka Labaka, 2016. *A Modular Chain of NLP Tools for Basque*, pages 93–100. Springer International Publishing.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Pirinen, Tommi A. 2015. Development and use of computational morphology of Finnish in the open source and open science era: Notes on experiences with OMorFi development. *SKY Journal of Linguistics*, 28:381–393.

San Vicente, Inaki and Iker Manterola. 2012. Paco2: A fully automated tool for gathering parallel corpora from the web. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, LREC2012, pages 1–6, Istanbul, Turkey.

Sánchez-Cartagena, Víctor M. and Antonio Toral. 2016. Abu-MaTran at WMT 2016 Translation Task: Deep Learning, Morphological Segmentation and Tuning on Character Sequences. In *Proceedings of the 1st Conference on Machine Translation*, WMT2016, Berlin, Germany.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725, Berlin,Germany.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Stroppa, Nicolas, Decan Groves, Andy Way, and Kepa Sarasola. 2006. Example-based machine translation of the basque language. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 232–241, Cambridge, MA USA.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, NIPS, pages 3104–3112, Montreal, Canada.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, NIPS, pages 6000–6010, Montreal,Canada.

Virpioja, Sami, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. *Aalto University publication series SCIENCE + TECHNOLOGY; 25/2013*.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

Evaluation of Terminology Translation in Instance-Based Neural MT Adaptation

M. Amin Farajian[1,2], **Nicola Bertoldi**[1], **Matteo Negri**[1], **Marco Turchi**[1], **Marcello Federico**[1]
[1] Fondazione Bruno Kessler, Trento, Italy
[2] University of Trento, Trento, Italy
{farajian,bertoldi,negri,turchi,federico}@fbk.eu

Abstract

We address the issues arising when a neural machine translation engine trained on generic data receives requests from a new domain that contains many specific technical terms. Given training data of the new domain, we consider two alternative methods to adapt the generic system: *corpus-based* and *instance-based* adaptation. While the first approach is computationally more intensive in generating a domain-customized network, the latter operates more efficiently at translation time and can handle on-the-fly adaptation to multiple domains. Besides evaluating the generic and the adapted networks with conventional translation quality metrics, in this paper we focus on their ability to properly handle domain-specific terms. We show that instance-based adaptation, by fine-tuning the model on-the-fly, is capable to significantly boost the accuracy of translated terms, producing translations of quality comparable to the expensive corpus-based method.

1 Introduction

When deployed in production lines, machine translation (MT) systems need to serve requests from various domains (*e.g.* legal, medical, finance, sports, etc.) with a variety of structural and lexical differences. Considering that technical translation (*e.g.* user guides, medical reports, etc.) represents the largest share in the translation industry (Kingscott, 2002) and that a significant part of it deals with domain-specific terms, it is important that machine translation delivers not only generic quality but also accurate translations of terms. The possibility of bearing different meanings in different contexts increases the difficulty of translating terms, making it an interesting and challenging topic in MT. Table 1 shows two examples in which Google Translate[1] (GT) and Bing translator[2] (BT) wrongly translate domain terminology. In the first example, the English word *apple* is wrongly recognized and translated as a term of the computer domain (*apple*) while it actually refers to the fruit type (*mele*). In the second example, on the contrary, Bing fails to recognize the multi-word term *broken Windows* by producing instead a literal translation that departs from the original sense. These examples show that existing MT systems still have difficulties in handling domain-specific terms, which calls for solutions to improve this aspect of MT.

Ideal solutions for this real-world multi-domain translation scenario should be scalable enough to enable the industrial deployment at a reasonable cost, while guaranteeing a high level of flexibility in delivering good-quality translations for all (or most of) the domains. This is of higher importance for the neural approach, where building the systems usually requires expensive GPU machines trained for several days to weeks on large amounts of parallel data.

In this paper we analyze the ability of instance-based adaptation strategy in handling domain terminology (technical terms) and compare its performance with a non-adaptive generic neural MT (NMT) system trained on a large pool of parallel data, and a corpus-based adaptive NMT system as

[1] https://translate.google.com

[2] https://www.bing.com/translator

Src.	Composition and nutritive value of apple products.
GT	Composizione e valore nutritivo dei prodotti apple.
Ref.	Composizione e valore nutritivo dei prodotti a base di mele.
Src.	It also contains system recovery tools you can use to repair broken Windows.
BT	Esso contiene anche gli strumenti di ripristino del sistema possibile utilizzare per riparare le finestre rotte.
Ref.	Esso contiene anche gli strumenti di ripristino del sistema che possibile utilizzare per riparare Windows non funzionante.

Table 1: Examples of incorrectly translating technical terms from English into Italian by online translation engines. Translation queries submitted on 29/03/2018. GT and BT refer to Google Translate and Bing translator, respectively.

a strong (and expensive) term of comparison.

Our results show that, in contrast to the generic and corpus-based adaptive solutions which compromise either the translation quality or the architectural cost, recently proposed instance-based adaptation methods (Farajian et al., 2017b) provide a flexible solution at reasonable costs. This adaptive system is based on a retrieval mechanism that, given a test sentence to be translated, extracts from the pool of parallel data the top (*source*, *target*) pairs in terms of similarity between the *source* and the test sentence. Using this small set of retrieved pairs, it then fine-tunes the model, and applies it to translate the input sentence. As shown in (Farajian et al., 2017b), by applying local adaptation to few training instances, not only the system is able to improve the performance of the generic NMT but, in some domains, it can also outperform strong specialized corpus-based NMT systems trained for several epochs on the corresponding domain-specific data.

In this paper, we further explore the effectiveness of the instance-based adaptation method reporting, in addition to global corpus-level BLEU scores, empirical results on how they perform in translating domain terminology. To this aim, we divide the terms into two categories of single- and multi-word phrases, and study the systems' behaviour in each class separately. Unsurprisingly, in both cases corpus-based adaptation improves the performance of the generic model by a large margin. Such improvements, however, come at the cost of computationally intensive adaptation on all the in-domain data. In contrast, instance-based adaptation achieves comparable results with a less demanding strategy based on adapting the model to few training instances retrieved from the pool of data on the fly. This empirical proof, focused on the proper treatment of domain terms in NMT adaptation, is the main contribution of this paper.

2 Related works

When exposed to new domains (Koehn and Knowles, 2017) or applied in multi-domain scenarios (Farajian et al., 2017a), machine translation systems in general and neural MT in particular, experience performance degradations due to the distance between the target domain and the domain(s) on which they were trained. Previous studies on multi-domain MT provide solutions for this issue, making it possible to cover more than one domain while reducing performance degradations in the target domains (Luong and Manning, 2015; Chen et al., 2016; Zhang et al., 2016; Freitag and Al-Onaizan, 2016; Chu et al., 2017; Farajian et al., 2017b; Kobus et al., 2017). These solutions can be categorized into *static* and *adaptive* approaches. To train one single model using heterogeneous data from many domains, static approaches assume to have simultaneous access to all the training data and their corresponding domain/topic information. This information, which is either manually assigned or automatically inferred from the data, is passed as additional signal to the MT system, helping it to produce higher quality translations for the desired target domain. Existing solutions in the field of NMT propose to incorporate this topic/domain information only on the source side (*i.e.* to support the encoder) (Kobus et al., 2017), only on the target side (*i.e.* to support the decoder) (Chen et al., 2016), or on both sides (Zhang et al., 2016). Although consistent and significant improvements have been reported by this approach, its application to new domains is not trivial, mostly due to the fact that it requires performing expensive NMT and topic model adaptations using the original multi-domain data and the training set for the new domain.

Adaptive approaches, on the other hand, propose to fine tune an existing MT system, trained either on another domain or pool of parallel data, to the new domain or task. While Luong and Manning (2015) report significant improvements by this approach on the new target domain, Freitag and Al-Onaizan (2016) observe a significant drop in system's performance on the original domain, which is due to the severe overfitting of the model to the new domain. To solve this issue, they propose a slightly different approach, which performs ensemble decoding using both the adapted and the generic model. The *mixed fine tuning* method proposed by Chu et al. (2017) is another approach for keeping under control the performance degradation on the original out-domain data while adapting the model to the new domain. Given the out-domain and in-domain training sets and a model pre-trained only on the out-domain data, this approach continues the training on a parallel corpus that is a mix of the two training corpora, in which the smaller in-domain corpus is oversampled to have the same size as the larger out-domain corpus. The specialized models obtained by these adaptation techniques are empirically shown to be effective, improving the translation quality of a generic NMT system on the target domains. However, the practical adoption of this approach results in developing and maintaining multiple specialized NMT engines (one model per domain), which increases the infrastructure's costs and limits its scalability in real-world application scenarios.

To combine the advantages of the two worlds, (*i.e.* to get close to the high quality of corpus-based adaptation still keeping the scalability of one single model), Farajian et al. (2017b) introduce an instance-based adaptation method for NMT inspired by Hildebrand et al. (2005). Instead of adapting the original generic model to the whole in-domain training corpus, the instance-based method retrieves from the pool of parallel data a small set of sentence pairs in which the source side is similar to the test sentence. Then, it fine-tunes the generic model on-the-fly by using the set of retrieved samples. This makes the instance-based adaptive approach a reasonable solution for real-world production lines, in which the MT system needs to cover a wide range of application domains while keeping under control the architecture's cost.

In addition to the architectural costs of NMT deployment in multi-domain application scenarios, there is another important factor that has to be considered, that is their ability in translating domain-specific words and phrases (*i.e.* terms). Based on their nature, these expressions can be frequently observed in their corresponding domains, being at the same time infrequent or even out of vocabulary (OOV) in the other domains. Nevertheless, data-driven MT systems need to be trained on large amounts of training data, which is generally collected from different sources. This further reduces the relative frequency of these words, making them less probable to be translated correctly by the system. This makes it even more challenging for NMT approach where rare and OOV words are either segmented into their corresponding sub-word units (Sennrich et al., 2016) or mapped to a special "`unk`" token (Luong and Manning, 2015). However, in the relatively recent history of NMT, there are few works that analyze its behavior focusing on domain terminology. Chatterjee et al. (2017) achieve significant improvements with a guide mechanism that helps the NMT decoder to prioritize and adequately handle translation options obtained from terminology lists. Arčan and Buitelaar (2017) empirically show that offline adaptation of a generic NMT system to the new domain improves its performance in translating domain-specific terms. However, they discuss only corpus-based adaptation techniques that, compared to instance-based methods, are less suitable for real-world application. Moreover, they mostly work in a setting in which domain terminology has to be translated in isolation without any context, while in our working scenario we work with full sentences.

3 Neural machine translation adaptation

In this section we first briefly review the state-of-the-art sequence-to-sequence neural machine translation and then describe the two corpus-based and instance-based adaptation approaches.

3.1 Neural machine translation

We build our adaptive systems on top of the state-of-the-art attention-based encoder-decoder neural MT (Bahdanau et al., 2015) in which given the source sentence $x = x_1, ..., x_M$, the translation is modeled as a two-step process. The source sentence x is first encoded into a sequence of hidden states by means of a recurrent neural network (RNN). Then, another RNN decodes the source hidden sequence into the target string. In particular, at each time step the decoder predicts the next target word from the previously generated target word, the last hidden state of the decoder, and a weighted combination of the encoder hidden states, where the weights are dynamically computed through a feed-forward network, called attention model.

Training of the presented NMT architecture is generally carried out via maximum-likelihood estimation, in which the model parameters such as word embedding matrices, hidden layer units in both the encoder and decoder, and the attention model weights are optimized over a large collection of parallel sentences. In particular, starting from a random initialisation of the parameters, optimization is performed via stochastic gradient descent (Goodfellow et al., 2016), in which at each iteration a randomly selected batch β is extracted from the data and each parameter θ is moved one step in the opposite direction of the mean gradient of the log-likelihood (L), evaluated on the entries of β:

$$\Delta\theta = -\eta \frac{1}{\mid \beta \mid} \sum_{(x,y)\in\beta} \frac{\partial L(x,y)}{\partial\theta} \qquad (1)$$

where η is a hyperparameter moderating the size of the step $\Delta\theta$ and is usually referred to as the learning rate. It can either be fixed for all parameters and all iterations, or vary along one or both dimensions (Goodfellow et al., 2016). During training, the optimization is performed by going through several so-called epochs, *i.e.* the number of times the whole training data is processed.

3.2 Corpus-based adaptation in neural MT

Given an existing NMT model, trained either on another domain or on a generic pool of parallel data, corpus-based adaptation methods fine-tune the model parameters by applying the same training procedure described in Section 3.1. Depending on the application scenario, the optimization is performed by iterating over a combination of both the current and new data (Chu et al., 2017) or only the training data of the new domain (Luong and Manning, 2015). The former is usually used when the goal is to adapt the model to the new domain while avoiding performance degradation in the domain on which the model was initially trained. Otherwise, only the training data of the new domain is used. In this paper, we opt for the latter solution because we are interested only in the performance of the system in the new domain.

These solutions, however, require a few hours to fine-tune the system to the target domains, which is scarcely compatible with application scenarios in which users need to instantly start interacting with the MT system. In spite of this (and the inherent cost and scalability-related issues), the competitiveness of this solution motivates its adoption as a strong term of comparison in this paper.

3.3 Instance-based adaptation in neural MT

Instance-based adaptation (Farajian et al., 2017b) is an extension of the aforementioned adaptation method, in which, instead of adapting the model to all the available in-domain training data, only few instances (*i.e.* sentence pairs) are used to tune the model. In particular, given an already existing NMT model, the pool of in-domain parallel data, and a sentence to be translated, it performs the following three steps: 1) retrieve from the pool a set of (*source, target*) pairs in which the *source* is similar to the test sentence; 2) locally adapt the parameters of the model using the retrieved sentence pairs; 3) translate the given test sentence by applying the resulting locally-tuned model. The diagram of the approach is shown in Figure 1. In order to leverage more effectively the information of the retrieved training samples, Farajian et al. (2017b) propose to set the hyperparameters of the training process (*i.e.* learning rate and number of epochs) proportional to the similarity of the retrieved set to the test. This results in fine tuning the model with larger learning rates and for more epochs when the retrieved sentence pairs are highly similar to the test and vice versa.

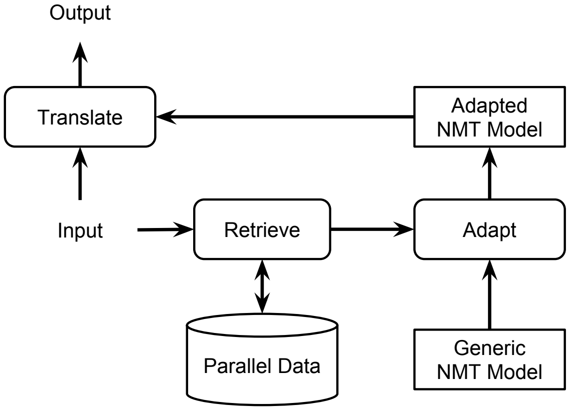


Figure 1: Instance-based NMT adaptation.

4 Experimental setup

4.1 Data

The experiments of this paper are carried out in the English-Italian translation direction. The train-

	Segments	Tokens	Types
Generic	20.8M	373.5M	1.7M
Gnome	76.5K	685.2K	36.0K
KDE4	179.5K	2.1M	75.3K

Table 2: Statistics of the Italian side of the training corpora. Generic data is used for training the generic NMT system, while the domain-specific data (*i.e.* Gnome and KDE4) are used only in the adaptation step.

ing set consists of a large collection of proprietary data collected from several industrial translation projects in different domains (*i.e.* medical, software documentations, user guides, etc.). The statistics of the training corpus are presented in Table 2 (first row).

To evaluate the performance of the systems in translating domain-specific terms we need test sets in which the terms are annotated. Moreover, both adaptive systems need in-domain training data in order to fine tune the generic model to the given domain. This further increases the difficulty of finding the evaluation data. The *BitterCorpus*[3] (Arčan et al., 2014) is a collection of parallel English-Italian documents in the information technology (IT) domain (extracted from Gnome and KDE4 projects) in which technical terms are manually marked and aligned. However, this corpus is not ready to be used in our task *as-is*, since: *i)* it contains only the annotated test data without any in-domain training set, and *ii)* test data are aligned at document level, while in our experiments we need sentence-level aligned corpora.

In order to compile an evaluation package that addresses our needs, we used the publicly available Gnome and KDE4 corpora[4] which are sentence-level aligned, divided them into training and test sets, and then automatically annotated the terminologies in the test by means of the term list extracted from the BitterCorpus[5]. The statistics of the Italian side of the training and test corpora are reported in Table 2 and 3. Some examples of the English terms and their corresponding Italian translations are presented in Table 4. As we see, there are several cases where, in addition to the specific translations used in IT domain (marked with *), the English term can have other translations that are valid in other domains. For example, depending on the domain, the English word *button* can refer to the object used to fasten something (*i.e.* in Italian referred to as *bottone*), or the electrical/electronic equipment that is pressed to turn on or off a device (*i.e.* translated as *pulsante* in Italian). This ambiguity is usually observed in the case of single-word terms, while multi-words often disambiguate each other.

	Seg.	Avg. Len.	Terms		
			single word	multi word	all
Gnome	2000	20.5	2,660	183	2,843
KDE4	2000	25.7	3,767	256	4,023

Table 3: Statistics of the Italian side of the test corpora.

English	Italian
list	lista*, elenco*
path	path*, percorso*, indirizzo*
button	pulsante*, bottone
toolbar	barra degli strumenti
wrapping	a capo automatico*, avvolgere
title bar	barra del titolo
wildcards	caratteri jolly
tree view	vista ad albero
konversation	konversation
mouse pointer	puntatore del mouse
destination folder	cartella di destinazione
regular expression	espressione regolare
right mouse button	tasto destro del mouse

Table 4: Examples of term pairs in our test corpora. Words marked with * represent in-domain translations of the term.

4.2 NMT systems

We conducted the experiments with our in-house developed and maintained branch of the OpenNMT-py toolkit (Klein et al., 2017), which is an implementation of the attention-based encoder-decoder architecture (Bahdanau et al., 2015). Our code is integrated with the open source `ModernMT` project[6], and is highly optimized and already deployed for production systems. In our experiments, we segmented the infrequent words into their corresponding sub-word units by applying the byte pair encoding (BPE) approach (Sennrich et al., 2016). In order to increase the consistency between the source and target segmentations, we learned the BPE merge rules from the concatenation of the source and target side of the training data. We set the number of merge rules to 32K, resulting in vocabularies of size 34K and 35K respectively for English and Italian. We used 2-layered LSTMs in both the encoder and decoder, each of which containing 500 hidden units. We

[3] `https://hlt-mt.fbk.eu/technologies/bittercorpus`

[4] `http://opus.lingfil.uu.se/`

[5] `https://gitlab.com/farajian/TermTraGS`

[6] `www.modernmt.eu`

set the word embedding size to 500 for both the source and target languages. The parameters are optimized with SGD using the initial learning rate of 1.0 with a decaying factor of 0.9. The batch size is set to 64, and the model is evaluated after each epoch. We trained the system for 11 epochs and selected the model with the highest BLEU score on our development set.

Our reimplementation of the instance-based adaptive system uses the open source Lucene library (McCandless et al., 2010) to store the training samples (*i.e.* pool of the generic and domain-specific data). Similar to (Farajian et al., 2017b), given the test sentence it retrieves the most similar instance from the pool (*i.e.* top-1) and adapts the aforementioned generic model accordingly. It sets the hyperparameters of the fine-tuning process proportional to the similarity of the retrieved instance and the test sentence. For example, it fine-tunes the model with the learning rate of 0.2 for 10 iterations if the similarity of the retrieved instance to the test is 1.0. In this work we used sentence-level BLEU (Chen and Cherry, 2014) as the similarity measure. In our experiments, the average time for updating the model was about 0.5 seconds per sentence.

The corpus-based adapted NMT systems are multiple instances of the generic system each of which trained on the corresponding domain-specific training data for several epochs, until no improvement is observed in the model perplexity on our development set for four consecutive epochs. We then used, for each domain, the model with minimum perplexity on the development set (*i.e.* model obtained after 26 and 4 epochs respectively for Gnome and KDE4). We used the same settings as the generic system for training these systems. However, for fine tuning we started with a learning rate of 0.5. In our experiments, the corpus-based adaptation of the model took about 3:00 and 1:15 hours for Gnome and KDE4 domains, respectively.[7]

4.3 Evaluation metrics

We evaluate the systems' performance both in terms of BLEU (Papineni et al., 2002) and *term hit rate* (THR). While the former measures the overall quality of the translations with respect to the manually-translated reference, the latter analyzes the ability of the system in learning the vocabulary of each specific domain. To this aim it computes the proportion of terms in the reference that are correctly translated by the MT system. However, in order to avoid assigning higher scores to the systems which over-generate the same term, it clips the counts of the matched terms by their frequency in the reference (2).

$$THR = \frac{\sum\limits_{term \in ref} count_{clip}(term)}{\sum\limits_{term \in ref} count_{ref}(term)} \qquad (2)$$

Since there are two types of single-word and multi-word terms in our test sets, in order to have a more detailed analysis of systems' behaviour we report the scores for each class separately in addition to the overall THR score.

5 Analysis and discussion

In this section we present a detailed analysis of the results obtained by different systems and compare the systems in translating the technical terms in Gnome and KDE4.

5.1 Translation quality

Table 5 reports the performance of the generic, instance-based, and corpus-based adaptive systems on Gnome and KDE4 test sets in terms of BLEU. As the results show (first two rows), the instance-based system significantly improves the performance of the generic system by +7.80 and +6.55 BLEU points. However, it obtains a lower BLEU score compared to its corpus-based counterparts. In our investigations, we noticed that in almost all the cases where the application domain is new (*i.e.* the training data of the target domain was not included in the pool of data used for training the generic system), the corpus-based adapted system produces translations of higher quality compared to the instance-based system. Nevertheless, this comes at the cost of training the system for several hours, instead of instantly starting the translation process.

Another interesting phenomenon that we observed in these experiments is the correlation of the performance gain and the average similarity of the retrieved samples to the test sentences. We noticed a larger performance gain in the case of Gnome compared to KDE4 (+7.80 vs. +6.55) while the average similarity of the retrieved sentence pairs in this domain is lower (0.36 compared to 0.56). Comparing the ratio of the successful retrievals in

[7] We carried out the experiments on Azure instances with NVIDIA Tesla k80 GPUs.

	Avg. Sim.	Generic	Instance based	Corpus based
Gnome	0.36	35.97	43.77	49.79
KDE4	0.56	35.09	41.64	46.26
Gnome †	0.43	38.06	51.36	56.00
KDE4 †	0.61	36.99	51.84	48.95

Table 5: BLEU score of the generic and adaptive NMTs on the test sets. The corpora marked with †are subsets of the original corpora for which a similar instance is retrieved.

the two systems partially explains this behaviour: in the case of Gnome, in 83.9% of the cases the system is able to find training samples similar to the test while in KDE4 this figure decreases to 75.8%. Moreover, by limiting our analysis to these cases, *i.e.* sentences for which the system has successfully retrieved a similar instance (last two rows in Table 5), we see a correlation higher than 0.9 between the performance gain and the similarity. Even more surprisingly, we observe that on this subset of KDE4 corpus the instance-based system outperforms its corpus-based counterpart. This is mostly due to the fact that retrieved instances in this case are highly similar to the test sentences.

5.2 Term translation

Table 6 presents the performance of the systems on both Gnome and KDE4 data. Since a large portion of the generic training data belongs to the IT domain we observe a reasonably high performance by the generic system in the studied domains, in particular on the single-word terms (79.58 and 73.70 on Gnome and KDE4 domains, respectively). However, translating multi-word terms is more challenging for all the systems as it involves producing sequences of words that might have several translations in different context. For example the English words *bar*, *path*, and *mouse* are usually translated into *bar*, *indirizzo*, and *topo* while their contextual translations in the technical terms *title bar*, *full path* and *mouse pointer* is *barra*, *path*, and *mouse*. This makes the translation more difficult for the systems, resulting in a significant performance drop compared to the case of single-word terms.

	Term Type	Generic	Instance based	Corpus based
Gnome	Single-word	79.58	82.16	86.55
	Multi-word	62.79	70.54	80.62
	Overall	78.59	81.48	86.20
KDE4	Single-word	73.70	79.48	81.94
	Multi-word	48.15	58.52	61.48
	Overall	72.24	78.28	80.78

Table 6: Performance of the generic and adaptive NMTs on the test sets, in terms of THR.

	Term Type	English	Italian
Gnome	Single-word	70.1	62.0
	Multi-word	60.0	51.6
	Overall	69.7	61.4
KDE4	Single-word	71.7	59.5
	Multi-word	68.2	45.5
	Overall	71.5	58.7

Table 7: Percentage of the retrieved samples that contain the desired terms.

5.3 Instance selection effect

In addition to the similarity of the retrieved samples to the test discussed in (Farajian et al., 2017b), the presence of domain terms in the retrieved sentence pairs is another important factor for instance-based adaptation. As Table 7 shows, in about 30% of the cases the retrieved English sentence does not contain the desired term. This proportion is even higher if we look at the target side of the retrieved instances, in which around 40% of the desired term translations are missing. However, this is expected since the retrieval is performed only based on the source side information (*i.e.* in our experiments English), with no additional filters based on the target side of the retrieved instance. Measuring the performance of the adaptive system in correcting the terms which are missed by the generic system shows that the instance-based system effectively learns the vocabulary of the application domain, correcting up to 76.64% of the mistakes made by the generic system if the desired term translation exists in the retrieved instance (Table 8).

6 Further analysis

In addition to the automatic evaluations we performed further manual analysis on the outputs of the instance-based adaptive system. The results of this analysis indicate that, compared to the generic system, its behavior differs in two main aspects: *i)* learning to translate the terms that are missed or wrongly translated by the generic system, *ii)* adapting to different style of the translation. When run on new domains, for which it has not seen any

	Single-word	Multi-word	Overall
Gnome	64.33	52.94	63.22
KDE4	76.92	73.91	76.64

Table 8: Percentage of the terms corrected by the instance-based adaptive system.

in-domain training data, it is highly probable that the generic system receives translation requests containing terms which are OOV or infrequently observed in the training data. In such cases, even after applying BPE, it might not be able to produce proper translations. As an example, the English word *dolphin*, which is rarely observed in the generic training data, is always translated in the Italian word *delfino* which refers to the animal. However, in the KDE4 domain it corresponds to a proper noun that indicates a file manager application. As we see in Table 9, the generic system wrongly translates it into *delphin*. By accessing in-domain training data (*i.e.* either the full corpus or just one single, highly similar instance), both the adaptive systems are able to correctly translate it.

The English terms *Control Center* and *mouse cursor* are two interesting examples of learning domain-specific translation styles. While in the generic training data these terms are usually translated into *Control Center* and *cursore del mouse*, in the KDE4 domain the human translators prefer them to be translated into *centro di controllo* and *puntatore del mouse*. As we see in the examples of Table 9, the generic system produces their commonly used translations, while the instance-based system is able to learn and produce the desired domain-specific translations.

We also observed a few cases in which the instance-based approach learns to properly generate Italian terms in the translation while there is no corresponding source English term in the given test sentence. The Italian word *pulsante* in the fourth example provided in Table 9 is one of these cases. As we see, the input English sentence does not contain the word *button*, hence both the generic and corpus-based adapted NMT systems do not produce any translation for it. On the contrary, the instance-based system, being trained on a very similar instance which contains the word *pulsante*, learns the pattern and produces a translation that is closer to the reference.

Finally, we noticed that inconsistent translations of the terms can affect the instance-based adaptive system, resulting in translations which are different than the manually produced references. The last example provided in Table 9 shows one of these cases. As we see, the English term *packages* can be translated into either *pacchetti* or *package*. So, based on the suggestion provided by the retrieval module, the instance-based system learns to translate it into *package* which is another valid translation of this term. This, however, does not affect the global performance of the system due to the small amount of similar situations.

7 Conclusions

We investigated the application of instance-based adaptive NMT in a real-world scenario where translation requests come from new domains that contain many technical terms. In particular, we analyzed its ability to properly handle domain terminology, comparing its output against the translations produced by a generic (unadapted) NMT system and a corpus-based specialized NMT system. Overall, our experiments with Gnome and KDE4 data reveal that the two adaptation methods significantly improve the performance of the generic system both in terms of global BLEU score and *term translation accuracy*. Unsurprisingly, by performing a computationally intensive fine tuning on the full in-domain training data, corpus-based adaptation produces specialized NMT systems that achieve better results at the cost of reduced scalability. However, the less demanding instance-based adaptation (performed on one parallel sentence pair retrieved from a pool of data based on its similarity to the test sentence), is able to effectively learn domain terms' translations, even for expressions that were never observed by the generic model. Such capability allows instance-based adaptation to significantly reduce the gap between generic and corpus-based specialized NMT models at manageable costs.

Acknowledgements

This work has been partially supported by the EC-funded H2020 projects QT21 (grant no. 645452) and ModernMT (grant no. 645487). This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft.

References

[Arčan and Buitelaar2017] Arčan, Mihael and Paul Buitelaar. 2017. Translating domain-specific expressions in knowledge bases with neural machine translation. CoRR. http://arxiv.org/abs/1709.02184.

[Arčan et al.2014] Arčan, Mihael, Marco Turchi, Sara Tonelli, and Paul Buitelaar. 2014. Enhancing statistical machine translation with bilingual terminol-

Source	[...]Files may be dragged and dropped onto & kwrite; from the Desktop, the filemanager & dolphin;[...]
Reference	[...]I file possono essere trascinati e rilasciati su & kwrite; dal Desktop, & dolphin;[...]
Ret. Src.	[...]Files may be dragged and dropped onto & kate; from the Desktop, the filemanager & dolphin;[...]
Ret. Trg.	[...]I file possono essere trascinati e rilasciati in & kate; dal Desktop, dal gestore di file & dolphin;[...]
Generic	[...]I file possono essere trascinati e rilasciati su & kwrite; dal Desktop, dal file manager e dal delphin;[...]
Instance-based	[...]I file possono essere trascinati e rilasciati in & kwrite; dal Desktop, dal gestore di file & dolphin;[...]
Corpus-based	[...]I file possono essere trascinati e caduti su kwrite; dal desktop, dal gestore file & dolphin;[...]
Source	[...]The mouse cursor is identified in the status bar.[...]
Reference	[...]Al puntatore del mouse è identificato nella barra di stato.[...]
Ret. Src.	[...]When you hold the mouse cursor still for a moment[...]
Ret. Trg.	[...]Mantenendo fermo per qualche istante il puntatore del mouse[...]
Generic	[...]Il cursore del mouse viene identificato nella barra di stato.[...]
Instance-based	[...]Il puntatore del mouse viene identificato nella barra di stato.[...]
Corpus-based	[...]Il cursore del mouse è identificato nella barra di stato.[...]
Source	Exiting the kde Control Center
Reference	Uscire dal centro di controllo di kde
Ret. Src.	The kde Control Center Screen
ret. Trg.	Lo schermo del centro di controllo di kde;
Generic	Uscita da kde Control Center
Instance-based	Uscita dal centro di controllo di kde
Corpus-based	Uscita dal centro di controllo di kde
Source	This saves the settings and closes the configuration dialog.
Reference	Questo pulsante salva le impostazioni e chiude la finestra di configurazione.
Ret. Src.	This saves the settings without closing the configuration dialog.
Ret. Trg.	Questo pulsante salva le impostazioni senza chiudere la finestra di configurazione.
Generic	~~pulsante~~ Salva le impostazioni e chiude la finestra di configurazione.
Instance-based	Questo pulsante salva le impostazioni e chiude la finestra di configurazione.
Corpus-based	Questo ~~pulsante~~ salva le impostazioni e chiude la finestra di configurazione.
Source	Automatically scan project's packages
Reference	Analizzare automaticamente i pacchetti del progetto
Ret. Src.	View or modify the UML system's packages.
Ret. Trg.	Consente di visualizzare o modificare i package di sistema UML.
Generic	Scansione automatica dei pacchetti del progetto
Instance-based	Scansione automatica dei package di progetto
Corpus-based	Scansiona automaticamente i pacchetti del progetto

Table 9: Translation examples produced by generic and adaptive NMT systems.

ogy in a CAT environment. In Proc. of AMTA'14, Vancouver, BC, Canada, October.

[Bahdanau et al.2015] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proc. of ICLR'15, San Diego, CA, USA, May.

[Chatterjee et al.2017] Chatterjee, Rajen, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frederic Blain. 2017. Guiding neural machine translation decoding with external knowledge. In Proc. of WMT'17, pages 157–168, Copenhagen, Denmark, September.

[Chen and Cherry2014] Chen, Boxing and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In Proc. of WMT'14, pages 362–367, Baltimore, Maryland, USA, June.

[Chen et al.2016] Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided Alignment Training for Topic-Aware Neural Machine Translation. In Proc. of AMTA'16, pages 121–134, Austin, Texas, October.

[Chu et al.2017] Chu, Chenhui, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. In Proc. of ACL'17-Short Papers, pages 385–391, Vancouver, Canada, July, August.

[Farajian et al.2017a] Farajian, M. Amin, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017a. Neural vs. phrase-based machine translation in a multi-domain scenario. In Proc. of EACL'17, pages 280–284, Valencia, Spain, April.

[Farajian et al.2017b] Farajian, M. Amin, Marco Turchi, Matteo Negri, and Marcello Federico. 2017b. Multi-domain neural machine translation through unsupervised adaptation. In Proc. of WMT'17, pages 127–137, Copenhagen, Denmark, September.

[Freitag and Al-Onaizan2016] Freitag, Markus and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. CoRR. `https://arxiv.org/abs/1612.06897`.

[Goodfellow et al.2016] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press.

[Hildebrand et al.2005] Hildebrand, Almut Silja, Mattias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In Proc. of EAMT'05, pages 133–142, Budapest, Hungary, May.

[Kingscott2002] Kingscott, Geoffrey. 2002. Technical translation and related disciplines. Perspectives, 10(4):247–255.

[Klein et al.2017] Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In Proc. of ACL'17, pages 67–72, Vancouver, Canada, July, August.

[Kobus et al.2017] Kobus, Catherine, Josep Maria Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In Proc. of RANLP'17, pages 372–378, Varna, Bulgaria, September.

[Koehn and Knowles2017] Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proc. of 1^{st} Workshop on Neural Machine Translation, pages 28–39, Vancouver, Canada, August.

[Luong and Manning2015] Luong, Minh-Thang and Christopher D Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In Proc. of IWSLT'15, pages 76–79, Da Nang, Vietnam, December.

[McCandless et al.2010] McCandless, Michael, Erik Hatcher, and Otis Gospodnetic. 2010. Lucene in Action. Manning Publications Co., Greenwich, CT, USA.

[Papineni et al.2002] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proc. of ACL'02, pages 311–318, Philadelphia, USA, July.

[Sennrich et al.2016] Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In Proc. of ACL'16, pages 1715–1725, Berlin, Germany, August.

[Zhang et al.2016] Zhang, Jian, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In Proc. of COLING'16, pages 1807–1817, Osaka, Japan, December.

Translation Quality Estimation for Indian Languages

Nisarg Jhaveri **Manish Gupta*** **Vasudeva Varma**
International Institute of Information Technology,
Gachibowli, Hyderabad, Telangana - 500 032, India.
nisarg.jhaveri@research.iiit.ac.in,
{manish.gupta, vv}@iiit.ac.in

Abstract

Translation Quality Estimation (QE) aims to estimate the quality of an automated machine translation (MT) output without any human intervention or reference translation. With the increasing use of MT systems in various cross-lingual applications, the need and applicability of QE systems is increasing. We study existing approaches and propose multiple neural network approaches for sentence-level QE, with a focus on MT outputs in Indian languages. For this, we also introduce five new datasets for four language pairs: two for English–Gujarati, and one each for English–Hindi, English–Telugu and English–Bengali, which includes one manually post-edited dataset for English–Gujarati. These Indian languages are spoken by around 689M speakers world-wide. We compare results obtained using our proposed models with multiple state-of-the-art systems including the winning system in the WMT17 shared task on QE and show that our proposed neural model which combines the discriminative power of carefully chosen features with Siamese Convolutional Neural Networks (CNNs) works best for all Indian language datasets.

1 Introduction

In recent years, Machine Translation (MT) systems have seen significant improvements. However, the quality of the output obtained from these MT systems is neither perfect nor consistent across multiple test cases. The task of Translation Quality Estimation (QE) aims to estimate the quality of an MT output without any reference translation.

QE is now critically important with the increasing deployment of MT systems in practical environments. QE has been shown to be extremely useful and is widely used in Computer Aided Translation (CAT) environments (Escartín et al., 2017; Turchi et al., 2015). QE can also be useful in various applications and systems such as cross-lingual summarization, cross-lingual information retrieval, etc., which rely on high quality translations. With the help of QE, such systems can automatically pick the best translation out of several proposed translations by multiple MT systems. If the estimated quality is still unsatisfactory the system can alert the user about the poor quality, or fall-back to some alternate way to find a better translation.

Word, phrase, sentence or document level QE has been studied extensively by various researchers. WMT12-17 (the 7^{th} to 10^{th} workshops on statistical machine translation and the 1^{st} and 2^{nd} conferences on machine translation) held a shared task on QE (Callison-Burch et al., 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017). The shared task has explored QE on several datasets and settings for English–Spanish and English–German language pairs over years.

Little work has been done to study QE for Indian languages. In this work, we focus on four Indian languages: Telugu, Hindi, Gujarati and Bengali. According to a 2007 estimate[1], there are 366

*The author is also a Principal Applied Scientist at Microsoft.

[1] https://web.archive.org/web/20071203134724/http://encarta.msn.com/media_701500404/Languages_Spoken_by_More_Than_10_Million_People.html

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 159–168
Alacant, Spain, May 2018.

million Hindi speakers (across five countries), 207 million Bengali speakers (across four countries), 69.7 million Telugu speakers (across four countries), and 46.1 million Gujarati speakers (across eight countries) worldwide, denoting the importance of our choice of these datasets. While English is a West Germanic language that originated from Anglo-Frisian dialects, Hindi, Bengali and Gujarati are Indo-Aryan languages[2], and Telugu is a Dravidian language[3].

Indian languages are relatively free word order languages and morphologically richer when compared to English. Additionally, some Indian languages, for example Telugu, are highly agglutinative. In comparison with English, Hindi has approximately twice as many vowels and consonants. Although Hindi has tenses similar to those used in English, there is a lack of correspondence in their use to express various meanings. Gender and status relations between speakers causes morphological changes in Hindi words, unlike English. Compared to English, Bengali uses onomatopoeia extensively, and so one has to convey that through particular adjectives and adverbs. Besides these differences, there are some phrases, idioms and compound words in English which do not have equivalents in Indian languages due to significant cultural differences.

Because of the differences in the characteristics of the languages involved, existing methods for QE may or may not be effective for all language pairs. We experiment with multiple datasets in different language pairs, each involving English and an Indian language, to study the effectiveness of various models on these datasets.

In addition to the different characteristics of Indian languages, many of these languages are resource-scarce, from a Computational Linguistics perspective. Linguistic resources like dependency parsers or semantic role labelers are not available for most languages we use in this paper. Additionally, large amount of manually annotated data, such as parallel corpora are also difficult and costly to obtain. Hence, in this work, we try to minimize dependency on external large datasets, especially ones which require manual annotation. We hope that the QE accuracy can be further improved using such extra information, and plan to explore it as future work.

To study QE for Indian languages, we also introduce five datasets, for four different language pairs. One dataset, *news.gu*, described in Section 3.2 has been prepared by manually post-editing MT outputs. The other four datasets, described in Section 3.3 make use of existing parallel corpora to create datasets for QE. All datasets are prepared using Neural Machine Translation (NMT) API provided by Google Translate[4]. To the best of our knowledge, we are the first to study QE when using the NMT system.

In this paper, we evaluate the effectiveness of various state-of-the-art systems (proposed for other language pairs) including the winning system of the WMT17 shared task on various Indian language datasets. We also propose and evaluate multiple neural network models for QE. Finally we show that one of our proposed models *CNN.Combined*, described in Section 4.2.2, gives best results on most Indian language datasets. Our major contributions through this paper are as follows.

- Introduction of a manually post-edited QE dataset for English–Gujarati language pair and four other datasets prepared using parallel corpora.

- Proposal of multiple neural network architectures for QE, of which the *CNN.Combined* model is shown to work best for most Indian language datasets in our experiments.

- Evaluation and comparison of several methods of QE on multiple datasets including the WMT17 English–German dataset.

The rest of the paper is organized as follows. We describe related work in Section 2. Section 3 describes the datasets used for the experiments. Section 4 describes different methods and proposed models used for our experiments. Section 5 contains a few notes about the experimental settings. Section 6 provides analysis and related discussions. Finally, we conclude with a brief summary in Section 7.

2 Related Work

Related previous work on translation quality estimation can be organized into two broad kinds of

[2] https://en.wikipedia.org/wiki/Indo-Aryan_languages

[3] https://en.wikipedia.org/wiki/Dravidian_languages

[4] https://translate.google.com/

approaches: manual feature engineering based approaches, and neural networks based approaches. WMT12-17 shared task on QE (Callison-Burch et al., 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017) has recorded the overview and progress of the field over years.

2.1 Manual Feature Engineering based Approaches

Many previous studies on QE were predominantly based on feature engineering. Manual feature engineering can be costly, especially because it needs to be done for each language pair separately.

For Indian languages, few studies have been done, predominantly for English–Hindi language pair. Most of the approaches, most recently Joshi et al. (2016), are based on manual feature engineering, and traditional classification methods. We show in our experiments, that the neural network based models perform significantly better for all language pairs and datasets.

2.2 Neural Network based Approaches

In recent years, many deep learning methods have also been proposed for QE. Patel and Sasikumar (2016) proposed the use of Recurrent Neural Network Language Modeling (RNN-LM) to predict word-level quality labels using bilingual context window proposed by Kreutzer et al. (2015). Several other neural models also use the bilingual context window approach to compose the input layer, which takes the target word and the aligned source word and their contexts as input (Martins et al., 2016, 2017a, 2017b). These models, however, require word alignment information from the MT system or need to align the words using some external parallel corpora. Since our datasets are prepared using neural MT systems, we do not have alignment information from MT system. Additionally, we do not have enough resources to create external word-aligners for each language-pair. As a result, we do not include systems that need word alignment information in our experiments.

Kim and Lee (2016a), Kim and Lee (2016b), Kim et al. (2017a) and Kim et al. (2017b) have studied and proposed different end-to-end neural network based models, primarily based on predictor-estimator architecture. We compare with the architecture described by Kim et al. (2017a) in our experiments. The architecture is explained in Section 4.1.2.

Dataset	Target Language	Train	Dev	Test
wmt17	German (de)	23,000	1,000	2,000
news.gu	Gujarati (gu)	4,489	561	562
ilci.gu	Gujarati (gu)	40,000	5,000	5,000
ilci.hi	Hindi (hi)	40,000	5,000	5,000
ilci.te	Telugu (te)	40,000	5,000	5,000
ilci.bn	Bengali (bn)	40,000	5,000	5,000

Table 1: Target Languages and the Number of Sentence Pairs in each Dataset

Paetzold and Specia (2017) propose a character-level Convolutional Neural Network (CNN) architecture combined with engineered features. The system is comparable to our proposed work in two ways: 1) They do not use any external data or resources. 2) They also use a CNN-based architecture for QE. However, the final architectures are significantly different. Their best system, *SHEF/CNN-C+F*, is explained in Section 4.1.3.

3 Datasets

We used six different datasets for five different language pairs for our experiments. Source language is English for all the datasets. All datasets are split into the typical train, development and test sets. Table 1 shows the target languages and sizes of all the datasets. We describe these datasets in detail in this section.

3.1 WMT17: English-German Dataset

We use the English–German dataset released as part of the WMT17 QE Shared Task (Bojar et al., 2017). The dataset contains text from the Information Technology domain, translated from English to German using a statistical MT system and post-edited by professional translators. The dataset contains source sentences, MT sentences and post-edited sentences, along with Human-targeted Translation Edit Rate (HTER) scores (Snover et al., 2006) for each sentence pair.

Translation Edit Rate (TER) is computed as the minimum number of insertion, deletion, substitution and shift operations needed to be done on MT sentence to match a reference sentence, normalized by the length of the reference sentence. The way the HTER differs from TER is that for HTER, there is no pre-decided reference sentence. There is a human in the loop. The human expert generates the targeted reference by editing the system hypothesis, until it is fluent and has the same meaning as the original source sentence. We use the

HTER scores reported as quality scores for this dataset. The dataset contains 23,000, 1,000 and 2,000 sentences in the training, development and test sets respectively.

3.2 news.gu: English-Gujarati Dataset

We introduce a new QE dataset for the English–Gujarati language pair, prepared using the workbench published by Jhaveri et al. (2018). News articles from various sources were translated to Gujarati from English using the Neural Machine Translation (NMT) API provided by Google Translate and post-edited by one professional translator (different from the authors), who is also a native Gujarati speaker, over a duration of two months. The quality scores, HTER, were computed using the *tercom 0.7.2*[5] tool. The dataset contains a total of 5612 sentences, which was split randomly into training, development and test sets of sizes 4489, 561 and 562 sentences respectively.

3.3 ILCI Parallel Corpora

A parallel corpora for many Indian language pairs, including English has been released by the Indian Languages Corpora Initiative (ILCI)[6] (Choudhary and Jha, 2014). We use the parallel corpora of the health and the tourism domain, having 25,000 sentences for each of the domains for each language pair. We prepare the QE datasets using this for translation from English to four Indian languages, namely, English–Gujarati, English–Hindi, English–Telugu and English–Bengali.

To use the parallel corpora for the QE task, we obtain translations using Google Translate[7] from English to all the target languages. We computed the quality scores as the TER between the MT output and the reference sentences using *tercom 0.7.2*.

The datasets contain a total of 50,000 sentences each, which was divided randomly into training, development and test sets of sizes 40,000, 5000 and 5000 sentences respectively.

4 Models for Translation Quality Estimation

This section describes various models used for experiments and evaluation. We first discuss the baseline models in Section 4.1 and then the proposed models in Section 4.2.

4.1 Baseline Models

In this sub-section, we discuss previously proposed models for QE and their variations. Section 4.1.1 describes baseline model based on Support Vector Regression (SVR). Section 4.1.2 describes *POSTECH.two-step* and *POSTECH.multi-task* models. Section 4.1.3 describes the *SHEF/CNN-C+F* model.

4.1.1 SVR Baseline

The official baseline for WMT17 QE shared task is a Support Vector Regression (SVR) (Drucker et al., 1997) model trained with 17 features for the task. Some of these features use external data such as language models or word alignments trained on large parallel corpora. These features were adapted to use whatever scarce resources are available for our set of target languages as follows. Two features requiring word alignment tables were removed. No external data was used to compute the language models or n-gram counts. Additionally, a few features were added such as, average target token length and depth of parse tree of source sentence. The source parse tree were computed using Stanford CoreNLP toolkit (Manning et al., 2014), this was possible as all the datasets have English as the source language. We call this model *SVR.baseline*.

4.1.2 POSTECH Approaches

POSTECH's participation was the winning system at the WMT17 shared task, which uses a predictor-estimator architecture, many variations of which have been studied and proposed by Kim et al. (2017a), Kim et al. (2017b), Kim and Lee (2016a) and Kim and Lee (2016b). We follow the architecture described by Kim et al. (2017a) for this work.

Kim et al. (2017a) describe a two-step end-to-end neural QE architecture, called predictor-estimator architecture. The predictor-estimator architecture consists of two types of neural network models: 1) word predictor, which is trained on parallel corpora, i.e. using source and reference translations. 2) quality estimator, a neural regressor, trained on QE data.

The first model, word predictor, tries to predict each word in the target sentence using the source sentence and the remaining target sentence as context. They propose an RNN encoder-decoder (Cho et al., 2014; Bahdanau et al., 2014) model based word predictor, which uses bidirectional RNN in

[5] http://www.cs.umd.edu/~snover/tercom/
[6] http://tdil-dc.in
[7] https://translate.google.com/

encoder as well as decoder to use the source sentence information as well as the entire left and right context of target sentence to predict each word.

The estimator part, then, extracts a *quality estimation feature vector* (QEFV) for each word in MT sentence using internal network connections of the word predictor network. For sentence-level QE, the QEFVs are then passed to bidirectional RNN to obtain a summary vector, which, then, is passed to regression layer which generates quality score for sentences.

We define two variations of the model for our experiments: *POSTECH.two-step* and *POSTECH.multi-task*.

POSTECH.two-step trains the two models, word predictor and quality estimator separately as described by Kim et al. (2017a). Input to the word prediction step is source and reference sentences, and the outputs are the predicted words. Whereas, the quality estimator takes source and MT sentence as input and outputs quality score for the sentence. No external parallel corpora have been used for pre-training the word predictor as it is not available for most of the language pairs we work with.

The main idea of POSTECH system proposed by Kim et al. (2017a) is to take advantage of pre-training of word predictor using large external parallel corpora. Since we do not use any external corpora, we propose a variation of this model, which jointly learns both, word predictor and quality estimator, in a multi-task setting. We call this model *POSTECH.multi-task*. The inputs to this model are the source and MT sentence, and the outputs are predicted words and quality score.

Recently, Kim et al. (2017b) proposed single-level and multi-level stack propagation based learning for the two steps. We experimented with single-level stack propagation, as we do not have necessary training data for all sentence, word and phrase level QE, which the multi-level model requires. In our experiments, we did not see any significant improvement across datasets between single-level stack propagation (Kim et al., 2017b) and two-step learning (Kim et al., 2017a).

4.1.3 SHEF/CNN Approach

Paetzold and Specia (2017) propose an architecture that combines engineered features and character-level information using deep Convolutional Neural Networks (CNN) and Multi-Layered Perceptrons (MLP). The model *SHEF/CNN-C+F* has three parts, sentence encoders for source and MT sentence, MLP for engineered features and a final layer to combine both and generate quality scores.

The sentence encoder takes the sequence of characters as input, and converts it to a sequence of character embeddings. They stack four pairs of convolution and max-pooling for each window size. Each stack is applied to character embeddings in parallel, and later flattened and concatenated to get a sentence vector. Two different encoders, each for source and MT sentences are created. The encoded source and MT sentence are then concatenated with the encoded features, which are obtained by applying MLP on engineered features. A final layer is applied on the concatenated vectors, which predicts the quality scores.

4.2 Proposed Models

In this section, we discuss our proposed neural architectures for QE. Section 4.2.1 describes two proposed RNN-based models: *RNN* and *RNN.summary-attention*. Section 4.2.2 describes the proposed CNN-based models: *CNN.Siamese*, *CNN.Combined*, and *CNN.Combined.no-features*,

4.2.1 Recurrent Neural Network (RNN) Approaches

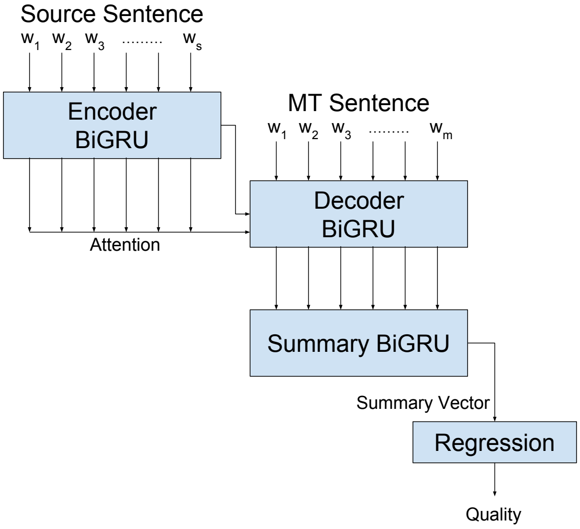


Figure 1: Architecture of the *RNN* model

The POSTECH architecture, described in Section 4.1.2, takes advantage of the pre-training of word predictor on large external parallel corpora. Since no such datasets are easily available for most language pairs in our case, we propose a simplified version of POSTECH removing the word prediction step, and simplifying the QEFV extraction.

The model takes source sentence and MT sentence as input. A bidirectional RNN encoder, is applied on the source sentence, which gives a fixed size representation, which in turn is used as the initial state for decoder. Decoder is also a bidirectional RNN, with attention over the encoder outputs for each word and predicts a QEFV for each word in MT sentence. The outputs of decoder, QEFVs, are then "summarised" by another bidirectional RNN, to generate a summary vector for the sentence pair. This summary vector is then passed to a regression layer, which outputs the predicted quality score. The predicted quality score is compared with the actual quality scores under the L2 loss function for training the network using back-propagation. Figure 1 shows the architecture of the *RNN* model.

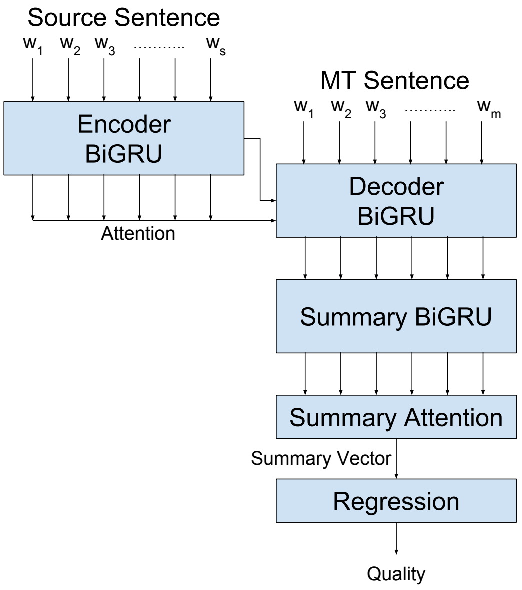


Figure 2: Architecture of the *RNN.summary-attention* model

We also propose a variation of this model, called *RNN.summary-attention*, in which the summary vectors are created using attention mechanism over bidirectional RNN outputs. The QEFVs obtained from decoder are passed to a bidirectional RNN, the outputs of which are then passed to a word attention mechanism, similar to Yang et al. (2016), to get a fixed length summary vector. Attention allows the model to give more importance to certain words in the context while ignoring the others, effectively learning the focus points to better predict the quality score. Figure 2 shows the architecture of the *RNN.summary-attention* model.

4.2.2 Convolutional Neural Network (CNN) Approaches

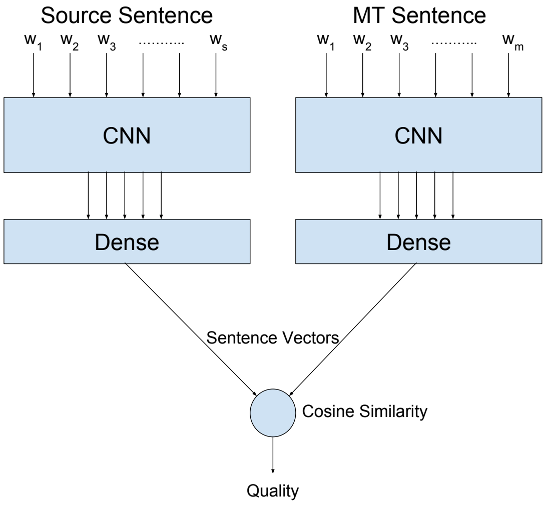


Figure 3: Architecture of the *CNN.Siamese* model

In the basic CNN model, we encode both the source and MT sentence, using CNN-based sentence encoders, similar to one proposed by Kim (2014) for the text classification task. The encoder takes a sentence as a list of word embeddings and applies multiple convolution filters with varying window sizes and applies max-over-time pooling (Collobert et al., 2011) operations for each filter, output of which is then passed to a dense layer, to obtain a sentence vector.

We create two independent encoders (weights are not shared), each for source and target language sentences. The source and MT sentences are encoded using encoder for their respective languages. Finally we take cosine similarity of the two encoded sentence vectors to obtain the quality score. We call this model *CNN.Siamese*. Figure 3 shows the architecture of this model.

We also propose an extension of *CNN.Siamese* model in which the model computes the quality scores in two different ways using the same encoded sentences. One path computes the cosine similarity between the two encoded sentences. The other path concatenates the sentence encodings, optionally along with feature embeddings, and applies a fully connected layer to produce quality scores, similar to *SHEF/CNN-C+F* model described in Section 4.1.3. The final quality score is computed by averaging the two quality scores given by different paths. The architecture of this model is shown in Figure 4. We include two variations, with and without engineered features in our experiments, called *CNN.Combined* and *CNN.Combined.no-features* respectively.

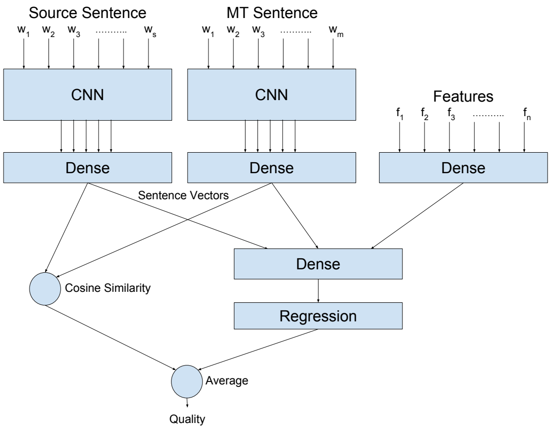


Figure 4: Architecture of the *CNN.Combined* model

For each CNN based model, we tried two initializations for word embeddings: 1) Random 2) Using the pre-trained models published by FastText[8] (Bojanowski et al., 2016), which are trained on Wikipedia[9] for corresponding languages. The experiments, which use the FastText embeddings are denoted by *+fastText* suffix.

5 Experimental Settings

The code used for experiments has been made publicly available at `https://goo.gl/gG9J6f`.

SVR.baseline model is trained using scikit-learn library (Pedregosa et al., 2011). Keras (Chollet and others, 2015), with Theano (Theano Development Team, 2016) is used to implement all the neural network models, including the baselines.

Development set was used for parameter tuning for *SVR.baseline* for each dataset. For neural models, development data was used as validation data while training models, to early stop the training to prevent overfitting.

GRU cells (Cho et al., 2014), with 500 hidden units, are used in RNNs in all the neural network models. Sentences are clipped to length of 100 words and padded with masking. Vocabulary size is limited to 40,000 words for all the experiments. Word embedding size is set to 300.

For all proposed CNN based models, 200 filters of sizes 3, 4 and 5 each were used in the sentence encoders. Sentence vector size was set to 500.

6 Evaluation and Results

Two types of evaluation are performed for all experiments: 1) Using Pearson's correlation coefficient between the predicted quality scores and the actual quality scores, to evaluate scoring. 2) Using Spearman's correlation coefficient to evaluate the ranking of sentences according to quality.

We also report statistical significance of the results considering *POSTECH.two-step* as baseline, over ten different runs.

Table 2 shows comparison of different models for the scoring task using Pearson's correlation. Table 3 shows comparison of different models for the ranking task using Spearman's correlation.

We find that *POSTECH.two-step* model works best for *WMT17* en–de dataset for both the tasks, but fails to give best results for any other dataset, in the low-resource settings explored in this paper. We also find that the proposed CNN-based models generally work better for Indian language datasets. The better performance of CNN-based models over RNN-based models for Indian languages might be because of the free word order property of Indian languages. CNN does not directly rely on entire sequence and order of words, rather it picks best phrases depending on filter sizes from the sentence without explicitly looking at the order.

Our final model *CNN.Combined*, with or without the use of FastText embeddings works best for four out of five Indian language datasets for the scoring task. For *news.gu* dataset, our combined CNN model, without engineered features, *CNN.Combined.no-features+fastText*, gives the best results. On investigating the relatively low results of the two variants of *CNN.Combined* model on *news.gu*, we found that due to some engineered features and the relatively small size of train set, the combined CNN model with features was rapidly overfitting. The similar model without engineered features, *CNN.Combined.no-features* works as expected and yields the best results on *news.gu*.

Similarly, for the ranking task, the two variants of *CNN.Combined* model outperform all the models for four out five datasets. For the remaining Indian language dataset, *news.gu*, *CNN.Siamese+fastText* model yields the best result.

We also notice that using *fastText* embeddings in CNN based models generally works better com-

[8] `https://fasttext.cc/docs/en/pretrained-vectors.html`

[9] `https://www.wikipedia.org/`

Model	wmt17	news.gu	ilci.gu	ilci.hi	ilci.te	ilci.bn
SVR.baseline (original features)	39.98	-	-	-	-	-
SVR.baseline	38.26	20.12	44.67	39.58	44.20	33.65
POSTECH.multi-task	42.44	38.85	45.63	46.51	45.21	38.66
POSTECH.two-step	**50.40**	30.14	49.47	50.23	46.18	44.43
SHEF/CNN-C+F (original features)	40.34†	-	-	-	-	-
SHEF/CNN-C+F	34.22†	29.05	44.32†	39.73†	46.60	34.93†
RNN	41.71†	37.74*	48.56	50.58	49.07*	45.14*
RNN.summary-attention	39.68†	37.30*	48.85	52.59*	49.42*	44.85
CNN.Siamese	44.22†	43.75*	49.29	52.71*	49.56*	44.83
CNN.Siamese+fastText	47.39†	48.60*	51.85*	53.06*	49.69*	45.40
CNN.Combined.no-features	45.83†	43.43*	48.88	52.01*	49.31*	44.68
CNN.Combined.no-features+fastText	48.14†	**49.06***	52.12*	53.17*	49.35*	45.00
CNN.Combined	46.98†	41.51*	52.46*	53.00*	**51.14***	**46.62***
CNN.Combined+fastText	48.96†	46.11*	**52.71***	**53.51***	50.06*	46.08*

Table 2: Results for the Scoring Task, Pearson's Correlation ($*$ and † indicate statistically significantly better or worse ($p < 0.05$) compared to *POSTECH.two-step* respectively)

Model	wmt17	news.gu	ilci.gu	ilci.hi	ilci.te	ilci.bn
SVR.baseline (original features)	43.16	-	-	-	-	-
SVR.baseline	40.65	7.06	42.15	38.44	41.20	31.62
POSTECH.multi-task	44.52	22.46	43.15	44.43	42.03	35.69
POSTECH.two-step	**52.06**	19.61	46.85	48.23	42.83	40.94
SHEF/CNN-C+F (original features)	43.37†	-	-	-	-	-
SHEF/CNN-C+F	37.98†	14.89†	42.97†	39.09†	44.39*	32.61†
RNN	43.42†	27.42*	46.00	48.77	46.33*	42.11*
RNN.summary-attention	41.74†	23.21	46.07	50.48*	46.34*	41.90*
CNN.Siamese	46.20†	31.98*	46.48	51.16*	46.05*	41.43
CNN.Siamese+fastText	49.49†	**41.87***	48.34*	51.67*	45.13*	41.27
CNN.Combined.no-features	47.90†	29.81*	46.03	50.37*	45.77*	41.23
CNN.Combined.no-features+fastText	50.10†	41.13*	49.08*	51.78*	45.13*	40.88
CNN.Combined	48.79†	30.70*	**50.21***	51.32*	**47.58***	**44.19***
CNN.Combined+fastText	51.06	38.20*	49.77*	**52.28***	45.90*	42.39*

Table 3: Results for the Ranking Task, Spearman's Correlation ($*$ and † indicate statistically significantly better or worse ($p < 0.05$) compared to *POSTECH.two-step* respectively)

pared to using random embeddings. However, in some cases, especially for Telugu and Bengali datasets, random initialization of embeddings performs better.

Our word-level CNN encoder based Siamese architecture, *CNN.Siamese* model outperforms the *SHEF/CNN-C+F* model, which is a character based deep CNN model, combined with engineered features. We also show that combining the Siamese architecture with MLP based architecture in *SHEF/CNN-C+F*, *CNN.Combined* model, further improves the results.

The RNN based models work comparably or better for all Indian language datasets, but are much simpler and have much lower number of trainable parameters compared to *POSTECH* models. However, the difference between the two RNN based models, *RNN* and *RNN.summary-attention*, across datasets is inconclusive.

In Table 4, we show some examples of scores predicted by our proposed system *CNN.Combined+fastText* and the baseline (*POSTECH.two-step*) system, along with source, MT and reference sentences and actual quality scores. Note that across examples with low to high quality scores, our method can accurately predict the quality score much better than the baseline.

Dataset	Source sentence	MT sentence	Correct sentence	Base-line	Our model	Actual TER
news.gu	Every year , loud sound from firecrackers causes stress , terror and even death in strays and birds .	દર વર્ષે , ફટાકડાથી ઘોંઘાટવાળા અવાજ તણાવ , આતંક અને ભટકતા અને પક્ષીઓમાં મૃત્યુ પણ થાય છે .	દર વર્ષે , ફટાકડાથી સર્જાતો ઘોંઘાટ પ્રાણીઓ અને પક્ષીઓમાં તણાવ , આતંક અને મૃત્યુ પણ સર્જે છે .	0.03	0.31	0.33
ilci.gu	The total distance of this route is 163 kilometers from Pathankot to Jogindernagar .	આ માર્ગની કુલ અંતર પઠાણકોટથી જોગીન્દ્રનગરથી 163 કિ.મી . છે .	પઠાનકોટથી જોગિન્દરનગર સુધીના આ રૂટનું કુલ અંતર ૧૬૩ કિલોમીટર છે .	0.31	0.75	0.73
ilci.hi	The tombs of Shahjahan and Mumtaz are surrounded by fine meshes .	शाहजहां और मुमताज की मकबरे परिश्रम से घिरे हैं ।	शाहजहाँ और मुमताज के मकबरे चारों तरफ से महीन जालियों से घिरे हैं ।	0.89	0.53	0.50
ilci.te	People of Hindustan , Pakistan , Bangladesh , Egypt do business in Manama Souk .	హిందూస్తాన్ , పాకిస్థాన్ , బంగ్లాదేశ్ , మనామ సౌక్ లోని ఈజిప్టు ప్రజలు .	భారతదేశం , పాకిస్థాన్ , బాంగ్లాదేశ్ , మిశ్ర్ ప్రజలు మానామా సూక్లో వ్యాపారం చేస్తారు .	0.98	0.62	0.62
ilci.bn	There are eight - ten houses of wood in Gejam village .	গাজাম গ্রামের আটটি কাঠের কাঠামো রয়েছে।	গেজম বসতিতে আট - দশটি কাঠের বাড়ি আছে।	0.58	0.85	0.89

Table 4: Example of output by baseline (*POSTECH.two-step*), compared with our proposed model (*CNN.Combined+fastText*), across all datasets.

7 Conclusions

In this paper, we study the effectiveness of different neural network architectures for QE for Indian languages. We also introduce multiple datasets for the task, which can be used as benchmark for future work in the area. We observe that our proposed *CNN.Combined* model beats the state-of-the-art methods by a significant margin.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural mt by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on SMT. In *Proc. of the Eighth Workshop on SMT*, pages 1–44, Aug.

Bojar, Ondřej, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on smt. In *Proc. of the Ninth Workshop on SMT*, pages 12–58, Jun.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on smt. In *Proc. of the Tenth Workshop on SMT*, pages 1–46, Sep.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conf. on mt. In *Proc. of the First Conf. on MT*, pages 131–198, Aug.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conf. on mt (wmt17). In *Proc. of the Second Conf. on MT, Volume 2: Shared Task Papers*, pages 169–214, Sep.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on smt. In *Proc. of the Seventh Workshop on SMT*, pages 10–51, Jun.

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for smt. In *EMNLP*, pages 1724–1734.

Chollet, François et al. 2015. Keras. `https://keras.io`.

Choudhary, Narayan and Girish Nath Jha. 2014. Creating multilingual parallel corpora in indian languages. In *Human Language Technology Challenges for Computer Science and Linguistics*, pages 527–537.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537.

Drucker, Harris, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *NIPS*, pages 155–161.

Escartín, Carla Parra, Hanna Béchara, and Constantin Orăsan. 2017. Questing for quality estimation a user study. *The Prague Bulletin of Mathematical Linguistics*, 108(1):343–354.

Jhaveri, Nisarg, Manish Gupta, and Vasudeva Varma. 2018. A workbench for rapid generation of cross-lingual summaries. In *LREC*, page to appear.

Joshi, Nisheeth, Iti Mathur, Hemant Darbari, and Ajai Kumar. 2016. Quality estimation of english-hindi mt systems. In *Proc. of the Second Intl. Conf. on Information and Communication Technology for Competitive Strategies*, page 53.

Kim, Hyun and Jong-Hyeok Lee. 2016a. Recurrent neural network based translation quality estimation. In *Proc. of the First Conf. on MT*, pages 787–792, Aug.

Kim, Hyun and Jong-Hyeok Lee. 2016b. A recurrent neural networks approach for estimating the quality of mt output. In *NAACL-HLT*, pages 494–498.

Kim, Hyun, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-estimator: Neural quality estimation based on target word prediction for mt. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):3.

Kim, Hyun, Jong-Hyeok Lee, and Seung-Hoon Na. 2017b. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proc. of the Second Conf. on MT, Volume 2: Shared Task Papers*, pages 562–568, Sep.

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Kreutzer, Julia, Shigehiko Schamoni, and Stefan Riezler. 2015. Quality estimation from scratch (quetch): Deep learning for word-level translation quality estimation. In *Proc. of the Tenth Workshop on SMT*, pages 316–322.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, pages 55–60.

Martins, André F. T., Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the wmt16 word-level translation quality estimation shared task. In *Proc. of the First Conf. on MT*, pages 806–811, August.

Martins, André F. T., Fabio Kepler, and Jose Monteiro. 2017a. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proc. of the Second Conf. on MT, Volume 2: Shared Task Papers*, pages 569–574, Sep.

Martins, André FT, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017b. Pushing the limits of translation quality estimation. *TACL*, 5:205–218.

Paetzold, Gustavo and Lucia Specia. 2017. Feature-enriched character-level convolutions for text regression. In *Proc. of the Second Conf. on MT, Volume 2: Shared Task Papers*, pages 575–581, Sep.

Patel, Raj Nath and M Sasikumar. 2016. Translation quality estimation using recurrent neural network. In *Proc. of the First Conf. on MT*, pages 819–824, Aug.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *JMLR*, 12:2825–2830.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of association for MT in the Americas*, volume 200.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

Turchi, Marco, Matteo Negri, and Marcello Federico. 2015. Mt quality estimation for computer-assisted translation: Does it really help? In *Proc. of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 2: Short Papers)*, volume 2, pages 530–535.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489.

A Reinforcement Learning Approach to Interactive-Predictive Neural Machine Translation

Tsz Kin Lam[†,*] and **Julia Kreutzer**[*] and **Stefan Riezler**[†,*]
[*]Computational Linguistics & [†]IWR, Heidelberg University, Germany
{lam,kreutzer,riezler}@cl.uni-heidelberg.de

Abstract

We present an approach to interactive-predictive neural machine translation that attempts to reduce human effort from three directions: Firstly, instead of requiring humans to select, correct, or delete segments, we employ the idea of learning from human reinforcements in form of judgments on the quality of partial translations. Secondly, human effort is further reduced by using the entropy of word predictions as uncertainty criterion to trigger feedback requests. Lastly, online updates of the model parameters after every interaction allow the model to adapt quickly. We show in simulation experiments that reward signals on partial translations significantly improve character F-score and BLEU compared to feedback on full translations only, while human effort can be reduced to an average number of 5 feedback requests for every input.

1 Introduction

Interactive-predictive machine translation aims at obtaining high-quality machine translation by involving humans in a loop of user validations of partial translations suggested by the machine translation system. This interaction protocol can easily be fit to neural machine translation (NMT) (Bahdanau et al., 2015) by conditioning the model's word predictions on the user-validated prefix (Knowles and Koehn, 2016; Wuebker et al., 2016). User studies conducted by Green et al. (2014) for phrase-based machine translation have shown that the interactive-predictive interaction protocol leads to significant reductions in post-editing effort. Other user studies on interactive machine translation based on post-editing have shown that human effort can also be reduced by improving the online adaptation capabilities of the learning system, both for statistical phrase-based (Bentivogli et al., 2016) or NMT systems (Karimova et al., 2017).

The goal of our work is to further reduce human effort in interactive-predictive NMT by combining the advantages of the interactive-predictive protocol with the advantages of learning from weak feedback. For the latter we rely on techniques from reinforcement learning (Sutton and Barto, 2017), a.k.a. bandit structured prediction (Sokolov et al., 2016; Kreutzer et al., 2017; Nguyen et al., 2017) in the context of sequence-to-sequence learning. Our approach attacks the problem of reducing human effort from three innovative directions.

- Firstly, instead of requiring humans to correct or delete segments proposed by the machine translation system, we employ the reinforcement learning idea of humans providing reward signals in form of judgments on the quality of the machine translation. Human effort is reduced since each partial translation receives a human reward signal at most once, rendering it a bandit-type feedback signal, and each reward signal itself is easier to obtain than a correction of a translation.

- In order to reduce the amount of feedback signals even further, we integrate an uncertainty criterion for word predictions to trigger requests for human feedback. Using the comparison of the current average entropy to the entropy of word predictions in the history as a measure for uncertainty, we reduce the amount of feedbacks requested from humans to an average number of 5 requests per input.

- In contrast to previous approaches to interactive-predictive translation, the parameters of our translation system are updated online after receiving feedback for partial translations. The update is done according to an actor-critic reinforcement learning protocol where each update pushes up the score function of the partial translation sampled by the model (called actor) proportional to a learned reward function (called critic). Furthermore, since the entropy criterion is based on the actor, it is also automatically updated. Frequent updates improve the adaptability of our system, resulting in a further reduction of human effort.

The rest of this paper is structured as follows. In Section 2, we will situate our approach in the context of interactive machine translation and analyze our contribution related to reinforcement learning for sequence prediction problems. Details of our algorithm are given in Section 3. We evaluate our approach in a simulation study where bandit feedback is computed by evaluating partial translations against references under a character F-score metric (Popović, 2015) without revealing the reference translation to the learning system (Section 4). We show that segment-wise reward signals improve translation quality over reinforcement learning with sparse sentence-wise rewards, measured by character F-score and corpus-based BLEU against references. Furthermore, we show that human effort, measured by the number of feedback requests, can be reduced to an average number of 5 requests per input. These implications of our new paradigm are discussed in Section 5.

2 Related Work

The interactive-predictive translation paradigm reaches back to early approaches for IBM-type (Foster et al., 1997; **?**) and phrase-based machine translation (Barrachina et al., 2008; Green et al., 2014). Knowles and Koehn (2016) and Wuebker et al. (2016) presented *neural interactive translation prediction* — a translation scenario where translators interact with an NMT system by accepting or correcting subsequent target tokens suggested by the NMT system in an auto-complete style. NMT is naturally suited for this incremental production of outputs, since it models the probability of target tokens given a history of target tokens sequentially from left to right. In standard supervised training with teacher forcing, this history comes from the ground truth, while in interactive-predictive translation it is provided by the prefix accepted or entered by the user. Both approaches use references to simulate an interaction with a translator and compare their approach to phrase-based prefix-search. They find that NMT is more accurate in word and letter prediction and recovers better from failures. Similar to their work, we will experiment in a simulated environment with references mimicking the translator. However, we do not use the reference directly for teacher forcing, but only to derive weak feedback from it. Furthermore, our approach employs techniques to reduce the number of interactions, and to update the model more frequently than after each sentence.

Our work is also closely related to approaches for *interactive pre-post-editing* (Marie and Max, 2015; Domingo et al., 2018). The core idea is to ask the translator to mark good segments and use these for a more informed re-decoding. Both studies could show a reduction in human effort for post-editing in simulation experiments. We share the goal of using human feedback more effectively by targeting it towards essential translation segments, however, our approach does adhere to the left-to-right navigation through translation hypotheses. In difference to these approaches, we try to reduce human effort even further by minimizing the number of feedback requests and by frequent model updates.

Reinforcing/penalizing a targeted set of actions can also be found in recent approaches to *reinforcement learning from human feedback*. For example, Judah et al. (2010) presented a scenario where users interactively label freely chosen good and bad parts of a policy's trajectory. The policy is directly trained with this reinforcement signal to play a real-time strategy game. Simulations of NMT systems interacting with human feedback have been presented firstly by Kreutzer et al. (2017), Nguyen (2017), or Bahdanau et al. (2017) who apply different policy gradient algorithms, William's REINFORCE (Williams, 1992) or actor-critic methods (Konda and Tsitsiklis, 2000; Sutton et al., 2000; Mnih et al., 2016), respectively. While

Bahdanau et al.'s (2017) approach operates in a fully supervised learning scenario, where rewards are simulated in comparison to references with smoothed and length-rescaled BLEU, Kreutzer et al. (2017) and Nguyen et al. (2017) limit the setup to sentence-level bandit feedback, i.e. only one feedback is obtained for one completed translation per input. In this paper, we use actor-critic update strategies, but we receive simulated bandit feedback on the sub-sentence level.

We adopt techniques from *active learning* to reduce the number of feedbacks requested from a user. González-Rubio et al. (2011; 2012) apply active learning for interactive machine translation, where a user interactively finishes the translation of an SMT system. The active learning component decides which sentences to sample for translation (i.e. receive full supervision for) and the SMT system is updated online (Ortiz-Martínez et al., 2010). In our algorithm the active learning component decides which prefixes to be rated (i.e. receive weak feedback for) based on their average entropy. Entropy is a popular measure for uncertainty in active learning: the rationale is to feed the learning algorithm with labeled instances where it is least confident about its own predictions. This *uncertainty sampling* algorithm (Lewis and Gale, 1994) is a popular choice for active learning for NLP tasks with expensive gold labeling, such as text classification (Lewis and Gale, 1994), word-sense disambiguation (Chen et al., 2006) and statistical parsing (Tang et al., 2002). Our method falls into the category of stream-based online active learning (as opposed to pool-based active learning, selecting instances from a large pool of unlabeled data), since the algorithm decides on the fly (online) which translation prefixes of the stream of source tokens to request feedback for. Instead of receiving gold annotations, as in the studies mentioned above, our algorithm receives weaker, bandit feedback — but the motivation of minimizing human labeling effort is the same.

3 Reinforcement Learning for Interactive-Predictive Translation

In the following, we will introduce the key ideas of our approach, formalize them, and present an algorithm for reinforcement learning for interactive-predictive NMT.

3.1 Actor-Critic Reinforcement Learning for NMT

The objective of reinforcement learning methods is to maximize the expected reward obtainable from interactions of an agent (here: a machine translation system) with an environment (here: a human translator). In our case, the agent/system performs actions by predicting target words y_t according to a stochastic policy p_θ parameterized by an RNN encoder-decoder NMT system (Bahdanau et al., 2015) where

$$p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T_y} p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t}). \qquad (1)$$

The environment/human can be formalized as a Markov Decision Process where a state at time t is a tuple $s_t = \langle \mathbf{x}, \mathbf{y}_{<t} \rangle$ consisting of the conditioning context of the input $\mathbf{x}$ and the current produced history of target tokens $\mathbf{y}_{<t}$. Note that since states s_{t+1} include the current chosen action y_t and can contain long histories $\mathbf{y}_{<t}$, the state distribution is sparse and deterministic. The reward distribution of the environment/critic is estimated by function approximation in actor-critic methods. The reward estimator (called critic) is trained on actual rewards and updated after every interaction, and then used to update the parameters of the policy (called actor) in a direction of function improvement. We use the advantage actor critic framework of Mnih et al. (2016) which estimates the advantage $A_\phi(y_t|s_t)$ in reward of choosing action y_t in a given state s_t over the mean reward value for that state. This framework has been applied to reinforcement learning for NMT by Nguyen et al. (2017). The main objective of the actor is then to maximize the expected advantage

$$L_\theta = \mathbb{E}_{p(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})} \left[\sum_{t=1}^{T_y} A_\phi(y_t|s_t) \right]. \qquad (2)$$

The stochastic gradient of this objective for a sampled target word $\hat{y}_t$ for an input $\mathbf{x}$ can be calculated following the policy gradient theorem (Sutton et al., 2000; Konda and Tsitsiklis, 2000) as

$$\nabla L_\theta(\hat{y}_t) = \sum_{t=1}^{T_y} \left[\nabla \log p_\theta(\hat{y}_t|s_t) A_\phi(\hat{y}_t|s_t) \right]. \qquad (3)$$

In standard actor-critic algorithms, the parameters of actor and the critic are updated online at each time step. The actor parameters θ are updated by sampling $\hat{y}_t$ from p_θ and performing a step in the opposite direction of the stochastic gradient of $L_\theta(\hat{y}_t)$; the critic parameters ϕ are updated by minimizing $L_\phi(\hat{y}_t)$, defined as the mean squared error of the reward estimator for sampled target word $\hat{y}_t$ with respect to actual rewards (for more details see Nguyen et al. (2017)). In our experiments, we simulate user rewards by character F-score (chrF) values of partial translations.

3.2 Triggering Human Feedback Requests by Actor Entropy

Besides the idea of replacing human post-edits by human rewards, another key feature of our approach is to minimize the number of requests for human feedback. This is achieved by computing the uncertainty of the policy distribution as the average word-level entropy $\bar{H}$ of an n-word partial translation, defined as

$$\bar{H}(\hat{y}_{1:n}) = \frac{1}{n}\sum_{t=1}^{n}\left[-\sum_{v\in\mathcal{V}} p_\theta(v|s_t)\log p_\theta(v|s_t)\right], \quad (4)$$

where $\hat{y}_{1:n} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ is a sequence of n predicted tokens starting at the sentence beginning, $\mathcal{V}$ is the output vocabulary, and $p_\theta(v|s_t)$ is the probability of predicting a word in $\mathcal{V}$ at state s_t of the RNN decoder.

A request for human feedback is triggered when $\bar{H}(\hat{y}_{1:n})$ is higher than a running average γ by a factor of ϵ or when <eos> is generated. Upon receiving a reward from the user, both actor and critic are updated. Hence, our algorithm takes the middle ground between updating at each time step t and performing an update only after a reward signal for the completed translation is received. In our simulation experiments, this process is repeated until the <eos> token is generated, or when a pre-defined maximum length, here $T_{\max} = 50$, is reached.

3.3 Simulating Human Rewards on Translation Quality

Previous work on reinforcement learning in machine translation has simulated human bandit feedback by evaluating full-sentence translations against references using per-sentence approximations of BLEU (Sokolov et al., 2016; Kreutzer et al., 2017; Nguyen et al., 2017). We found that when working with partial translations, user feedback on translation quality can successfully be simulated by computing the chrF-score (Popović, 2015) of the translation with respect to the reference translation truncated to the same length. If the length of the translation exceeds the length of the reference, no truncation is used. We denote rewards as a function $\mathrm{R}(\hat{y}_{1:t})$ of only the partial translation $\hat{y}_{1:t}$, in order to highlight the fact that rewards are in principle independent of reference translations.

3.4 Sampling versus Forced Decoding via Prefix Buffer Ξ

The standard approach to estimate the expected reward in policy gradient techniques is to employ Monte-Carlo methods, in specific, multinomial sampling of actions. This guarantees an unbiased estimator and allows sufficient exploration of the action space in learning. In contrast, interactive-predictive machine translation usually avoids exploration in favor of exploitation by decoding the best partial translation under the current model after every interaction. Since in our framework, learning and decoding are interleaved, we have to find the best compromise between exploration and exploitation.

The general modus operandi of our framework is simultaneous exploration and exploitation by multinomial sampling actions from the current policy. However, in cases where a partial translation receives a high user reward, we store it in a so-called prefix buffer Ξ, and perform forced decoding by feeding the prefix to the decoder for the remaining translation process.

3.5 Algorithm for Bandit Interactive-Predictive NMT

Algorithm 1 gives pseudo-code for **B**andit-**I**nteractive-**P**redictive **N**eural **M**achine **T**ranslation (BIP-NMT). The algorithm receives an input source sequence $\mathbf{x_i}$ (line 4), and incrementally predicts a sequence of output target tokens up to length $T_{\max}$ (line 6). At each step t, a partial translation $\hat{y}_{1:t}$ is sampled from the policy distribution $p_\theta(\cdot|\mathbf{x_i}, \mathbf{y}_{<t}, \Xi)$ that implements an RNN encoder-decoder with an additional prefix buffer Ξ for forced decoding (line 7). User feedback is requested in case the average entropy $\bar{H}(\hat{y}_{1:t})$ of the policy is larger than or equal to a running average by a factor of ϵ or when <eos> is generated (line 9). If the reward $\mathrm{R}(\hat{y}_{1:t})$ is larger than or equal to a threshold μ, the prefix is stored in a buffer for forced decoding (lines 11-12). Next,

Algorithm 1: Algorithm BIP-NMT

```
1:  **Input:** $\theta_0$, $\phi_0$, $\alpha_A$, $\alpha_C$
2:  **Output:** Estimates $\theta^*$, $\phi^*$
3:  **for** i = 1, ... N **do**
4:      Receive $\mathbf{x_i}$
5:      Initialize $\gamma \leftarrow 0$, $\Xi \leftarrow \emptyset$
6:      **for** t = 1 ... $T_{\max}$ **do**
7:          Sample $\hat{y}_{1:t} \sim p_{\theta_{t-1}}(\cdot|\mathbf{x_i}, \mathbf{y}_{<t}, \Xi)$
8:          Compute $\bar{H}(\hat{y}_{1:t})$ using Eq. (4)
9:          **if** $\bar{H}(\hat{y}_{1:t}) - \gamma_{t-1} \geq \epsilon \times \gamma_{t-1}$ or <eos>
            in $\hat{y}_{1:t}$ **then**
10:             Receive feedback R$(\hat{y}_{1:t})$
11:             **if** R$(\hat{y}_{1:t}) \geq \mu$ **then**
12:                 $\Xi \leftarrow \hat{y}_{1:t}$
13:             **end if**
14:             Update $\theta_t \leftarrow \theta_{t-1} - \alpha_A \nabla L_{\theta_{t-1}}(\hat{y}_t)$
                (Eq. (3))
15:             Update $\phi_t \leftarrow \phi_{t-1} - \alpha_C \nabla L_{\phi_{t-1}}(\hat{y}_t)$
                (see Eq. (7) in Nguyen et al. (2017))
16:         **end if**
17:         Update $\gamma_t = \gamma_{t-1} + \frac{1}{t}\left(\bar{H}(\hat{y}_{1:t}) - \gamma_{t-1}\right)$
18:         **break** if <eos> in $\hat{y}_{1:t}$
19:     **end for**
20: **end for**
```

updates of the parameters of the policy (line 14), critic (line 15), and average entropy (line 17) are performed. Actor and critic each use a separate learning rate schedule (α_A and α_C).

Figure 1 visualizes the interaction of the BIP-NMT system with a human for a single translation: Feedback is requested when the model is uncertain or the translation is completed. It is directly used for a model update and, in case it was good, for filling the prefix buffer, before the model moves to generating the next (longer) partial translation.

4 Experiments

We simulate a scenario where the learning NMT system requests online bandit feedback for partial translations from a human in the loop. The following experiments will give an initial practical assessment of our proposed interactive learning algorithm. Our analysis of the interactions between actor, critic and simulated human will provide further insights into the learning behavior of BIP-NMT.

4.1 Setup

Data and Preprocessing. We conduct experiments on French-to-English translation on Europarl (EP) and News Commentary (NC) domains. The large EP parallel corpus is used to pre-train the actor in a fully-supervised setting with a standard maximum likelihood estimation objective. The critic network is not pre-trained. For interactive training with bandit feedback, we extract 10k sentences from the NC corpus. Validation and test sets are also chosen from the NC domain. Note that in principle more sentences could be used, however, we would like to simulate a realistic scenario where human feedback is costly to obtain. Data sets were tokenized and cleaned using Moses tools (Koehn et al., 2007). Furthermore, sentences longer than 50 tokens were removed from the training data. Each language's vocabulary contains the 50K most frequent tokens extracted from the two training sets. Table 1 summarizes the data statistics.

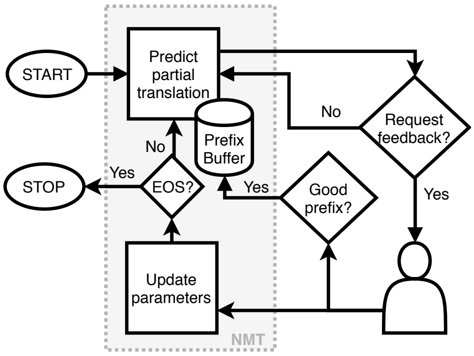


Figure 1: Interaction of the NMT system with the human during learning for a single translation.

Dataset	EP (v.5)	$\bar{n}$	NC (WMT07)	$\bar{n}$
Training (filt.)	1,346,679	23.5	9,216	21.9
Validation	2,000	29.4	1,064	24.1
Test	-	-	2,007	24.8

Table 1: Number of parallel sentences and average number of words per sentence in target language (en), denoted by $\bar{n}$, for training (filtered to a maximum length of 50), validation and test sets for French-to-English translation for Europarl (EP) and News Commentary (NC) domains.

Model Configuration and Training. Following Nguyen et al. (2017), we employ an architecture of two independent but similar encoder-decoder frameworks for actor and critic, respectively, each using global-attention (Luong et al., 2015) and unidirectional single-layer LSTMs[1]. Both the size of word embedding and LSTM's hidden cells are 500. We used the Adam Optimizer (Kingma and

[1] Our code can be accessed via the link `https://github.com/heidelkin/BIPNMT`.

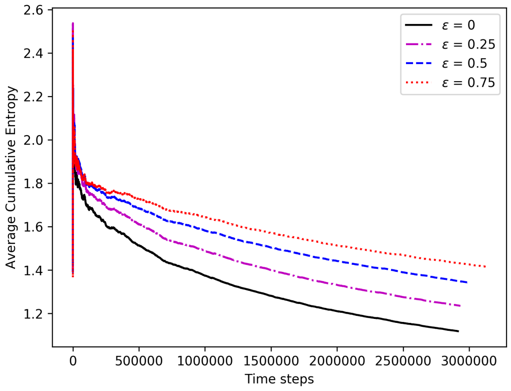


Figure 2: Average cumulative entropy during one epoch of BIP-NMT training with $\mu = 0.8$ and $\epsilon = \{0, 0.25, 0.5, 0.75\}$.

Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. During supervised pre-training, we train with mini-batches of size 64, and set Adam's $\alpha = 10^{-3}$. A decay factor of 0.5 is applied to α, starting from the fifth pass, when perplexity on the validation set increases. During interactive training with bandit feedback, we perform true online updates (i.e. mini-batch size is 1) with Adam's α hyper-parameter kept constant at 10^{-5} for both the actor and the critic. In addition, we clip the Euclidean norm of gradients to 5 in all training cases.

Baselines and Evaluation. Our supervised out-of-domain baseline consists of the actor NMT system described as above, pre-trained on Europarl, with optimal hyperparameters chosen according to corpus-level BLEU on the validation set. Starting from this pre-trained EP-domain model, we further train a bandit learning baseline by employing Nguyen's (2017) actor-critic model, trained on one epoch of sentence-level simulated feedback. The choice of comparing models after one epoch of training is a realistic simulation of a human-system interaction on a sequence of data where each input is seen only once. The feedback signal is simulated with chrF, using character-n-grams of length 6 and a value of $\beta = 2$ of the importance factor of recall over precision. While during training exploration through sampling is essential, during inference and for final model evaluation we use greedy decoding. We evaluate the trained models on our test set from the NC-domain using average sentence-level chrF and standard corpus-level BLEU (Papineni et al., 2002) to measure how well they got adapted to the new domain.

4.2 Results and Analysis

Table 2 shows the results of an evaluation of a baseline NMT model pre-trained by maximum likelihood on out-of-domain data. This is compared to an actor-critic baseline that trains the model of Nguyen et al. (2017) on sentence-level in-domain bandit feedback for one epoch. This approach can already improve chrF (+0.95) and BLEU (+0.55) significantly by seeing bandit feedback on in-domain data. BIP-NMT, with optimal hyperparameters $\epsilon = 0.75$, $\mu = 0.8$ chosen on the validation set, is trained in a similar way for one epoch, however, with the difference that even weaker sub-sentence level bandit feedback is provided on average 5 times per input. We see that BIP-NMT significantly improves both BLEU (+2.18) and chrF (+2.04) by even larger margins.

Table 3 analyzes the impact of the metaparameter ϵ of the BIP-NMT algorithm. We run each experiment three times and report mean results and standard deviation. ϵ controls the margin by which the average word-level entropy needs to increase with respect to the running average in order to trigger a feedback request. Increasing this margin from 0 to 0.25, 0.5 and 0.75 corresponds to decreasing the number of feedback requests by a factor of 3 from around 16 to around 5. This reduction corresponds to a small increase in chrF (+0.29) and a small decrease in BLEU (-0.47).

Figure 2 shows another effect of the metaparameter ϵ: It shows the variation of the average word-level entropy $\bar{H}$ over time steps of the algorithm during one epoch of training. This is computed as a cumulative average, i.e., the value of $\bar{H}$ is accumulated and averaged over the number of target tokens produced for all inputs seen so far. We see that average cumulative entropy increases in the beginning of the training, but then decreases rapidly, with faster rates for smaller values of ϵ, corresponding to more updates per input.

The metaparameter μ controls the threshold of the reward value that triggers a reuse of the prefix for forced decoding. In our experiments, we set this parameter to a value of 0.8 in order to avoid re-translations of already validated prefixes, even if they might sometimes lead to better final full translations. We found the effect of lowering μ from 1.0 to 0.8 negligible on the number of feedback requests and on translation quality but beneficial for the usability.

System	**chrF (std)**	**BLEU (std)**	**Δ chrF**	**Δ BLEU**
Out-of-domain NMT	61.30	24.77	0	0
Nguyen et al. (2017)	62.25 (0.08)	25.32 (0.02)	+0.95	+0.55
BIP-NMT (ϵ = 0.75, μ = 0.8)	63.34 (0.12)	26.95 (0.12)	+2.04	+2.18

Table 2: Evaluation of pre-trained out-of-domain baseline model, actor-critic learning on one epoch of sentence-level in-domain bandit feedback (Nguyen et al., 2017) and BIP-NMT with settings $\epsilon = 0.75$, $\mu = 0.8$ trained on one epoch of sub-sentence level in-domain bandit feedback. Results are given on the NC test set according to average sentence-level chrF and corpus-level BLEU. Result differences between all pairs of systems are statistically significant according to `multeval` (Clark et al., 2011).

ϵ	**chrF (std)**	**BLEU (std)**	**Avg # Requests**	**Δ chrF**	**Δ BLEU**	**Δ Avg # Requests**
0	61.86 (0.06)	25.54 (0.17)	15.91 (0.01)	0	0	0
0.25	62.15 (0.17)	25.84 (0.13)	11.06 (0.07)	+0.29	+0.3	-5
0.5	61.95 (0.05)	25.46 (0.09)	7.26 (0.03)	+0.09	-0.08	-9
0.75	62.15 (0.04)	25.07 (0.12)	4.94 (0.02)	+0.29	-0.47	-11

Table 3: Impact of entropy margin ϵ on average sentence-level chrF score, corpus BLEU and average number of feedback requests per sentence on the NC validation set. The feedback quality threshold μ is set to 0.8 for all models.

4.3 Example Protocols

Table 4 presents user-interaction protocols for three examples encountered during training of BIP-NMT with $\epsilon = 0.75, \mu = 0.8$. For illustrative purposes, we chose examples that differ with respect to the number of feedback requests, the use of the prefix buffer, and the feedback values. Prefixes that receive a feedback $\geq \mu$ and are thus stored in the buffer and re-used for later samples are indicated by underlines. Advantage scores < 0 indicate a discouragement of individual tokens and are highlighted in red.

In the first example, the model makes frequent feedback requests (in 8 of 17 decoding steps) and fills the prefix buffer due to the high quality of the samples. The second example can use the prefix buffer only for the first two tokens since the feedback varies quite a bit for subsequent partial translations. Note how the token-based critic encourages a few phrases of the translations, but discourages others. The final example shows a translation where the model is very certain and hence requests feedback only after the first and last token (minimum number of feedback requests). The critic correctly identifies problematic parts of the translation regarding the choice of prepositions.

5 Conclusion

We presented a novel algorithm, coined BIP-NMT, for bandit interactive-predictive NMT using reinforcement learning techniques. Our algorithm builds on advantage actor-critic learning (Mnih et al., 2016; Nguyen et al., 2017) for an interactive translation process with a human in the loop. The advantage over previously presented algorithms for interactive-predictive NMT is the low human effort for producing feedback (a translation quality judgment instead of a correction of a translatioin), even further reduced by an active learning strategy to request feedback only for situations where the actor is uncertain.

We showcased the success of BIP-NMT with simulated feedback, with the aim of moving to real human feedback in future work. Before deploying this algorithm in the wild, suitable interfaces for giving real-valued feedback have to be explored to create a pleasant user experience. Furthermore, in order to increase the level of human control, a combination with the standard paradigm that allows user edits might be considered in future work.

Finally, our algorithm is in principle not limited to the application of NMT, but can furthermore — thanks to the broad adoption of neural sequence-to-sequence learning in NLP — be extended to other structured prediction or sequence generation tasks.

Acknowledgments.

This work was supported in part by DFG Research Grant RI 2221/4-1. We would like to thank the members of the Statistical NLP Colloquium Heidelberg for fruitful discussions and ideas for improvement of our algorithm.

SRC depuis 2003 , la chine est devenue le plus important partenaire commercial du mexique après les etats-unis .
REF since 2003 , china has become mexico 's most important trading partner after the united states . < /s>

Partial sampled translation	**Feedback**
since	1
since 2003 , china has	1
since 2003 , china has become	1
since 2003 , china has become mexico	1
since 2003 , china has become mexico 's	1
since 2003 , china has become mexico 's most	1
since 2003 , china has become mexico 's most important	1
since 2003 , china has become mexico 's most important trading partner after the us . < /s>	0.8823

SRC la réponse que nous , en tant qu' individus , acceptons est que nous sommes libres parce que nous nous gouvernons nous-mêmes en commun plutôt que d' être dirigés par une organisation qui n' a nul besoin de tenir compte de notre existence .
REF the answer that we as individuals accept is that we are free because we rule ourselves in common , rather than being ruled by some agency that need not take account of us . < /s>

Partial sampled translation	**Feedback**
the	1
the answer	1
the answer we	0.6964
the answer we ,	0.6246
the answer we as individuals allow to 14 are	0.6008
the answer we , as individuals , go down to speak 8 , are being free because we govern ourselves , rather from being based together	0.5155
the answer we , as people , accepts is that we principle are free because we govern ourselves , rather than being led by a organisation which has absolutely no need to take our standards . < /s>	0.5722

SRC lors d' un rallye "journée jérusalem" tenu à l' université de téhéran en décembre 2001 , il a prononcé l' une des menaces les plus sinistres du régime .
REF at a jerusalem day rally at tehran university in december 2001 , he uttered one of the regime 's most sinister threats . < /s>

Partial sampled translation	**Feedback**
in	0
in a round of jerusalem called a academic university in teheran in december 2001 , he declared one in the most recent hostility to the regime . < /s>	0.5903

Table 4: Interaction protocol for three translations. These translations were sampled from the model when the algorithm decided to request human feedback (line 10 in Algorithm 1). Tokens that get an overall negative reward (in combination with the critic), are marked in red, the remaining tokens receive a positive reward. When a prefix is good (i.e. $\geq \mu$, here $\mu = 0.8$) it is stored in the buffer and used for forced decoding for later samples (underlined).

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA.

Bahdanau, Dzmitry, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2008. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Bentivogli, Luisa, Nicola Bertoldi, Mauro Cettolo, Marcello Federico, Matteo Negri, and Marco Turchi. 2016. On the evaluation of adaptive machine translation for human post-editing. *IEEE Transactions on Audio, Speech and Language Processing (TASLP))*, 24(2):388–399.

Chen, Jinying, Andrew Schein, Lyle Ungar, and Martha Palmer. 2006. An empirical study of the behavior of active learning for word sense disambiguation. In *Human Language Technologies: The 2006 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, New York City, NY.

Clark, Jonathan, Chris Dyer, Alon Lavie, and Noah

Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, OR.

Domingo, Miguel, Álvaro Peris, and Francisco Casacuberta. 2018. Segment-based interactive-predictive machine translation. *Machine Translation*.

Foster, George, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12(1-2):175–194.

González-Rubio, Jesús, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2011. An active learning scenario for interactive machine translation. In *Proceedings of the 13th International Conference on Multimodal Interfaces (ICMI)*, Barcelona, Spain.

González-Rubio, Jesús, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2012. Active learning for interactive machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Avignon, France.

Green, Spence, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar.

Judah, Kshitij, Saikat Roy, Alan Fern, and Thomas G. Dietterich. 2010. Reinforcement learning via practice and critique advice. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, GA.

Karimova, Sariya, Patrick Simianer, and Stefan Riezler. 2017. A user-study on online adaptation of neural machine translation to human post-edits. *CoRR*, abs/1712.04853.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA.

Knowles, Rebecca and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, Austin, TX.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Birch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, Czech Republic.

Konda, Vijay R. and John N. Tsitsiklis. 2000. Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Kreutzer, Julia, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

Lewis, David D and William A Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Dublin, Ireland.

Luong, Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.

Marie, Benjamin and Aurélien Max. 2015. Touch-based pre-post-editing of machine translation output. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York, NY.

Nguyen, Khanh, Hal Daumé, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated feedback. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.

Ortiz-Martínez, Daniel, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Los Angeles, CA.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, Philadelphia, PA.

Popović, Maja. 2015. chrF: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translat ion (WMT)*, Lisbon, Portugal.

Sokolov, Artem, Julia Kreutzer, Christopher Lo, and Stefan Riezler. 2016. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain.

Sutton, Richard S. and Andrew G. Barto. 2017. *Reinforcement Learning. An Introduction*. The MIT Press, second edition.

Sutton, Richard S., David McAllester, Satinder Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processings Systems (NIPS)*, Vancouver, Canada.

Tang, Min, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Pennsylvania, PA.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Wuebker, Joern, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.

Machine Translation Evaluation beyond the Sentence Level

Jindřich Libovický[*]
Faculty of Mathematic and Physics
Charles University
libovicky@ufal.mff.cuni.cz

Thomas Brovelli (Meyer) **Bruno Cartoni**
Google
{thomasmeyer,
brunocartoni}@google.com

Abstract

Automatic machine translation evaluation was crucial for the rapid development of machine translation systems over the last two decades. So far, most attention has been paid to the evaluation metrics that work with text on the sentence level and so did the translation systems. Across-sentence translation quality depends on discourse phenomena that may not manifest at all when staying within sentence boundaries (e.g. coreference, discourse connectives, verb tense sequence etc.). To tackle this, we propose several document-level MT evaluation metrics: generalizations of sentence-level metrics, language-(pair)-independent versions of lexical cohesion scores and coreference and morphology preservation in the target texts. We measure their agreement with human judgment on a newly created dataset of pairwise paragraph comparisons for four language pairs.

1 Introduction

Automatic machine translation (MT) evaluation is a crucial technique that accompanied the development of machine translation systems over the last two decades. It allows replacing accurate, but prohibitively slow manual evaluation by a fast and replicable automatic evaluation routine approximating human judgment. So far, the most attention has been paid to the evaluation metrics that work with text on the sentence level and most of the MT systems work at sentence level as well.

The recent advances in neural machine translation (Wu et al., 2016) demonstrated that the state-of-the-art systems, are not too far from the human-level quality on the sentence level. Translating paragraphs or even entire documents is thus becoming a new challenge for MT systems. While this progress is underway, one also needs to assess the translation quality at the paragraph level.

Quality of coherent text translation depends on discourse phenomena that cannot be resolved within sentence boundaries. For instance, the correct sequence of events in the text or the correct placement of gendered pronouns needs to be retained in the target language text to provide a correct translation. Recent experiments with incorporating a broader context into neural machine translation (Wang et al., 2017; Jean et al., 2017) brought only a modest improvement. As these approaches were evaluated using only sentence-level metrics, some important properties of the models might have been missed.

Another important motivation for developing paragraph- or document-level metrics is the growing popularity of reinforcement learning in neural MT, optimizing the model directly towards a given metric (Ranzato et al., 2015; Shen et al., 2016; Gu et al., 2017). If we want to take advantage of this setup at the paragraph level, more elaborated metrics are necessary.

In this paper, we propose several paragraph-level MT evaluation metrics. We evaluate how these metrics agree with human judgment while deciding which translation is better when only a single paragraph of text is used for the comparison on four different language pairs. Because of the lack of annotated data, we create our own

[*] Work done during an internship at Google.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 179–188
Alacant, Spain, May 2018.

dataset consisting of the system outputs submitted to the shared translation tasks of the Workshop on Machine Translation (WMT) between 2014 and 2016 (Bojar et al., 2014; Bojar et al., 2015; Bojar et al., 2016). The dataset with anonymized paragraph translation ratings will be published with the final version of this paper.

The remainder of the paper is organized as follows: Section 2 summarizes the previous work, Section 3 introduces the paragraph-level level metrics, Section 4 describes the evaluation dataset. In Section 5, we describe the experiments we conducted to estimate agreement of the proposed metrics with human judgment.

2 Previous Work

There have been a few attempts so far to measure translation quality beyond the sentence level. With most of the MT frameworks still translating sentence by sentence, there was no urgent need to measure quality at higher levels. The fact that the standard MT scoring methods such as BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002), seem to correlate well with human judgment further supported and established that practice.

With the advent of high-quality sentence-level machine translation (Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017), one of the next challenges is to translate entire paragraphs and documents consistently, i.e. in a lexically coherent and pragmatically appropriate manner. Argumentative structure of text, consistency of lexical choice, and the right 'tone' for its pragmatic intent are the next problems to focus on.

Simple n-gram matching (as with BLEU) and/or allowing for certain word order and synonym variants (as with METEOR (Lavie and Agarwal, 2007)), will likely not be able to capture the aforementioned linguistic phenomena that are crucial for the coherence of the entire text. More aggravatingly, both BLEU and METEOR heavily rely on comparison against one (or sometimes up to 4) human reference translations. These are however not usually available for an entire document. The BLUE score is technically a corpus-level metric because it computes the brevity penalty over the whole corpus. Nevertheless, it does not make use of cross-sentence information in a particularly useful way.

Carpuat (2009) empirically showed that enforcing the one-sense-per-discourse hypothesis by repeating the same words in an MT output can potentially improve the MT quality. Wong and Kit (2012) proposed measuring the semantic similarity of previously seen words in a text in order to capture lexical cohesion of documents in the target language. Lexical cohesion relates to word choice, that Wong and Kit measure by tracking collocation and reiteration (of word stems), additionally allowing for synonyms, near-synonyms and superordinates (for collocation). We take on this approach as well and provide a *language-independent* variant in Section 3.1.

Soricut and Echihabi (2010) on the other hand, viewed the document-level MT evaluation as a ranking problem. They built an MT system that relies on regression models to find BLEU-like numbers for good translations at the document-level which are then ranked higher than others. Similarly to what we will find below, Soricut and Echihabi have shown that an averaged BLEU score over a document is a useful indicator of actual good translation quality and can be used as a feature to find pseudo-reference translations (coming from a secondary MT system) that in turn can be used to estimate the quality of the former MT system.

Similarly, Scarton and Specia (2014) are concerned with quality estimation at the document level, especially when no human reference translations are available. They use a mix of pseudo-reference scores, as Soricut and Echihabi (2010), together with the lexical cohesion features by Wong and Kit (2012). They take the word form repetitions to make the metric language-independent, while we rely on word embeddings that account for richer encoding of synonyms, antonyms etc. than just pure repeated mentions. The main discursive features Scarton and Specia use are LSA scores. They rely on Spearman rank correlation of the word vector of a current sentence compared to all sentences of the document. Whereas both Soricut and Echihabi's and Scarton and Specia's papers need human reference translations or at least pseudo-references for training their regression models, our metrics below can be deployed fully automatically and rely mostly on a monolingual (but automatic) word aligner and freely available, automatic syntactic and semantic parsers.

Hardmeier and Federico (2010) and Miculicich Werlen and Popescu-Belis (2017) use F-score based metrics for pronoun translation evaluation. In Sections 3.2 we take a similar approach from computing coreference preservation.

Besides the approaches presented above, there have also been a few attempts to measure translation quality for certain discourse phenomena in isolation. Meyer et al. (2015) have developed a metric to measure improvements on MT for discourse connectives, whereas for example Gojun and Fraser (2012) and Loaiciga et al. (2014) specifically looked at measuring translation quality for verb tense. Although these approaches have presented interesting results, they can unfortunately not point to the overall translation quality of an entire paragraph.

3 Implemented Paragraph-Level Metrics

We implement two sets of metrics. The first ones operate on the paragraph level and are mostly generalizations of existing MT evaluation metrics (see Section 3.1).

The second set of metrics relies on monolingual word alignment between the reference paragraph and the translation hypothesis (see Section 3.2). Word alignment allows us to measure linguistically motivated statistics about the translation. Nevertheless, alignment errors can pose the danger of bringing additional noise to the evaluation. Moreover, word alignment is only an approximation of what we would really need for thorough document level statistics which would be phrase-level alignment.

In order to find linguistic features (especially entities, coreference and morphology) for the metrics described in the following section, we have been analyzing the respective texts with the Google Cloud Natural Language API[1].

3.1 Metrics without the Monolingual Alignment

Paragraph-Level BLEU. We implemented a simple extension of the standard sentence-level BLEU score (Papineni et al., 2002). Unlike the standard BLEU score, we compute the n-gram statistic throughout the whole paragraph.

The BLEU score is a product of modified n-gram precision and a brevity penalty. The modified n-gram precision approximates the lexical adequacy of the translation and its local fluency. Note that the longer the text is, the less reliable the short n-gram precision becomes because the most frequent words from a language are more likely to get covered by chance. The brevity penalty prevents overrating of longer texts as the probability of accidental covering of the reference text by the hypotheses' n-grams grows with the text length.

Lexical Cohesion Score. One of the features we attribute to a good translation is its stylistic consistency which also includes lexical cohesion. Especially in non-fiction text, we expect the same terms to be used for the same concepts as well as their belonging to the same language register.

Wong and Kit (2012) tried to capture these phenomena in a lexical cohesion score for MT evaluation. The original metric is an average ratio of semantically similar content words observed previously in the text. We propose a language independent extension of the metric.

Formally, we define the score in the following way:

$$\frac{1}{|C|-1}\sum_{i=2}^{|C|}\mathbf{1}\left[\exists c_j : j < i \;\&\; c_i \text{ is related to } c_j\right] \tag{1}$$

where $C = (c_1, \ldots, c_{|C|})$ is a sequence of content words in the text. Semantic similarity was originally defined by a graph distance threshold in WordNet (Fellbaum, 1998) which does not have sufficiently high coverage for languages other than English.

In order to overcome this drawback we reformulate the score:

$$\frac{1}{|C|-1}\sum_{i=2}^{|C|}\max_{j=1..i-1}\operatorname{sim}(c_i, c_j). \tag{2}$$

As function sim, we use cosine similarity of pretrained word embeddings (Mikolov et al., 2013) instead of the binary indication of semantic similarity based on WordNet.

3.2 Metrics Requiring Monolingual Alignment

For monolingual alignment, we re-implemented the state-of-the-art rule-based monolingual aligner (Sultan et al., 2014). In order to make the aligner language-independent, we transferred the rules for finding equivalent dependency structures from Stanford-style dependencies to Universal

[1] Publicly available at: `https://cloud.google.com/natural-language/docs/`

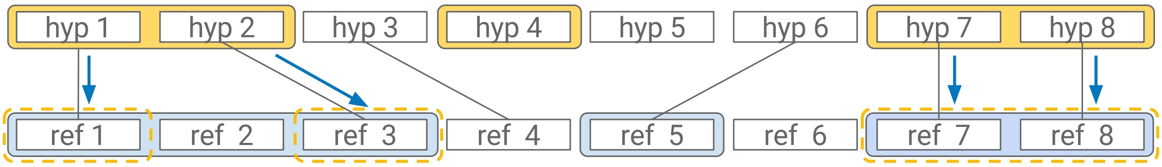


Figure 1: Examples of coreference chain projection via monolingual alignment.

	Prec.	Recall	F_1	Acc.
METEOR	89.7	71.0	79.3	10.8
Our aligner	88.2	65.7	75.3	10.1

Table 1: Comparison of the METEOR aligner and our aligner on the Edinburgh++ dataset.

Dependencies (Marneffe et al., 2014). Unlike the original aligner, our aligner does not require explicitly aligned sentences and is agnostic to the sentence boundaries as it is treating the paragraphs as dependency forests.

The alignment algorithm is a pipeline of rule-based steps. In the first step, it aligns identical word sequences and named entities. In the second step, the dependency surroundings of already aligned content words are aligned if their dependency labels belong to manually designed categories. Then, linear surroundings of the content words are aligned if the words in the surroundings are similar enough. For that purpose, we use the lexical paraphrases table of PPDB 2.0 paraphrase database (Pavlick et al., 2015) and a word embedding distance. We repeat the procedure for non-content words, with the only difference that we use semantic similarity in the dependency context alignment as well.

The aligner has similar results to the METEOR aligner (Lavie and Agarwal, 2007) when comparing against the Edinburgh++ dataset (Cohn et al., 2008) (see Table 1). It does not use longer phrases from the paraphrase database which would increase the aligning complexity prohibitively in case of long texts.

Paragraph-Level METEOR. We extended the METEOR score to operate on the paragraph level in a straightforward manner. As the standard METEOR, it is a product of a disfluency score d and an adequacy score a. The disfluency score is computed as

$$d = \frac{1}{2}\left(\frac{\text{\# alignment steps}}{\text{\# unigrams matched}}\right)^3 \tag{3}$$

and captures how much the hypothesis paragraph would need to be torn apart in order to be aligned with the reference.

Lexical adequacy is computed as a weighted harmonic mean of precision and recall:

$$a = \frac{10 \cdot P \cdot R}{R + 9P} \tag{4}$$

where P and R stands for precision and recall of the hypothesis words computed over the monolingual alignment.

We evaluate two methods of computing precision and recall. In the standard way, which we refer to as *Hard METEOR*, we assign a unit weight to all alignment links. As an alternative, we introduce *Soft METEOR* where we weight the alignment links by word similarity estimated from word embeddings distance and weight the precision and recall accordingly.

Morphology Preservation. Similarly to the METEOR score, where we compute the lexical adequacy of all words in the text, we can measure preservation of morphological categories that can provide information about phenomena that are crossing sentence boundaries.

As in METEOR, we measure the F-score of the morphological categories being the same. The F-score takes into account also the false negatives and false positives. Alternatively, we calculate the accuracy of only those word pairs that have been aligned together. Computing the accuracy instead of the F-measure is more appropriate in cases where morphological categories are not well-covered by the monolingual alignment, e.g. pronouns.

We measure the preservation of pronoun number and gender, which should capture the extent of coreference chains throughout the text. Additionally, verb gender, tense and number will also capture how the sequence of described events is preserved between the translation hypothesis and the reference.

Computing morphology preservation is not only limited by the quality of the monolingual align-

English source: *The fertile ground and the rainforest climate of Isla del Rey are ideal for growing marijuana plants. Three days ago the authorities in Panama tore out the 4,500 plants and burnt them.*

German reference	System A	System B
Der fruchtbare Boden und das regenwaldtypische Klima der Isla del Rey sind für das Gedeihen der Marihuana-Pflanzen bestens geeignet. Seit drei Tagen reissen die Behörden Panamas die 4500 Pflanzen aus und verbrennen sie.	*Der fruchtbare Boden und das Regenwaldklima von Isla del Rey sind ideal, um Marihuanaanlagen zu wachsen. Vor drei Tagen rissen die Behörden in Panama die 4.500 Anlagen heraus und verbrannten sie.*	*Der fruchtbare Boden und das Regenwaldklima von Isla del Rey sind ideal für wachsende Marihuana-Pflanzen. Vor drei Tagen haben die Behörden in Panama die 4.500 Pflanzen ausgebrannt und verbrannt.*
Sentence-level BLEU	.229	.233
Sentence-level METEOR	.392	.376
Sentence-level TER	.545	.545
Paragraph BLEU score	.203	.229
Coreference: BLANC	.791	.890
Coreference: Non-link F_1 score	.966	.980
Hypotheses lexical cohesion	.594	.632
Meteor – hard	.447	.530
Meteor – soft	.434	.522
Pronoun gender accuracy	.941	.900
Pronoun number accuracy	1.000	.950
Verb gender F_1 score	.667	.500
Verb number F_1 score	.667	.250
Verb tense F_1 score	.222	.250

Table 2: Score values for the implemented document-level metrics. This illustrates proof-of-concept and good correlation with sentence-level metrics.

ment. It can also generate false positives, e.g. in cases where grammatical gender is not preserved because of different but still correct lexical choice. We believe that averaged over a longer dataset, this type of metrics can still bring interesting linguistic insight.

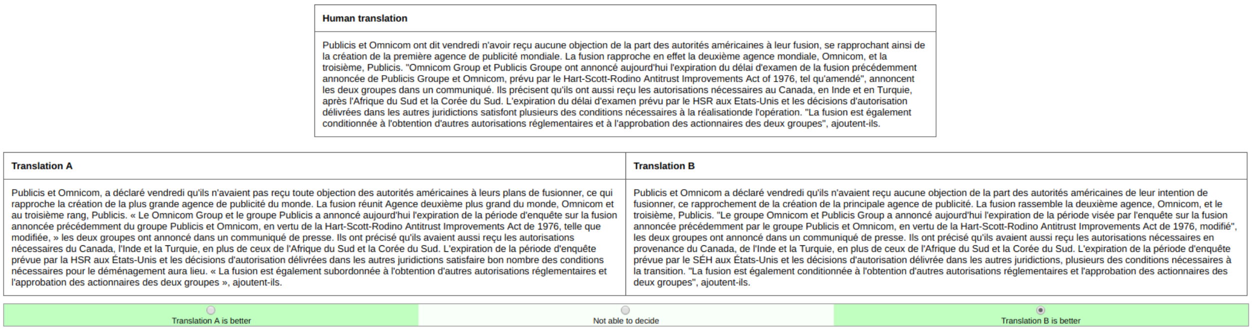
Coreference Preservation. Coreference chains can easily get broken during machine translation, especially when the translation is done on the sentence level. Except for indirect measurements of the coreference preservation via morphological categories of pronouns and verbs, we also explicitly compute coreference preservation via projection of the reference coreference chains to the translation hypothesis.

We apply entity and coreference resolution on the translation hypothesis (by detecting all nominal elements such as noun phrases, proper names and pronouns, as well as their coreference links). We project these mentions of entities in the hypothesis text to the reference translation using the alignment links as illustrated in Figure 1. No restrictions are imposed on this projection, so that the projected mentions do not even have to be continuous chunks of text. This also gives a mention matching that can be used during metric computation.

Once the projection is done, we treat the coreference chains in the reference text as the ground truth and measure the quality of the projected chains (i.e. treat them as the response).

There are two main approaches to the evaluation of coreference resolution. We can either measure how well the resolver spotted the words in the entity mentions or how well it preserved the coreference links. We therefore implemented two coreference metrics: the B^3 average F_1 score for treating the problem as retrieval of mentions, and the BLANC score (Luo et al., 2014), which is an average of the F_1 score of the coreference links and the F_1 measure of the complements of the coreference links (complement of the complete graph).

Table 2 shows the values of the document-level translation evaluation metrics in a real example from the WMT 2016 test set. When judged by a human, the hypothesis from system A is slightly



Human translation

Publicis et Omnicom ont dit vendredi n'avoir reçu aucune objection de la part des autorités américaines à leur fusion, se rapprochant ainsi de la création de la première agence de publicité mondiale. La fusion rapproche en effet la deuxième agence mondiale, Omnicom, et la troisième, Publicis. "Omnicom Group et Publicis Groupe ont annoncé aujourd'hui l'expiration du délai d'examen de la fusion précédemment annoncée de Publicis Groupe et Omnicom, prévu par le Hart-Scott-Rodino Antitrust Improvements Act of 1976, tel qu'amendé", annoncent les deux groupes dans un communiqué. Ils précisent qu'ils ont aussi reçu les autorisations nécessaires au Canada, en Inde et en Turquie, après l'Afrique du Sud et la Corée du Sud. L'expiration du délai d'examen prévu par le HSR aux Etats-Unis et les décisions d'autorisation délivrées dans les autres juridictions satisfont plusieurs des conditions nécessaires à la réalisationde l'opération. "La fusion est également conditionnée à l'obtention d'autres autorisations réglementaires et à l'approbation des actionnaires des deux groupes", ajoutent-ils.

Translation A	**Translation B**
Publicis et Omnicom, a déclaré vendredi qu'ils n'avaient pas reçu toute objection des autorités américaines à leurs plans de fusionner, ce qui rapproche la création de la plus grande agence de publicité du monde. La fusion réunit Agence deuxième plus grand du monde, Omnicom et au troisième rang, Publicis. « Le Omnicom Group et le groupe Publicis a annoncé aujourd'hui l'expiration de la période d'enquête sur la fusion annoncée précédemment du groupe Publicis et Omnicom, en vertu de la Hart-Scott-Rodino Antitrust Improvements Act de 1976, telle que modifiée, » les deux groupes ont annoncé dans un communiqué de presse. Ils ont précisé qu'ils avaient aussi reçu les autorisations nécessaires du Canada, l'Inde et la Turquie, en plus de ceux de l'Afrique du Sud et la Corée du Sud. L'expiration de la période d'enquête prévue par la HSR aux États-Unis et les décisions d'autorisation délivrées dans les autres juridictions satisfaire bon nombre des conditions nécessaires pour le déménagement aura lieu. « La fusion est également subordonnée à l'obtention d'autres autorisations réglementaires et l'approbation des actionnaires des deux groupes », ajoutent-ils.	Publicis et Omnicom a déclaré vendredi qu'ils n'avaient reçu aucune objection de la part des autorités américaines de leur intention de fusionner, ce rapprochement de la création de la principale agence de publicité. La fusion rassemble la deuxième agence, Omnicom, et le troisième, Publicis. "Le groupe Omnicom et Publicis Group a annoncé aujourd'hui l'expiration de la période visée par l'enquête sur la fusion annoncée précédemment par le groupe Publicis et Omnicom, en vertu de la Hart-Scott-Rodino Antitrust Improvements Act de 1976, modifié", les deux groupes ont annoncé dans un communiqué de presse. Ils ont précisé qu'ils avaient aussi reçu les autorisations nécessaires en provenance du Canada, de l'Inde et la Turquie, en plus de ceux de l'Afrique du Sud et la Corée du Sud. L'expiration de la période d'enquête prévue par le SÉH aux États-Unis et les décisions d'autorisation délivrée dans les autres juridictions, plusieurs des conditions nécessaires à la transition. "La fusion est également conditionnée à l'obtention d'autres autorisations réglementaires et l'approbation des actionnaires des deux groupes", ajoutent-ils.

○ Translation A is better ○ Not able to decide ● Translation B is better

Figure 2: Example evaluation task for human annotators

better, but has a lower sentence-level BLEU score than system B. Our document-level metrics can hint at the better quality of A with e.g. the lexical cohesion score as well as the pronoun and verb morphology scores.

4 Dataset

Unlike sentence-level MT evaluation which can benefit from evaluation campaigns like the WMT tasks of annual metrics evaluation (Bojar et al., 2017), there is no dataset consisting of human judgments on machine translation quality beyond the sentence level. Even the metrics that were discussed in Section 2 were only evaluated against human judgments collected at the sentence level.

In order to evaluate our metrics reliably, we created a new dataset consisting of pairwise paragraph comparisons of machine translation outputs that have been rated by several human annotators per pair. The paragraphs are extracted from the freely accessible test sets provided for the WMT workshops (years 2014 to 2016). Our rated data sets will be made available publicly with the final version of this paper.

4.1 Pilot Annotation

In order to determine a reasonable length for paragraphs to be evaluated by human raters, we conducted a pilot experiment where we sampled 30 paragraphs from the WMT datasets for the English to German, German to English, English to French and French to English translation directions. The length of these paragraphs has arbitrarily been set to approximately 180 words each. At this stage, the target side translations have been sampled randomly from system outputs submitted to the WMT shared news tasks of the years 2014 to 2016. The annotators were provided with a simple user interface that showed them the human reference translation, a system output A to the left and a system output B to the right. The annotators task was to select either *system A is better*, *undecided* or *system B is better* compared to the reference translation (Figure 2). In the pilot round, the evaluators were trained linguists and native speakers of the target languages. The annotators were afterwards informally interviewed.

We learned from the feedback of annotators that the sampled paragraph length of 180 words is enough to capture phenomena in translation that cross sentence boundaries. Metric-wise, our paragraph-level extensions of BLEU and METEOR are reasonable choices, especially for English to French and French to English translation and align well with the human judgment (which is not to be expected to be perfect either when rating over several sentences). Lexical cohesion difference and linked-based coreference scores also confirm that the more lexically coherent a paragraph is, the higher it is rated by humans, independently of the reference translation. The annotators relative agreement was over 70 % ($\kappa = 0.4$) and only a minority of paragraph pairs remained undecided.

4.2 Large-Scale Annotation

The annotation of a bigger evaluation dataset was done for four language pairs: English to Czech, English to German, English to French and English to Russian. The paragraphs were randomly sampled from the same set of WMT system submissions as in the pilot round[2]. In addition to the MT systems submitted to WMT, we also translated the sampled paragraphs with Google's neural MT (Wu et al., 2016).

Unlike the pilot round, which was conducted

[2] If you would like to use the dataset, please use the following form: https://goo.gl/forms/zvpOddi9FelFkJxJ2.

language pair		agr.	κ	BLEU	ΔBLEU
en → cs	all	.68	.53	12.3	6.1
	good	.42	.12	22.9	5.8
en → de	all	.55	.33	17.0	6.1
	good	.37	.06	26.0	5.5
en → fr	all	.58	.37	25.0	8.6
	good	.40	.05	36.5	6.5
en → ru	all	.58	.37	20.9	8.9
	good	.42	.13	30.5	6.7

Table 3: Statistics on the collected dataset: annotator agreement (agr.) as a proportion of cases when all three annotators agreed, Cohen's κ, average BLEU score and average BLEU score difference (ΔBLEU). Labels 'good' and 'all' refer the quality of the translation the paragraphs were sampled from. The former contains pairs of paragraphs only from outputs of systems that achieved a total sentence-level BLEU score of over 30 points on the selected paragraphs. The latter contains samples irrespective of BLEU scores (also see Section 4.2).

by trained linguists, the only requirement for this larger crowd-sourced annotation was that the raters must be native speakers of the target language and must understand English. Every paragraph pair was evaluated independently by three raters and the majority vote was used as final rating decision.

To be able to better evaluate how the document-level metrics behave under different circumstances, we created two test sets for each of the language pairs. In the first test set, the paragraphs are sampled randomly from the WMT submissions which are often of different quality. The second, more challenging test set, contains pairs of paragraphs only from outputs of systems that achieved a total sentence-level BLEU score of over 30 points on the selected paragraphs. Both variants contain 400 paragraph pairs for all the four language pairs. The statistics of the dataset are tabulated in Table 3. One notable fact is that the annotator agreement (proportion of cases when all three annotators agreed) is relatively low and even decreases when using a higher quality system.

5 Experiments

We evaluated the metrics proposed in Section 3 on the collected datasets on English to German and English to French translation directions. For every metric, we computed the proportion of cases when the paragraph annotated as the better one has also been assigned the higher score, i.e. which of the two system outputs provides a better entire paragraph translation when comparing to the reference. All the paragraphs were also evaluated with the standard sentence-level metrics (BLEU, METEOR, TER)[3]. The detailed results are presented in Table 4.

If we interpret the annotator agreement as probability that all three annotators agree, we can factorize this probability into two steps: first that two agreed (and thus did the majority vote) and that the third annotator agreed with them. Therefore, we can estimate the probability of the third annotator agreeing with the majority vote as a square root of the annotator agreement. These are presented in the first line of Table 4.

The main finding of the analysis is that the agreement of both the traditional sentence-level metrics and the proposed metrics with the human judgment is relatively low in pairwise comparison. In fact, only a small majority of the pairwise comparisons is done correctly. This particular finding contradicts the training techniques based on the REINFORCE algorithm (Williams, 1992) where the update rule explicitly contains the pairwise comparison. Moreover, it is not clear whether there is a room for improvement given that for good translation systems, the performance of the automatic metrics is on par with the estimated human agreements.

The other interesting result is that it is possible to estimate which translation is better almost equally well when focusing only on a particular phenomenon (coreference, lexical cohesion, morphology) as with metrics that should capture the translation quality holistically (METEOR, BLEU).

The metrics based on morphological analysis achieved better performance on paragraph pairs consisting of good translations. It might be so because the morphological analysis is more likely to fail in case of malformed translation outputs where the monolingual alignment is more difficult, because the hypothesis is different from the reference.

A similar trend can also be observed for coreference preservation. The BLANC score used for coreference evaluation is an average of F_1 scores of estimating correctly the coreference links and

[3] We used the metrics as implemented in MultEval, `https://github.com/jhclark/multeval`

	en → de		en → fr	
Metric	good	all	good	all
Estimated human agreement	.610	.743	.629	.762
Sentence-level BLEU	.615	.594	.643	.629
Sentence-level METEOR	.612	.594	.640	.629
Sentence-level TER	.567	.559	.610	.594
Paragraph BLEU score	.610	.572	.658	.629
Coreference: BLANC	.577	.428	.533	.542
Coreference: Non-link F_1 score	.584	.425	.538	.548
Hypotheses lexical cohesion	.542	.489	.635	.499
Meteor – hard	.587	.562	.640	.598
Meteor – soft	.584	.562	.643	.601
Pronoun gender accuracy	.484	.438	.495	.505
Pronoun number accuracy	.524	.348	.443	.433
Verb gender F_1 score	.529	.198	.510	.492
Verb number F_1 score	.537	.214	.510	.464
Verb tense F_1 score	.537	.208	.508	.495

Table 4: Average agreement of the proposed metrics with the majority vote on human judgment on pairwise paragraph comparison. Columns denotes as 'all' contain randomly sampled system pairs, columns denoted as 'good' contain only pairs where both compared paragraphs achieved a BLEU score of at least 30.

its complement-non-link relations. Often, a better aggrement was achieved with the score computed only over the non-link relations which are much denser than the coreference links. We hypothesize this makes the score more robust to alignment errors.

6 Conclusions

The presented study focused on two main new contributions.

First, we implemented an entire package of automatic paragraph-level MT quality metrics that are language-(pair)-independent and track MT quality at different levels throughout entire paragraphs. Our extensions of the METEOR and lexical cohesion scores thereby showed promising results for most adequately and consistently measuring paragraph-level MT quality. We also experimented with more linguistically motivated scores, such as coreference preservation that could be interesting for future experiments, once the alignment of pronouns and referential expressions is more reliable.

Second, we prepared a dataset of human judgments on pairwise comparisons of MT quality at the paragraph level which can be used for new metrics evaluation. The dataset consists of system translations from English to Czech, French, German and Russian submitted to WMT in recent years. For each language pair, 400 pairwise comparisons of randomly selected paragraphs and another 400 pairs of more similar, high-quality translation pairs have been rated humanly for paragraph translation quality.

Future work will try to improve the monolingual alignment. Better performance of parsers and coreference resolvers would indirectly also help the presented metrics. Integration of pseudo-references (where no human reference translations are available) and training an ensemble of all the metrics in our package can also be a promising direction.

References

[Bojar et al.2014] Bojar, Ondřej, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

[Bojar et al.2015] Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September. Association for Computational Linguistics.

[Bojar et al.2016] Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August. Association for Computational Linguistics.

[Bojar et al.2017] Bojar, Ondřej, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 489–513, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Carpuat2009] Carpuat, Marine. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27. Association for Computational Linguistics.

[Cohn et al.2008] Cohn, Trevor, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Comput. Linguist.*, 34(4):597–614, dec.

[Fellbaum1998] Fellbaum, Christiane. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May.

[Gehring et al.2017] Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In Precup, Doina and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug. PMLR.

[Gojun and Fraser2012] Gojun, Anita and Alexander Fraser. 2012. Determining the placement of german verbs in english-to-german smt. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 726–735, Avignon, France. Association for Computational Linguistics.

[Gu et al.2017] Gu, Jiatao, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1958–1968, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Hardmeier and Federico2010] Hardmeier, Christian and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation, Paris.*

[Jean et al.2017] Jean, S., S. Lauly, O. Firat, and K. Cho. 2017. Does neural machine translation benefit from larger context? *CoRR*, abs/1704.05135, apr.

[Lavie and Agarwal2007] Lavie, Alon and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Loaiciga et al.2014] Loaiciga, Sharid, Thomas Meyer, and Andrei Popescu-Belis. 2014. English-french verb phrase alignment in europarl for tense translation modeling. In *Proceedings of the 9th international conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland.

[Luo et al.2014] Luo, Xiaoqiang, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of blanc to system mentions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Baltimore, Maryland, June. Association for Computational Linguistics.

[Marneffe et al.2014] Marneffe, Marie-Catherine De, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: a cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

[Meyer et al.2015] Meyer, T., N. Hajlaoui, and A. Popescu-Belis. 2015. Disambiguating discourse connectives for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1184–1197.

[Miculicich Werlen and Popescu-Belis2017] Miculicich Werlen, Lesly and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (apt). In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 17–25, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Mikolov et al.2013] Mikolov, Tomáš, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

[Papineni et al.2002] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Pavlick et al.2015] Pavlick, Ellie, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

[Ranzato et al.2015] Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

[Scarton and Specia2014] Scarton, Carolina and Lucia Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*, pages 101–108. Association for Computational Linguistics.

[Shen et al.2016] Shen, Shiqi, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany, August. Association for Computational Linguistics.

[Soricut and Echihabi2010] Soricut, Radu and Abdessamad Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 612–621. Association for Computational Linguistics.

[Sultan et al.2014] Sultan, Md, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230.

[Vaswani et al.2017] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

[Wang et al.2017] Wang, Longyue, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. *CoRR*, abs/1704.04347.

[Williams1992] Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

[Wong and Kit2012] Wong, Billy T. M. and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068, Jeju Island, Korea, July. Association for Computational Linguistics.

[Wu et al.2016] Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

An Analysis of Source Context Dependency in Neural Machine Translation

Xutai Ma
Electrical and Computer Engineering
Johns Hopkins University
xutai_ma@jhu.edu

Ke Li
Electrical and Computer Engineering
Johns Hopkins University
kli26@jhu.edu

Philipp Koehn
Computer Science Department
Johns Hopkins University
phi@jhu.edu

Abstract

The encoder-decoder with attention model has become the state of the art for machine translation. However, more investigations are still needed to understand the internal mechanism of this end-to-end model. In this paper, we focus on how neural machine translation (NMT) models consider source information while decoding. We propose a numerical measurement of source context dependency in the NMT models and analyze the behaviors of the NMT decoder with this measurement under several circumstances. Experimental results show that this measurement is an appropriate estimate for source context dependency and consistent over different domains.

1 Introduction

Neural machine translation (NMT) with encoder-decoder structure and attention mechanism (Bahdanau et al., 2015; Luong et al., 2015) has achieved great success on several machine translation tasks. Different from phrase based systems, neural machine translation is trained end-to-end and learns the alignment and translation jointly.

At each decoding step, the alignment is predicted in the attention layer and represented as a distribution over words in a source sequence. Then the source context information, which is an attention-weighted sum over encoder hidden states, is fed into the decoder for the prediction of the next word.

The decoder in a NMT model is similar to a recurrent neural network language model (RNNLM) (Mikolov et al., 2010), with additional input from the source side. It takes the previous hidden state, the previous predicted target word embedding, and source context information as inputs and produces a distribution over the next target words.

This end-to-end approach can achieve state-of-the-art performance on several machine translation tasks. Joint training of the translation model and alignment gives a soft alignment between source side and target side. However, some of its flaws are observed under certain settings (Koehn and Knowles, 2017). One of the most common and important issue of neural machine translation is that it often generates fluent but inadequate translations especially under domain mismatch conditions.

An example[1] is shown in Figure 1. Here, the translation generated by the NMT models — while being fluent English — has no semantic connection to the source sentence. Moreover, out-of-domain models cause even more severe inadequacy.

An intuitive explanation for this observation is that the NMT decoder lacks effective attention to the source information. Because of the similarity between the NMT decoder and an RNNLM, it is possible that NMT models generate sentence based on its internal language model without properly taking advantage of the source information. While several researchers have explored the atten-

[1] In this example, we choose a sentence that has been processed by byte pair encoding (BPE) (Sennrich et al., 2016) and "@@" is used as a splitter token. The reason for this is that BPE has become a standard pre-processing step, which helps reducing the vocabulary size. Long words in original text will be split into sub-word "phrases". It is also very interesting to investigate how sub-word prediction related to the source information.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 189–197
Alacant, Spain, May 2018.

Source	*der hat also die Was@@ er@@ stoff@@ emission bei verschiedene Frequ@@ enzen aufgenommen .*
Reference	*it recorded the hydrogen radio emission at different frequencies .*
In-domain Translation	*so he recorded the security clearance on several frequencies .*
Out-of-domain Translation	*indeed , He has [more] example in suc@@ cession .*

Figure 1: An example of an inadequate translation of a Germany to English NMT model. In domain data is subtitle dataset and out of domain data is koran dataset

tion mechanism in NMT, none of them have numerically analyzed whether an NMT decoder sufficiently utilizes the source information.

In this paper, we propose a numerical approach for source context dependency analysis in NMT models. We list some reasons why we should care this dependency.

1. While translation of content words, such as nouns and verbs, highly depends on source information, function words, such as determiners and prepositions, depend more on language-internal properties. We want to investigate whether NMT models are able to learn this difference.

2. The NMT decoder functions similarly to a RNN language model. It takes both previous hidden state, the previous predicted word embedding, and the source context vector as inputs for every recurrent neural network (RNN) cell. It is possible that under certain circumstances the source context vector has little impact on updating the state in the decoder. That means the decoder may fail to use sufficient information from the source sentence. This could be one of the reasons why NMT models sometimes generate fluent but inadequate sentences.

3. As observed by Koehn and Knowles (2017), under some data conditions such as domain mismatch, some failed translations seem to ignore the source sentence. By analyzing source context dependency, we can gain insight into the reason of the failure.

Our contributions in this paper include:

- We propose a numerical measurement for source context dependency in NMT models. It is based on the distribution of words generated from the decoder. The measurement is very general to sequence to sequence models and their variations.

- We carried out a series of experiments under different settings to analyze the behavior of NMT models with this measurement. Moreover, we numerically analyze source context dependency related to part of speech categories, domain mismatch and translation length.

2 Related Work

A number of researchers have been working on exploring the “black box” of neural machine translation models. Belinkov et al. (2017a) investigated how NMT models learn word structure and representation quality on part-of-speech and morphological tags. Belinkov et al. (2017b) and Dalvi et al. (2017) explored the capability of representation in NMT hidden layers of part-of-Speech and semantic tagging in neural machine translation using multi-task training.

There is some research focusing on the attention mechanism in NMT. Liu et al. (2016) proposed a training scheme to learn attention under the guidance from conventional alignment models. Cohn et al. (2016) incorporates structural alignment biases to improve the alignment quality learned in the attention layer. Ghader and Monz (2017) proposed a numerical approach for analyzing the capability of attention.

Some research also focuses on analysis and visualization of NMT models for better understanding. Visualization of attention weights is a common tool for NMT analysis (Ding et al., 2017). Moreover, Shi et al. (2016) correlated activation values of individual LSTM nodes in the translation model with the length of the translated sentences.

Some research has addressed a similar topic as we tackle in our paper. Instead of doing numerical analysis, they proposed new structures to improve both the adequacy and fluency in NMT. Tu et al. (2017) proposed a context gate structure in NMT decoders to control the portion of source or target side information fed into the decoder and they obtained a 2.3 BLEU points improvement compared a standard attention based NMT baseline. Zheng et al. (2018) introduced a novel mechanism to separate the source information into two parts: translated past context and untranslated future context. They fed the two parts to both the attention model and the decoder states and reported improvement on several translation tasks compared with the conventional coverage model.

3 Methodology

3.1 Neural Machine Translation

A variety of alternative neural machine translation approaches have been recently proposed (Gehring et al., 2017; Vaswani et al., 2017). In this paper, we will focus on the most common model used today, the encoder-decoder based NMT model with an attention layer (Bahdanau et al., 2015; Luong et al., 2015).

The encoder in the neural machine translation model is a bi-directional recurrent neural network structure which encodes the source tokens sequence into a sequence **H** of context-related vector representations h_j upon an embedding layer.

$$\mathbf{H} = h_0, h_1, \ldots, h_{n-1}, h_n \tag{1}$$

The decoder of the NMT model is a recurrent neural network (Elman, 1990). There are several widely used variations, such as Long Short Time Memory (Gers et al., 1999) and Gated Recurrent Unit (GRU) (Chung et al., 2014) In this paper, we choose to use GRU for analysis.

Let us now introduce the structure of the NMT decoder. In decoder, the distribution for next possible words at each step is generated by:

$$P(y_i \mid y_{<i}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{2}$$

where **x** is a sequence of vectors representing the source sentence, and s_i is RNN hidden state and calculated by:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{3}$$

g and f are some nonlinear functions.

The context vector c_i at step i comes from:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{4}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{5}$$

where

$$e_{ij} = a(s_{i-1}, h_j) \tag{6}$$

is an alignment model which scores how well inputs around position j and the output at position i match. h_j is the encoder hidden state at step j.

3.2 Source Context Dependency Measurement

If an NMT model properly considers source context information, a significant difference should be observed between distributions with and without the source context vector. Considering this, we propose a distribution distance based method to calculate the source context dependency in an NMT model.

We first train an attention based NMT model. During decoding we have two decoders, a main decoder and an auxiliary decoder as shown in Figure 2. The main decoder is a normal NMT decoder with the source context vector computed by a weighted sum of encoder hidden states. The auxiliary decoder shares parameters with the NMT decoder but zeros out the source context vector at each decoding step. The previous predicted target word embedding for the auxiliary decoder is from the main NMT decoder. The hidden states of the NMT and auxiliary decoders are denoted separately as s_i and s_i^{aux} in Figure 2 at each step i while they are the same indeed.

We then introduce the source context dependency measure. At i-th decoding step, we have two distributions for predicting the next translated word given history and context from the NMT decoder and the auxiliary decoder denoted as $P_{main}(y_i)$ and $P_{aux}(y_i)$ in Figure 2, where y_i is the i-th predicted word. We then define the source context dependency measure of word y_i as

$$D_{y_i}^{p} = d_{KL}\left(P_{main}(y_i), P_{aux}(y_i)\right) \tag{7}$$

$$= d_{KL}\Big(P(y_i \mid y_{<i}, c_i), P(y_i \mid y_{<i}^{aux}, \vec{\mathbf{0}})\Big) \tag{8}$$

$$= d_{KL}\Big(g(y_{i-1}, s_i, c_i), g(y_{i-1}, s_i^{aux}, \vec{\mathbf{0}})\Big) \tag{9}$$

where

1. d_{KL} is a function to calculate the KL-divergence between the two distributions.

2. $P_{main}(y_i) = P(y_i \mid y_{<i}, c_i)$ is the distribution over the next word given history information and source context vector c_i at step i.

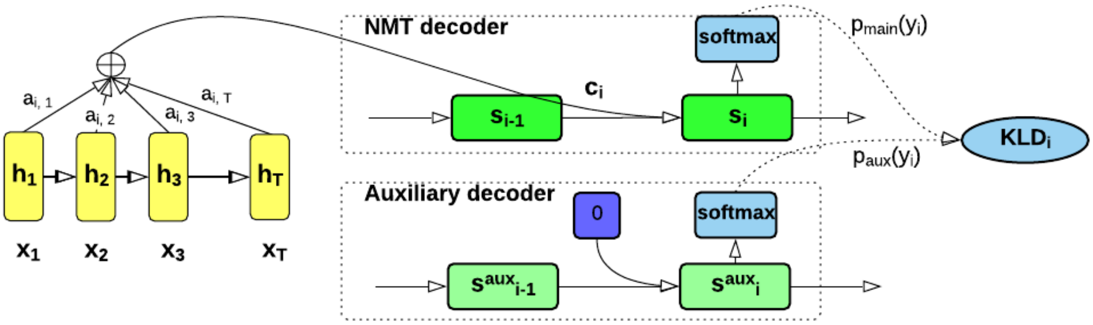


Figure 2: Context dependency measure: Both standard NMT and an auxiliary decoder that ignores the source context make word predictions. We measure the KL divergence between these predictions.

3. $P_{aux}(y_i) = P(y_i \mid y^{aux}_{<i}, \vec{0})$ is the distribution over the next word given history information and a zeroed out source context vector at step i. Notice that $y^{aux}_{<i}$ and $y_{<i}$ are actually the same sub-sequence. However, we are using an "aux" superscript here to emphasize that their representations, which are the hidden states, are different when predicting the next word.

The first distribution comes from main decoder and second distribution comes from auxiliary decoder.

Notice that we compute source context dependency scores during decoding, not training. Furthermore, it is also compatible with beam search. In addition to main decoder hidden states and previous predictions, hidden states from auxiliary decoder and source context dependency scores of previous words are also tracked for each hypothesis in the beam. Since the two decoders share parameters, no additional training is needed for the source context dependency calculation given a trained NMT model.

An alternative implementation for computing the source context dependency score would be to only use one decoder. At each decoding step, we can calculate the distance between distributions from the main decoder with and without the source context vector. However, the previous hidden state potentially contains both history and previous source context information. Thus, source context creeps into the decoder state. With a auxiliary decoder, we can completely eliminate the influence of source context.

4 Experimental Setup

We use the toolkit Nematus (Sennrich et al., 2017) for training and decoding. We use the gated recurrent unit (GRU) (Chung et al., 2014) in both encoder and decoder with a dimension of 1024. The dimension of embedding layer is 500. For optimizer, Adadelta (Zeiler, 2012) with learning rate 0.0001 is used. Dropout (Srivastava et al., 2014) with 0.2 probability was used to prevent overfitting. For decoding, we use beam search with a beam width 12.

Byte pair encoding (BPE) (Sennrich et al., 2016) is used for processing training data to fit a 50,000 subwords vocabulary limit. We use BPE since it has been a very popular preprocessing procedure for machine translation, so that our evaluation method can be used in more general cases.

In part-of-speech (POS) analysis, we use Stanford POS tagger (Toutanova et al., 2003) with a universal POS tagset. We first convert the translated subwords to complete words and tag the sequences with the Stanford POS tagger We find that the amount of subwords is significantly smaller than complete words. So we let each subword inherit the tag from the corresponding complete word[2].

We carried out our experiments on German–English translation tasks. We used five corpora in five domains from OPUS (Tiedemann, 2012), which is briefly described in Table 1. We use five corpora because we want to show that our metric and its analysis are general and consistent over different domains. Moreover, we would like to know

[2] An alternative would be to distinguish tags for split and unsplit words. We did this as well, but found no significant difference.

Dataset	Abbreviates	Descriptions	Size(English)
OpenSubtitle2016	subtitles	Translatio Movie subtitles	118.8M
JCR-Acquis	acquis	Legislative text of the European Union	34.1M
EMEA	emea	Documents from the European Medicines Agency	12.0M
Tanzil	koran	Translations of Koran	11.3M
IT	it	Documents of GNOME, OpenOffice, KDE, PHP, Ubuntu	2.6M

Table 1: Summary of five corpora from OPUS

how domain mismatch affects the source content dependency.

5 Analysis

5.1 Auxiliary Decoder

We briefly described the auxiliary decoder above in Section 3.2. The basic assumption is that an auxiliary decoder contains history information and behaves similar to a recurrent neural language model. The difference between an auxiliary decoder and a standard RNNLM (for the target language) is in the aspect of training. The auxiliary decoder sharing parameters with the main NMT decoder is trained with source side information while the standard RNNLM is only trained on the target corpus. Considering the mismatch of training and testing situation for the auxiliary decoder, the performance on language modeling task of it can be worse than a standard RNNLM.

To demonstrate the similarity with a standard language model and verify our assumption about the performance of the auxiliary decoder, we evaluate the auxiliary decoder and several standard language models on the language model task. The standard language models include two n-gram models and a RNN based language model. A n-gram model is a statistical model that predicts the probability of the next word given the previous $n-1$ history words. We use a 2-gram and a 3-gram model with Kneser-Ney smoothing (Kneser and Ney, 1995). The two n-gram models are trained using the toolkit SRILM (Stolcke, 2002). The RNN based language model is a two-layer LSTM with both embedding and hidden dimensions 500. We trained the LSTM LMs using Pytorch. The optimization method is Adam with initial learning rate 0.001. The architecture and optimization settings of the LSTM LM are the same as the auxiliary decoder.

The perplexity results on four datasets by the two n-gram models, the LSTM-LM, and the auxiliary decoder are in Table 2. We can see that although the auxiliary decoder has worse perplexity than a standard LSTM language model, its performance are similar to a n-gram language model for most situations. This observation is consistent with our assumption.

Model	Acquis	EMEA	Koran	IT
2-gram	72.3	86.7	86.3	120.0
3-gram	38.1	44.6	61.4	46.7
LSTM-LM	19.8	19.2	19.6	30.9
Auxiliary Decoder	44.0	51.9	103.7	57.1

Table 2: Perplexities from different models on four test corpora from different domains.

5.2 Part of Speech

Different words have different dependencies on source context. It is very natural to assume that content words, such as nouns and verbs, tend to have higher dependencies on the source context, while function words like adpositions depend more on the target side.

Figure 4 is an example of the source context dependency measurement on a translated English sentence from Germen meaning *“after my study of electronics, I came here in 1954.”*[3]. We can see that content words such as *“study”*, *“electr@@”*, *“19@@”* and *“54”* have relatively high dependency scores. Meanwhile, functional words like *“of”*, “,”, and *“in”* have lower scores.[4]

[3] This sentence comes from subtitle dataset

[4] One might notice that it is not always true that all content words scores are high and all function words scores are low in this sentence. For example, word *“after”* has a very high score, and word (or sub-word) *“onics”* has very low score. The reason for the first case is that while predicting the first word, internal language model in decoder will always prefer the most common word in training data since there is no history. So even *“after”* is a functional word, the most of the information decoder needs to generate *“after”* comes from attention vector, which results into a very high score according to our metric. As to the second case, *“onics”* is a sub-word of *“electronics”*. Since the sub-word phrase (*“electro@@”*, *“onics”*) is relatively frequent and these two sub-words are highly unlikely to appear independently, the decoder can be confident to predict *“onics”* given previous prediction *“elec-*

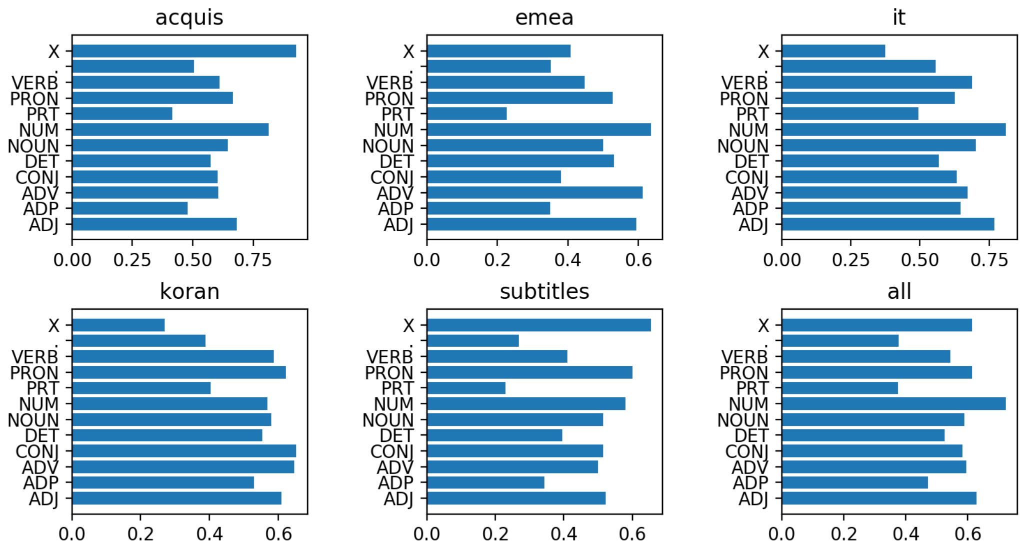


Figure 3: Scores for different categories of part of speech (POS).

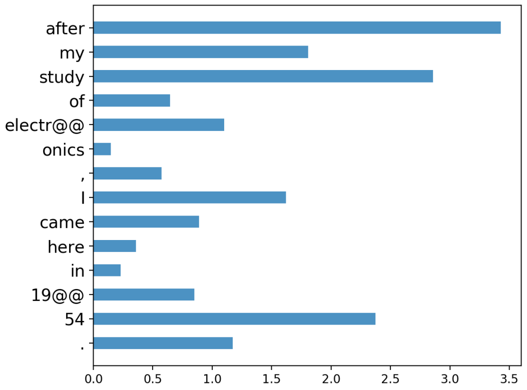


Figure 4: English translation and source context dependency of Germen sentence *"nach meinem Studium in Elektroni@@ k kam ich hier in 19@@ 54 ."*.

We then compare the source context dependency of translated words with part-of-speech (POS) tags. We calculate the average source context dependency score for each POS category over test sets from five corpora, shown in Figure 3. We can observe that although the distribution of scores are different among domains, they all have a similar tendency. Adpositions and particles have lower source context dependency than other categories, especially numbers. This observation is consist with our intuition.

Another interesting observation is that the average score of functional words in certain situations can be high. For example, determinators in EMEA dataset is even higher than nouns. There are two reasons for this observation. First, EMEA is a highly structured and repetitive corpus. The NMT models can generate nouns with little context information since noun phrases in this corpus are frequent. The decoder can easily determine the remaining words given the first word of a phrase. The second reason is that although functional words seems to rely more on decoder, some of them still need context information. For example, if a sentence contains the noun phrase *"an apple"*, the model will generate the correct determiner *"an"* rather than *"a"* from source information — the determiner *"a"* is highly dispreferred by the language model.

5.3 Domain Mismatch

Domain mismatch is a major challenge for NMT. Training an NMT model in one domain can make the decoder overfit that particular domain. Thus, during decoding the decoder can produce fluent but inadequate sentences on out-of-domain test data. Therefore, we wondered if domain mismatch can cause less source context dependency.

We calculate source context dependency scores under domain mismatch settings. Five NMT models were trained on five datasets shown in Table 1. We then apply them on five test datasets and calculate source context dependency scores, respectively. Next, means and variance of scores among test sets with different models are calculated.

The results are shown in Table 3 and Table 4. Domains of training data are in columns and do-

tro@@" with very limited source side information. These two cases are actually quite rare in our corpora, so we did not use them for POS tagging analysis.

Test Train	law	medical	it	koran	subtitles
all	2.594	2.831	2.283	2.418	2.372
law	**3.956**	3.695	3.463	3.382	3.818
medical	1.694	**2.186**	1.615	1.383	1.414
it	3.597	3.536	**4.312**	3.423	3.937
koran	1.965	1.765	2.024	**3.14**	1.93
subtitles	0.955	0.982	0.971	1.021	**1.489**

Table 3: Means of source context dependency scores on different test datasets translated by different models.

Test Train	law	medical	it	koran	subtitles
all	3.5	3.44	2.07	1.725	2.061
law	**15.06**	10.9	9.93	8.83	13.5
medical	3.74	**5.26**	2.94	1.89	2.8
it	8.61	8.13	**14.15**	8.46	11.3
koran	3.74	3.37	4.02	**7.87**	4.2
subtitles	0.676	0.619	0.66	0.571	**1.521**

Table 4: Variances of source context dependency scores on different test datasets translated by different models.

Test Train	law	medical	it	koran	subtitles
all	31.1	45.1	35.3	17.9	26.4
law	**31.1**	12.1	3.5	1.3	2.8
medical	3.9	**39.4**	2.0	0.6	1.4
it	1.9	6.5	**42.1**	1.8	3.9
koran	0.4	0.0	0.0	**15.9**	1.0
subtitles	7.0	9.3	9.2	9.0	**25.9**

Table 5: BLEU scores of source context dependency scores on different test datasets translated by different models, reported by Koehn and Knowles (2017).

mains of test data are in rows. ”all” in the Table 3 and Table 4 means the model was trained on a combination of the five datasets. Since we care more how one certain model behaves on test sets from different domains, we compare the scores along the rows. It is noticeable that all the five models have highest source context dependency scores when translating in-domain test data. Higher means indicate that in-domain models depend more on source information. Higher variances show that in-domain models are also better at learning differences among different word, because we expect a good model has more context dependency on content related words and less on history related words. This can be one of the reasons why NMT models often generate fluent but inadequate translations in domain mismatch settings.

We also list the BLEU score reported by Koehn and Knowles (2017) on the same task, shown as Table 5. We can see that inability of incorporating context information into the decoder can be a main reason for the failure in domain mismatch setting.

5.4 Sentence Length

The translation quality is sensitive to the lengths of the sentences. Moreover, for longer sentences, it is possible that a NMT model considers more history information rather than source context information. We want to know how sentence length affects source context dependency.

We calculate source context dependency for sentences with different lengths. Results are shown in Figure 5. We find that longer sentences have lower source context dependency, which is consistent with our hypothesis[5].

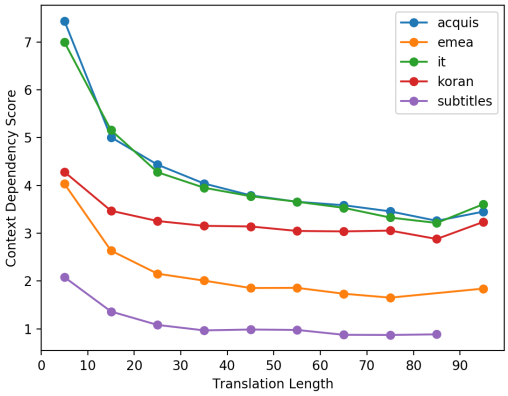


Figure 5: Source dependency scores with length of sentence

However, we detect a different tendency compared with the analysis by Koehn and Knowles (2017) which show lower translation quality for longer sentences. However, source context dependency is not the only factor that determines translation quality. When the length of translation increases, history information from the language model is increasingly informative (and hence predictive).

[5] One can notice that there are some small fluctuations in sentence length from 70 to 90. This can be caused by a small percentage of sentences in that length range (70-80: $\sim 4\%$, 80-90: $\sim 3\%$).

6 Conclusion and Future Work

In this paper, we proposed a measurement of source context dependency in neural machine translation models. With our measurement, we analyzed source context dependency with different POS tags, domains and sentence lengths. From the analysis, we can see our measurement is a good estimation of source context dependency.

In the future, we plan to extend our research in two directions. One is to investigate the relationship between source context dependency and word level translation quality, so that we can immediately detect when the system goes off track. The other is to improve the performance of NMT models. Since our measurement is differentiable, we can use it as an auxiliary term of the training objective function.

Acknowledgement

We would like to appreciate Center of Language and Speech Processing, Johns Hopkins University for providing hardwares for our experiments. We also appreciate valuable suggestions and advice from people in Johns Hopkins University machine translation group.

This work was partially supported by the IARPA MATERIAL program.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. (2017a). What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Belinkov, Y., Màrquez, L., Sajjad, H., Durrani, N., Dalvi, F., and Glass, J. (2017b). Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10. Asian Federation of Natural Language Processing.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning Workshop.*

Cohn, T., Hoang, C. D. V., Vymolova, E., Yao, K., Dyer, C., and Haffari, G. (2016). Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885. Association for Computational Linguistics.

Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., and Vogel, S. (2017). Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 142–151. Asian Federation of Natural Language Processing.

Ding, Y., Liu, Y., Luan, H., and Sun, M. (2017). Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Ghader, H. and Monz, C. (2017). What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39. Asian Federation of Natural Language Processing.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acous-*

tics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, volume 1, pages 181–184. IEEE.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Liu, L., Utiyama, M., Finch, A. M., and Sumita, E. (2016). Neural machine translation with supervised attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3093–3102.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shi, X., Knight, K., and Yuret, D. (2016). Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282, Austin, Texas. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Stolcke, A. (2002). Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180.

Tu, Z., Liu, Y., Lu, Z., Liu, X., and Li, H. (2017). Context gates for neural machine translation. *TACL*, 5:87–99.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zheng, Z., Zhou, H., Huang, S., Mou, L., Dai, X., Chen, J., and Tu, Z. (2018). Modeling past and future for neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:145–157.

Gist MT Users: A Snapshot of the Use and Users of One Online MT Tool

Mary Nurminen
University of Tampere
Kalevantie 4
FI-33100 Tampere, Finland
mary.nurminen@uta.fi

Niko Papula
Multilizer
Kimmeltie 3
FI-02110 Espoo, Finland
niko.papula@multilizer.com

Abstract

This study analyzes usage statistics and the results of an end-user survey to compile a snapshot of the current use and users of one online machine translation (MT) tool, Multilizer's PDF Translator[1]. The results reveal that the tool is used predominantly for assimilation purposes and that respondents use MT often. People use the tool to translate texts from different areas of life, including work, study and leisure. Of these, the study area is currently the most prevalent. The results also reveal a tendency for users to machine translate documents that are in languages they have some understanding of, rather than texts they do not understand at all. The findings imply that gist MT is becoming a part of people's everyday lives and that perhaps people use gist MT in a different way than they use publishing-level translations.

1. Introduction

Online machine translation (MT) tools have been in use for almost 25 years and people are finding numerous ways to integrate MT into the processes of their everyday lives. However, although research on professional translators' use of MT has grown rapidly, the literature on all other users of MT remains limited. This paper aims to contribute to that limited body of research with a study on the users of one online MT tool, Multilizer's PDF Translator.

1.1. Purpose of the Study

Our study focuses on users of MT for assimilation, or scenarios in which people use raw, unedited machine translated text for some other purpose than editing it for publication. Because users most often want just a basic understanding of the information (or gist) of the text, we term them *gist MT users.* We also use it because it is shorter than the term *users of MT for assimilation*; however, we use the two terms interchangeably.

The overall purpose of the study is to present a snapshot of the use and users of one online MT tool. Our questions concern who is using MT, where these users are, how they are using it, when they are using it, and in what areas of life they are using it.

We had several motivations in doing the study. First, because online MT is in such wide use today, we can assume that the number of gist users is much larger than the number of professional translator users. Yet the latter group has been studied far more than gist MT users. We believed it was time to put some focus on other user groups and we hoped to contribute to that with this study. Second, our literature review revealed only one gist MT user survey conducted in the past 10 years. We felt it was time to conduct another one. Finally, this analysis will serve as a basis for a second study we are planning, a qualitative study that will probe more deeply into the specific ways people are using gist MT.

1.2. Related Work

The pioneer study of MT users, by Henisz-Dostert in 1979, was also the first study on gist MT users. In the 40 years since it was published, a relatively small number of articles have been written about gist MT users. These studies can be grouped into two categories: experimental studies on *potential* users of gist MT and survey studies on *actual* gist MT users.

In the experimental studies, groups of potential gist MT users were asked to evaluate specific aspects of MT or the use of MT. Fuji et al. (2001) tested user success with machine translat-

[1] pdf.multilizer.com

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 199–208
Alacant, Spain, May 2018.

ed texts, measured through reading comprehension, against users' *impressions* of comprehensibility and awkwardness. Gaspari (2006) had users evaluate their *confidence* in understanding raw MT. Bowker and Ehgoetz (2007), Bowker (2009), Bowker and Buitrago (2015) and Castilhjo and O'Brien (2017) had users evaluate the *acceptability* of raw MT. They often had users compare preference or acceptability of raw MT, post-edited MT, and human translation. Gaspari (2004), Stewart et al. (2010), and Doherty and O'Brien (2012) had users evaluate raw MT against traditional usability criteria. Finally, Doherty and O'Brien (2014) used eye tracking to measure MT output usability.

The studies on actual gist MT users include research on market or usage reports, end-user surveys, or a combination of the two. A small number of these were agnostic to MT systems, focusing on groups who were using any number of the systems available at the time. A larger group of research focuses on users of one specific system. A limitation of this second group of studies is that they describe only a specific type of user and therefore the results cannot be considered representative of all MT users. However, they do contribute information on those users and, seen collectively, help to paint an overall picture of gist MT usage.

The first studies on users of various MT systems were sponsored by the International Association for Machine Translation (IAMT) in 1993 and 1995. These studies used participants they recruited through the manufacturers of MT systems or through the AMTA website. Although they focused mainly on professional translators, who used MT for dissemination, they did include a small amount of data on gist MT users in the form of eight testimonials (Lawson and Vasconcellos, 1993). The Asia-Pacific Association for Machine Translation (AAMT) recruited participants for a series of studies in 2003-2005 through their website, so the user group represented was again not specific to any one tool. These surveys focused much more on gist MT users, indicating that "the main use of machine translation" was assimilation (Yamada et al. 2005, p. 58). The final study that was not dependent on any one MT tool was that carried out by Gaspari in 2007. The survey, conducted at several UK university campuses, used students as informants and covered user demographics, experience with computers and MT, languages translated, use of MT for assimilation vs. dissemination, genres translated, and user evaluations of MT.

The first study that focused on users of a specific system was the study on the users of the Georgetown MT system cited earlier in this article (Henisz-Dostert, 1979). It used a survey, although that survey was administered almost entirely through face-to-face interviews. It provided a rich and multifaceted description of the users, how they used the system, and their experience regarding usefulness, speed, and quality. The study also included a few interesting questions on how users experience cognitive processes, which subsequent surveys have not touched on. These included questions such as "If the style of the MT is awkward, can you correct it mentally?" and "Do you get 'used to' reading MT?" (Henisz-Dostert 1979, p. 193) The only other study we are aware of that address cognitive processes was Doherty and O'Brien's (2014) previously mentioned eye tracking study.

The next study of the users of one system was conducted in Japan by Hoshino (1995), focusing on users of the Korya Eiwa ("It's Nice! English–Japanese") consumer desktop system. The survey was comprehensive, covering user demographics, genres and subject matters translated, users' fluency in English, experience with MT, purpose, motivations, and expectations for MT. Flanagan's (1996) paper described the usage of CompuServe's online MT service as well as users' reactions to it. Another online service, AltaVista Translation with Systran, was the focus of a study by Yang and Lange (2003). The study included both an analysis of usage and feedback data in the form of 5,005 e-mails received in 1998.

A few studies have been conducted on company-internal MT systems and their users. Smith (2003) analyzed PriceWaterhouseCooper's intranet-based MT system and its users. This was perhaps the first study on a system that supports a large number of language pairs, 37 in total. It described how people used the system, their reactions to it, and factors that affected users' satisfaction with the system. Another company-internal study was conducted by Nuutila (2005), who reported on a survey conducted with users of Nokia's Roughlate MT service.

The latest user study we are aware of was a study by Burgett (2015) on the users of Intel's machine-translated support content. This study asked users to perform usability tests while working with Intel's machine translated content.

2. Multilizer's PDF Translator

The tool in our study, PDF Translator, is an online MT tool that translates full documents that are in either PDF or Word format. A user submits a document, then the tool extracts the texts, puts them through machine translation, rebuilds the document with the original pictures in place, and returns it to the user in the requested language. PDF Translator utilizes the MT engines of Microsoft, Google and PROMT to perform the translations. Due to the proprietariness of the engines and the dynamic nature of MT development, we do not have information on the exact type of MT (rule-based, statistical or neural) used for each language pair during the time of the study.

PDF Translator is meant for any type of document that people want to have translated, so it is not trained for specific genres or subject matters. Two versions are available, a desktop and an online version. The desktop version, which was developed first, is downloaded onto the user's computer and used from there. Its user interface is available in 14 languages. Users can translate up to 3 pages at a time for a total of 15 pages for free. PDF Translator offers three levels of paid licenses: Standard, Pro and Business, and after initial purchase of a license, additional pages can be purchased in batches. The desktop version supports 47 source languages and 39 target languages. The newer online version has been in use since 2016 and it is currently available through an English, Spanish or Chinese user interface. Users can translate a small amount of text (one page) free of charge and thereafter they can purchase packages of translation (10, 50, 100, etc. pages). The online version supports translation between 42 languages.

2.1. MT for PDF and DOC Documents

One important aspect of PDF Translator is that it translates entire documents instead of pieces of text typed or copy/pasted into a text field. This holds several implications for our study and the types of users it addresses. First, it excludes incidences when people enter only one or two words, essentially using MT as a bilingual dictionary. Previous studies have found this to constitute a large portion of MT use. For example, Yang and Lange reported that "more than 50% of translations are of one- or two-word phrases" (Yang and Lange, 2003, p. 199) and Gaspari was led to devote a whole section of his PhD to "(Mis-) Using Free Web-based MT Services as Online Dictionaries" (Gaspari, 2007, p. 108). Another implication of translating whole documents is that the materials people submit for translation tend to be well-formed and written, published documents instead of more informal texts such as chat messages or personal correspondence. This can influence the areas of life where people use MT – for work and study or in their free time. A final implication is that, due to the very nature of PDF as a publication instead of an editing format, users are far more likely to be gist MT users than to be people who want to edit the material for publication. All of these factors contribute to profiling a specific type of user and need to be kept in mind when reading this study.

3. Materials and Methods

Our goal was to capture a snapshot of the use and users of PDF Translator in a short, specific point of time. We chose a four-month period, November 1, 2017 through February 28, 2018, and collected two types of data from the period for analysis. We collected log files from both the desktop and the online systems, and we conducted an online end-user survey with users of the desktop system.

Our first batch of data consisted of the logs from the desktop and online versions of PDF Translator. We used the logs to examine the times that submissions for translation were made, the places they were made from, and the source and target languages involved.

The end-user survey was short, consisting of eight questions in three categories:

Category	Questions
Basic demographics	1. What is your gender? 2. What is your age? 3. What language are you most proficient in? 4. What is the highest degree or level of school you have completed?
Frequency of use of MT tools	5. How often do you use tools that automatically translate texts, similarly to PDF Translator or Google Translate?
Questions on the specific document submitted for translation	6. Why did you want to translate the document? 7. Did you need the document for work, study, or leisure purposes? 8. How well did you understand the language of the original written document?

Table 1. Survey questions.

The reason for the brevity of the survey was that, in keeping with the idea of a snapshot, our focus was on quantity more than quality. The survey needed to be short enough so that a large number of people would be willing to answer it.

Besides keeping the survey short, we used other strategies to encourage users to respond. We offered all respondents the chance to participate in a drawing for five small prizes: 100 pages of free translation through PDF Translator. We also named it *3-minute Survey for Users of PDF Translator* under the assumption that precise information on how long it would take to answer the survey would encourage people decide to devote time to it. The average response time was, in fact, three minutes.

Due to limited resourcing, we had to make decisions on what languages to offer the survey in. We decided to offer the survey to users of the most popular 6 of the 14 languages the desktop version of PDF Translator is available in: English, Spanish, Portuguese, French, Russian and Indonesian.

An invitation to answer the survey was offered to users after they had submitted a document into PDF Translator and received the translation back. It was offered to everyone who submitted a document during that period, meaning that both heavy users of the tool and first-timers could answer.

4. Discussion

Besides the log files, our data included 1,579 responses to the three-minute survey. The response distribution by language survey is displayed in the following table.

Language survey	Number of responses
Spanish	652
Portuguese	283
French	211
Russian	188
English	147
Indonesian	98
Total	**1579**

Table 2. Survey response distribution.

PDF Translator has a large customer base in Spanish-speaking countries and this is reflected in the high number of responses to the Spanish survey. The placement of the other language surveys correlate roughly with our statistics on the countries and target languages with the most traffic during the study period. While compiling responses, we noticed that a large number of responses to the English survey (49 responses, comprising 25% of all responses), were from people who marked Indonesian as their most proficient language. We did not observe a similar phenomenon in any other language survey. We decided to move these 49 responses from the English survey to the Indonesian one. The previous table reflects the numbers *after* that change.

4.1. Locations and Languages

PDF Translator is used widely across the world. Our logs indicated that requests for translation during the study period came from 181 countries and territories. The tool's large customer base in Spanish-speaking countries is reflected in the list of the countries with the most traffic, with 10 of the top 20 spots being occupied by those countries. Other countries in the top 20 include Brazil, Indonesia, Poland, Germany, Italy, Russia, Turkey, France, Ukraine, and Portugal.

English was the most popular source language, with 85% of all documents translated during the study period being originally in English. The next languages on the list of source languages included German, Spanish, French, Portuguese, Italian, Russian, Polish, Dutch and Indonesian. Spanish led the list of the most popular target languages, followed by Portuguese, English, French, Russian, Indonesian, German, Polish, Italian and Turkish.

The top language pair of English–Spanish comprised 47% of all requests. This was expected, considering PDF Translator's customer base. Also, this language pair has appeared at the top of lists in survey and market studies for a long time, including those by Yang and Lange (2003), Smith (2003), Gaspari and Hutchins (2007) and Turovsky (2016).

Indonesian's position near the top of the language lists was interesting. The past ten years have seen a major expansion in the language palette of online MT tools (e.g. Turovsky, 2016). It appears that this expansion is beginning to produce results and new language pairs are emerging at the top of the lists of the most-translated languages. For example, Google's recent reports on the most-translated languages include the ones that have appeared at the top of these lists for years—Spanish, Russian and Portuguese—but also relative newcomers to online MT, such as Arabic and Indonesian (Turovsky, 2016). Indonesian proved to be an interesting and different market in other areas of our study as well.

4.2. Survey Participant Demographics

The overall gender demographic of survey participants showed males comprising 68% of responses, females 32%, and the group of *other*, 3%. Small differences surfaced when comparing the results of different language areas. In the Portuguese, Spanish and English surveys, males made up 61–68% of responses while in the French and Russian surveys, 82–83% of respondents were male. Indonesia was the only country in which female respondents outnumbered male (54% and 46%, respectively). The high proportion of men in most of the language surveys seems to be typical in studies of technological systems.

The age distribution shown in survey answers was also typical of that shown in technology studies, with the 19–29 age group providing the largest number of responses, 46% altogether. Similarly to the results of the gender demographic, the age demographic also contained differences in the results from different language surveys, as is shown in the following figure.

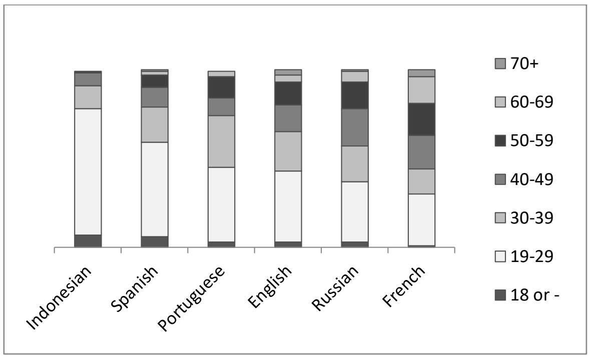


Figure 1. Age distribution of respondents in different language surveys.

Indonesian again displayed a different profile from the other surveys. In that survey, the 19-29 age group made up 71% of the total, 18 percentage points higher than the next (Spanish) survey. The French and Russian surveys were again at the opposite end of the scale, with a much more even distribution of ages. Another interesting point was that the French-speaking older respondents seem to be the most active. Whereas in most of the language surveys, the two highest age groups comprised 3–7% of respondents, in the French survey this group comprised 19% of all respondents. Although the total overall number of answers in the highest age groups, 60–69 and 70 or older, was small (68 and 19 responses respectively), it was good to note that people in these age groups are also using MT actively.

The following figure shows how much of each respondent age group was comprised of female, male and other genders.

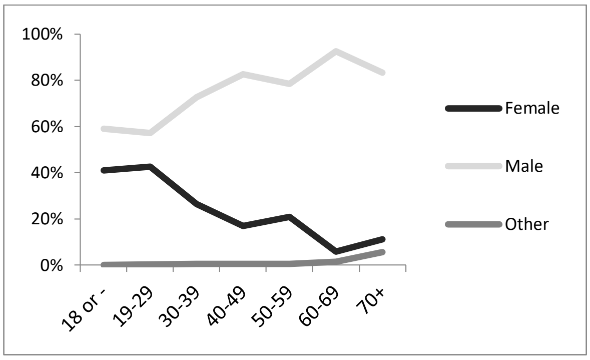


Figure 2. Percentage distribution of survey respondents by gender and age group.

The chart shows that in the younger age groups, a smaller gap exists between the male and female composition of the respondents. This gap grows and peaks in the 60–69 age group before becoming smaller again in the 70 or older group. A somewhat even number of people identify as some other gender throughout all age groups, although the relatively small overall number of respondents in the 70 or older group resulted in the *other* group comprising a higher percentage of the whole.

The highest degree or level of school reported by respondents is shown in the following table.

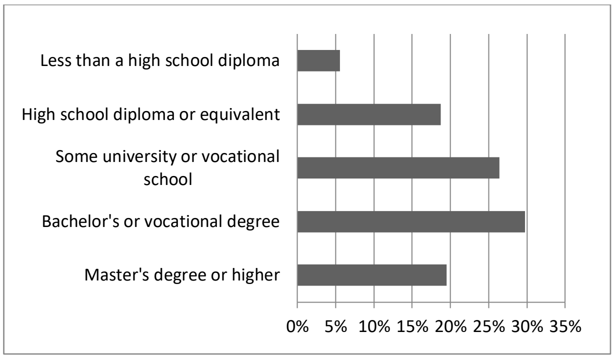


Figure 3. Highest level of education of respondents.

Respondents appear to be fairly highly educated, with the largest group being comprised of people who already have a vocational or bachelor's degree. In comparing the different language surveys, the French and Russian surveys once again stood out in that they had high percentages of respondents who held a master's degree or higher. In fact, the educational level with the most responses in both surveys was a master's degree.

4.3. Frequency of MT Use

As has been noted by Gaspari (2007) and others, a self-administered survey such as this one can result in responses being given by people who are relatively more active in the technology area than the general user population. This factor needs to be considered when examining the responses to our survey question on how often respondents use MT, which are displayed in the following chart.

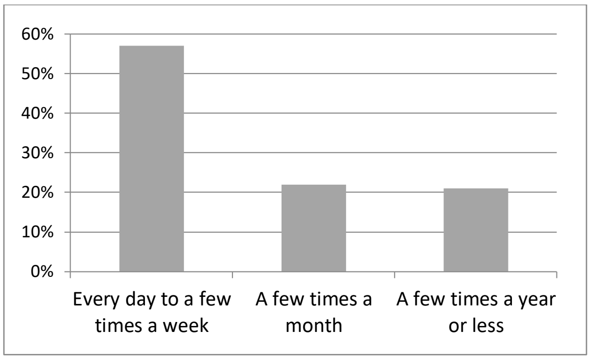


Figure 4. How often respondents report using machine translation.

These results indicate that a majority of the overall respondents of this survey tend to use MT on a very regular basis. In comparing to previous studies that have asked this question, Yamada et al. (2005) reported that only 13–18% of users used MT as frequently in 2003–2005. However, Nuutila's (2005) study showed that 63% of Nokia's in-house Roughlate system users reported using the system several times a day or at least every week.

The next chart shows a breakdown of reported frequency of use by age group.

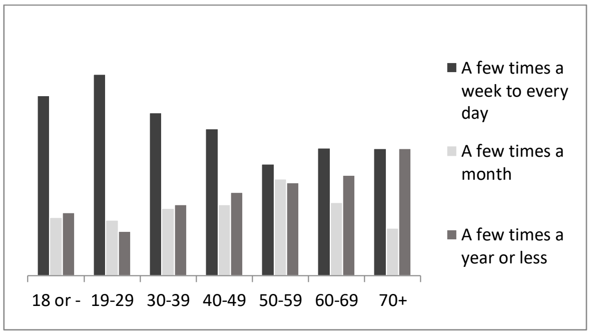


Figure 5. Frequency of MT use by different age groups.

As is shown here, the younger age groups, 18 or under and 19–29, showed a stronger tendency to use MT very frequently than respondents in older age groups. In fact, the level of very frequent use for the 19–29 group was remarkably high, 67%.

4.4. Purpose: Assimilation, Dissemination, or Something Else

To explore users' purposes for using MT, the area of life they were using MT in, and their proficiency in the languages involved, the survey included three questions that asked specifically about the document the respondent had submitted for translation right before being invited to take the survey. The first of these questions concerned whether users were using the submitted document for assimilation, dissemination, or some other purpose. Although we could assume that people translating whole documents (many of them PDFs) are mainly using MT for assimilation purposes, we wanted to verify this. We started with the questions and answer choices used by Gaspari in his survey of students (Gaspari, 2007, p. 102–103) and edited them a bit. The following table shows the overall responses.

Why did you translate the document?	**% of responses**
I wanted to understand it myself. (assimilation)	58%
I wanted to verify that I understood it myself. (assimilation)	18%
I wanted to translate it into my own language so that someone else can understand it. (assimilation for other person)	14%
I wanted to translate it from my language into another language so that someone else can understand it. (dissemination)	6%
Some other reason (please specify).	4%

Table 3. Purpose of translating the document submitted for translation.

Combining the first and second answers gives an overall view to assimilation and shows that a majority of respondents, 76%, are indeed using the machine-translated documents for their own assimilation. However, the second answer taken alone is also interesting in that it shows that people are using MT for understanding documents, but also for verifying their understanding. Another interesting point arises when comparing the responses of different language surveys. In Indonesia, 25% of respondents reported that they translated the document into their own language so that someone else could understand it. In other language surveys, the rate was only 10–16%. Combining this with the relatively young demographics of that market, could this reflect an effort by younger people to help their technologically more reticent elders?

4.5. Area of Life Where MT was Used

The second of the questions we asked about the document the respondent had translated regarded the area of life that the document concerned: work, study, or leisure. We allowed respondents to select more than one choice in case the document was used in various areas. However, only 11% chose more than one area. The following figure displays the overall compiled results of responses to the question.

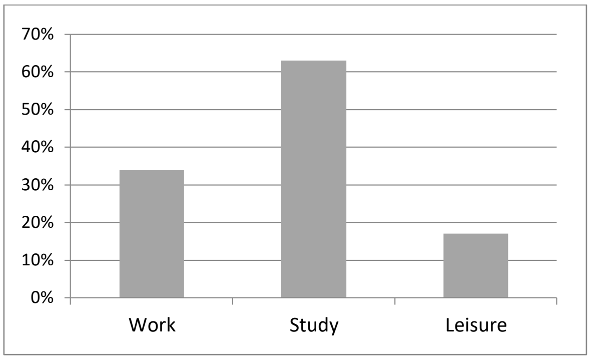


Figure 6. Percentage of respondents who listed work, study, and/or leisure as the purpose of the document they translated.

Overall, 63% of the respondents reported that at least one of the areas of life in which they needed the translated document was study. This would indicate that, at least for the type of user who is translating whole documents (and willing to answer surveys), MT is being used widely for learning purposes.

This figure shows the responses by age group.

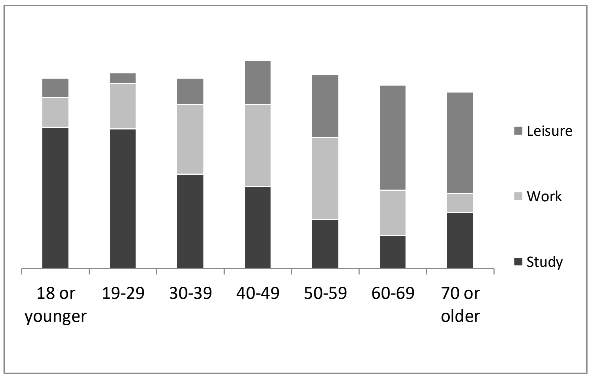


Figure 7. Reported area of life where machine translated document was used, by age group.

This distribution seems logical and perhaps expected, with users in the younger age groups showing a relatively strong emphasis on study. It is interesting that the study category increased again in the 70 or older age group, though it should be kept in mind that the number of responses in that group was small (19), and that respondents who are active users of MT, and are willing to answer surveys, might well also have a keen interest in self-study.

Two factors seem to have contributed to making *study* the top area reported. First, a relatively high number of responses to the survey came from the 19–29 age group. Second, responses from the Spanish and Portuguese surveys were also relatively high, and as can be seen in the following table, both of those languages showed very high scores for *study*.

Survey	**Work**	**Study**	**Leisure**
Indonesian	19%	88%	4%
Portuguese	30%	73%	15%
Spanish	31%	75%	9%
English	46%	49%	19%
French	43%	34%	39%
Russian	44%	36%	31%

Table 4. Percentage of respondents who listed work, study, and/or leisure as the purpose of the document they translated in different surveys.

In this table, the English, French, and Russian answers reflect more of an emphasis on work. In fact, in the French and Russian results, work surpasses study as the area of life the translated document concerned. As discussed earlier in this article, the demographics of the French and Russian respondents were somewhat different than those of the other language surveys. These differences seem to indicate that the way MT is used can be different in different groups or geographical areas.

In addition to analyzing the responses to our survey, we also used the log files to analyze the day of the week and time of day when people requested translations. We converted all log time stamps to local times. The results of that analysis are presented in the following figure, which shows usage levels for the seven days of the week and hour-by-hour. Each of the seven lines in the graph represents one day of the week. Black lines were used for Monday–Thursday and gray for Friday–Sunday.

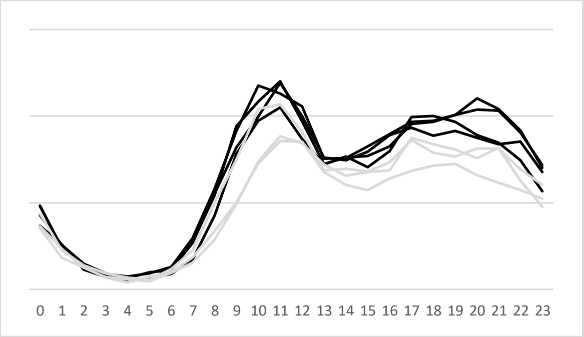


Figure 8. Usage by day of the week and time of day. The black lines are Monday–Thursday and gray lines Friday–Sunday.

Although all lines demonstrate activity during the evening hours, a clearly higher activity level emerges on Monday–Thursday than on Friday–Sunday. This analysis seemed to support the result that study and work are areas of life where users of the tool request translations, more than leisure.

It should be noted that these results reflect the situation for one tool at a specific point in time. As the technology and users mature, the overall emphasis could shift from study to other areas of life. Another point of consideration is that our results do not provide details on the level of education users are at when they use MT for study. It could be anything from grade school through Ph.D. research. The results also do not tell us exactly how users are using the machine-translated information: to help them in language production, for self-study, or to read scientific articles in a language they do not know. These questions should be addressed in future studies.

4.6. Understanding of Source Language

The third question in the survey related to the document that each respondent had submitted for translation was the following: *How well do you understand the language of the original written document (before it was translated)?* The possible answers were *Very well, Well, A little* and *Not at all.*

In the overall results, 51% of people reported that they understood *a little* of the source text and 33% said they understood the source text *well* or *very well*. By contrast, only 17% labeled their understanding as *not at all*. A few differences emerged when comparing the results of different language surveys. The Portuguese and English surveys had the highest percentage of people answering that their understanding of the source language was *not at all* (23% in English, 36% in Portuguese). In all other languages, 15% or fewer reported having no understanding.

As participants reported using PDF Translator for a variety of purposes, including dissemination, we conducted a separate analysis of people who specifically used it for assimilation, or gist users. For that analysis, we used only the answers of respondents who said their reason for translating the text was either that they wanted to understand it themselves or that they wanted to verify that they understood it themselves. As is shown in the following chart, a large majority of this specific group displayed at least a basic understanding of the source texts they translated. This result was similar to the overall results.

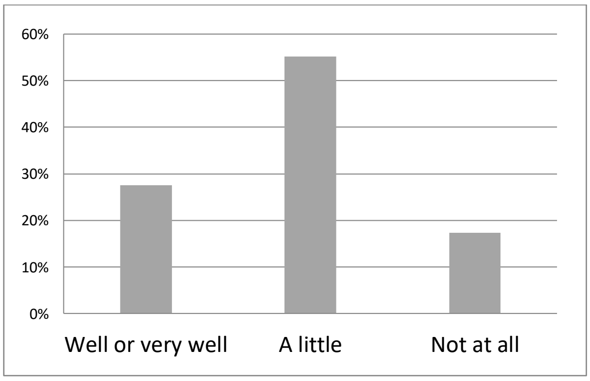


Figure 9. Reported understanding of the source text of the document submitted for translation by gist users.

The responses showed that in general, users of this tool often seem to translate texts that are in languages in which they already have some proficiency. Some previous survey studies have asked about users' competence in the source language, including Henisz-Dostert (1979), Hoshino (1995) and Yamada et al. (2005). A few other studies have uncovered indications of a link between knowledge of the source language and use of MT (Nurminen, 2016; Ogura et al., 2004).

Of course, people who are translating documents they already have some understanding of might also be simply testing PDF Translator or MT. Although we offered such people the choice to answer that their reason for translating the document was *some other reason*, some people may have instead indicated that their purpose was their own understanding and are therefore included in the assimilation group. In spite of this, there did appear to be a tendency to translate documents that respondents already had some understanding of, and this tendency has some interesting implications. First, this might be one reason why, even with the onslaught of new language pairs available in online MT tools, the same European-based languages still tend to dominate the lists of the most translated languages. Because they are being taught widely in schools, these are languages in which people may have low-to-medium (although existing) competence.

Second, this could reflect a tendency to use MT with caution. Users want to be able to compare the machine translated text to the original so that they can evaluate the general level of MT output. This tendency might decrease in the future, as MT improves and users' trust in its quality increases.

Third, this raises a question that was asked in Henisz-Dostert's survey (1979): how do people find the texts they have machine translated? Do they need to have a basic understanding of the text (or even the title) to be able to make the decision to machine translate it? This would restrict the texts and the languages involved in gist MT use.

Finally, the phenomenon raises a question about how people use MT. Is MT in these cases being used as some type of language tool, which users can combine with other resources, such as their limited competence in the source language or their familiarity with the topic of the text, to gain understanding of a text in another language? If so, does this mean that the way people use gist MT (in raw or possibly also lightly post-edited form) is inherently different than the way they use publishing-level translations? Perhaps we need to begin seeing gist MT as a different translatorial activity than human translation, and to stop comparing them to each other.

5. Conclusions

This study provided a snapshot of the use and users of a specific type of gist MT tool. It presented a picture of who is using PDF Translator, where these users are, how they are using it, when they are using it, and in what areas of life they are using it.

The study confirmed some findings of previous studies. English continues to be the most-translated language and English-Spanish the most commonly translated language pair. However, it also showed that new languages such as Indonesian are beginning to appear at the top of lists of languages involved in MT. The demographics of the survey respondents indicate that, even though overall statistics reflect a bias toward young and male users, which is commonly found in technology studies, differences do emerge in the demographics of different language areas.

A few new tendencies that deserve further study surfaced also. First, gist MT users who translate whole documents seem to use MT often, multiple times a week. Second, the importance of MT in the area of study, at least for the current users of PDF Translator, was a noteworthy result. Finally, users' tendency to machine translate texts in a language that they have some level of proficiency in was a new finding.

Our study shares a limitation with a number of similar surveys in that it studied the users of only one tool and therefore cannot be considered representative of any larger or more general population of users. A second limitation was the use of a self-administered survey, which can lead to a disproportionately enthusiastic picture of MT users. A more random sampling of respondents could produce different results.

The study nevertheless contributes to the small body of literature on gist MT users. The main contribution is that that users' competence in the source language seems to play some role in their use of MT. Users' reports on having some level of proficiency in the source language of the document they translated, plus the tendency some users have to use MT not only for assimilation but also for verifying their understanding of documents, lead to questions of exactly how people are using gist MT. Is it comparable to their use of human translation, or do they use MT in very different ways?

Further studies on how people are using MT in their studies would be called for. We would also like to see new studies that focus on general populations of gist MT users, instead of the users of one tool. However, the most urgent need we envision right now is for deep, qualitative data on exactly how people use gist MT. After the first study in 1979, very little insight has been gained as to how people have integrated MT into their daily lives, what types of processes they use, and the cognitive processes they rely on to extract meaning from imperfect language. As the quality of MT improves and more uses are found for MT in its raw form, the already-pressing importance of this type of data will increase.

References

Bowker, Lynne, and Jairo Buitrago Ciro. 2015. Investigating the usefulness of machine translation for newcomers at the public library. *Translation and Interpreting Studies* 10 (2): 165-186.

Bowker, Lynne, and Melissa Ehgoetz. 2007. Exploring user acceptance of machine translation output: A recipient evaluation. In *Across boundaries: International perspectives on translation studies.* Eds. Dorothy Kenny & Kyongjoo Ryou, 209-224. Newcastle, UK: Cambridge Scholars Publishing.

Bowker, Lynne. 2009. Can machine translation meet the needs of official language minority communities in Canada? A recipient evaluation. *Linguistica Antverpiensia* 8: 123-55.

Castilho, Sheila, and Sharon O'Brien. 2017. Acceptability of machine-translated content: A multi-language evaluation by translators and end-users. *Linguistica Antverpiensia* 16: 120-136.

Burgett, Will. 2015. Unmoderated remote usability testing of machine translation content. *TAUS Review of Language Business and Technology* IV.

Doherty, Stephen, and Sharon O'Brien. 2014. Assessing the usability of raw machine translated output: A user-centered study using eye tracking. *Intl. Journal of Human-Computer Interaction* 30: 40-51.

—— 2012. A user-based usability assessment of raw machine translated technical instructions. 10th Conference of the Association for Machine Translation in the Americas, San Diego, California, USA.

Fuji, M., N. Hatanaka, E. Ito, S. Kamei, H. Kumai, T. Sukehiro, T. Yoshimi, and H. Isahara. 2001. Evaluation method for determining groups of users who find MT “useful”. MT Summit VIII, 2001, Santiago de Compostela, Spain.

Flanagan, Mary. 1996. Two years online: Experiences, challenges and trends. Proceedings of the 2nd Conference of the Association for Machine Translation in the Americas "Expanding MT Horizons", Montreal, Canada.

Gaspari, Federico. 2007. The role of Online MT in Webpage Translation. Ph.D. thesis, University of Manchester.

——2006. The added value of free online MT services: Confidence boosters for linguistically-challenged internet users, a case study for the language pair Italian-English. 7th Conference of the Association for Machine Translation of the Americas, Cambridge, Massachusetts, USA.

—— 2004. Online MT services and real users' needs: An empirical usability evaluation. 6th Conference of the Association for Machine Translation in the Americas, AMTA 2014, Washington, DC, USA

Henisz-Dostert, Bozena. 1979. Users' evaluation of machine translation. *Machine Translation,* ed. Werner Winter, 149-244. The Hague: Mouton Publishers.

Hoshino, Sadao. 1995. Survey report: Korya eiwa: 100-dollar commodity has expanded the market. *Newsletter of the International Association for Machine Translation* (12).

Lawson, Veronica, and Muriel Vasconcellos. 1993. Forty ways to skin a cat: Users report on machine translation. Machine Translation Today: Translating and the Computer 15, London, UK.

Nuutila, Pertti. 2005. Rough Machine Translation in the Communication Process. Licentiate thesis, University of Tampere.

Nurminen, Mary. 2016. Machine translation-mediated interviewing in qualitative research: A pilot project. *New Horizons in Translation Research and Education* 4.

Ogura, Kentaro, Yoshihiko Hayashi, Saeko Nomura, and Toru Ishida. 2004. User adaptation in MT-mediated communication. The 1st International Joint Conference on Natural Language Processing (IJCNLP-04), Hainan Island, China.

Smith, Ross. 2003. Overview of PwC/Sytranet on-line MT facility. The 25th International Conference on Translating and the Computer, London.

Stewart, Osamuyimen, David Lubensky, Scott Macdonald, and Julie Marcotte. 2010. Using machine translation for the localization of electronic support content: Evaluating end-user satisfaction. Denver, CO, USA.

Turovsky, Barak. April 28, 2016. *Ten years of Google Translate* [Blog post]. Retrieved from https://blog.google/products/translate/ten-years-of-google-translate/.

Yamada, Setsuo, Syuuji Kodama, Taeko Matsuoka, Hiroshi Araki, Yoshiaki Murakami, Osamu Takano, and Yoshiyuki Sakamoto. 2005. A report on the machine translation market in Japan. The 10th Machine Translation Summit, Phuket, Thailand.

Yang, Jin, and Elke Lange. 2003. Going live on the internet. In *Computers and translation: A translator's guide,* ed. Harold Somers, 191-210. Amsterdam/Philadelphia: John Benjamins.

Letting a Neural Network Decide Which Machine Translation System to Use for Black-Box Fuzzy-Match Repair

John E. Ortega
Universitat d'Alacant
E-03071, Alacant, Spain
jeo10@alu.ua.es

Weiyi Lu
New York University, 60 5th Avenue
New York, New York 10011, USA
weiyi.lu@nyu.edu

Adam Meyers
New York University, 60 5th Avenue
New York, New York 10011, USA
meyers@cs.nyu.edu

Kyunghyun Cho
New York University, 60 5th Avenue
New York, New York 10011, USA
kyunghyun.cho@nyu.edu

Abstract

While systems using the Neural Network-based Machine Translation (NMT) paradigm achieve the highest scores on recent shared tasks, phrase-based (PBMT) systems, rule-based (RBMT) systems and other systems may get better results for individual examples. Therefore, combined systems should achieve the best results for MT, particularly if the system combination method can take advantage of the strengths of each paradigm. In this paper, we describe a system that predicts whether a NMT, PBMT or RBMT will get the best Spanish translation result for a particular English sentence in DGT-TM 2016[1]. Then we use fuzzy-match repair (FMR) as a mechanism to show that the combined system outperforms individual systems in a black-box machine translation setting.

1 Introduction

Natural Language Processing (NLP) systems designed to do the same task often belong to different methodological paradigms. At any time in history, the best-scoring systems may tend to come from a particular paradigm. For example, in Machine Translation (MT), the current dominant paradigm is Neural Network-based MT (NMT). The previously dominant paradigm was Phrase Based MT (PBMT), and so on. When comparing MT results for different types of input, systems from certain paradigms perform better on certain types of input and vice versa (Bentivogli et al., 2016). In some cases NMT suffers more than other paradigms (Koehn and Knowles, 2017). Thus, it may be premature to completely abandon "old" methods in favor of "new" ones.

Newer methods, especially NMT, tend to achieve higher BLEU scores than previous methods including PBMT and Rule-based MT (RBMT) systems. However, professional translators and users of computer-assisted translation (CAT) tools seem to prefer PBMT output for particular sentences (Arenas, 2013). Many recent systems (e.g., participants in WMT17 (Bojar et al., 2017)) use NMT, because it obtains higher scoring results, but does not require time-consuming procedures like feature generation or Quality Estimation (QE) to achieve quality MT translations.

CAT tools, and other systems using black-box MT, could benefit from a way of predicting which MT system will perform the best at translating a particular source segment. Such systems which typically use only one MT tool to translate all input could benefit from selectively using the output of multiple systems in this way.

This paper describes a series of experiments that attempt to take advantage of the strengths of alternative systems and combine system output to produce the best result. First, we describe our system, **SelecT**, which uses a neural-network based approach to predict which system provides the best output for translating a particular English sentence to Spanish using: Nematus (Sennrich et al., 2017), an NMT system; Moses (Koehn et al., 2007), a PBMT system; and Apertium (Forcada et al., 2011), an RBMT system. Then we use the MT system predicted to be the *best* [2] to improve pre-

[1] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

[2] according to BLEU score

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 209–218
Alacant, Spain, May 2018.

vious work on fuzzy-match repair (FMR), an approach that uses black-box machine translation as its primary method for translating sub-segments to be repaired (Ortega et al., 2014).

Most previous hybrid approaches to MT focus on ways to combine individual translations from different MT systems. In contrast, our system uses multiple predictive model types to choose the optimal sentence-level translations, without previous knowledge of the internal workings of the MT system. **SelecT** predicts which of 3 translation systems will produce the best translation. **SelecT** uses the performance differences seen on various tasks with different data where typically one MT system, be it rule-based, phrase-based, or neural, outperforms the other systems, to improve results by providing sentence-level predictions where often times differences in MT system quality can occur depending on the data.

In a professional setting, MT systems may have a higher cost due to quality performance issues and it would make sense that a translator has the most appropriate translation at hand when using the MT tool. Relying on a single MT system could be costly as shown in previous investigations (Rosti et al., 2007). We propose a prediction system that integrates easily into *any* system that uses black-box MT. Black-box MT systems would use the MT engine that **SelecT** predicts using a pre-translation performance metric. **SelecT** accepts any source sentence input s and produces a translation σ in a transparent way by predicting beforehand the system to use and querying the black-box MT system with the ideal (best-predicted) MT engine to use. Our goal is to measure how well mainstream MT systems perform and compare their differences for commercial use situations where often times translation quality should be determined beforehand to determine economic value.

2 Related Work

2.1 Fuzzy-Match Repair

Our system tests our MT results on an active implementation of fuzzy-match repair (Ortega et al., 2016) that uses Apertium (Forcada et al., 2011) as its MT engine. While previous work (Knowles et al., 2018) has already tested the black-box nature of FMR using 3 MT systems (Apertium, Moses, and Nematus), they do not attempt to predict which of those systems would perform best in a black-box translation task like we do here.

2.2 System Combination

There have been many papers about system combination in MT, so we will only highlight a few of them. Most researchers chose to combine systems using different methodologies. Published in 1994, Frederking et. al (1994) describe a Spanish to English system for synthesizing single translations of each sentence from parts of the translations produced by 3 MT engines: knowledge-based MT (PanGloss), example-based (EBMT) and a lexical-transfer+morphology system. Their combined system scores are measured by the number of keystrokes required to correct the automatic translations. In a similar way, Sańchez-Cartagena et. al (2016) show that an ensemble of an NMT and a PBMT system outperforms each of these systems individually when translating Finnish to English, as measured by BLEU. They use CMU's Multi-Engine Machine Translation (MEMT) Scheme (Heafield and Lavie, 2010) for system combination. MEMT aligns translations using METEOR (Lavie and Agarwal, 2007) and uses a beam search and a variety of features. Chaterjee et. al (2016) describes an MT system called "Primary" that includes an RNN implementation along with Moses. Their work, like others from WMT16 (Bojar et al., 2016), is mainly focused on translation tasks and improving translation by interchanging models. They do not chose the best system for each translation output; rather, they combine systems to produce the best output possible. Unlike approaches of system combination described above, our work focuses on predictions at the black-box, system-level input by predicting, beforehand, the optimum MT system to use. Our models are trained using a minimal amount of features and use sentence-level BLEU scores as the determining metric for labeling positive translation examples.

2.3 Evaluation and Quality Assessment

MT evaluation has been performed using many different metrics, e.g., those described in White et. al (1995). Those evaluations are very helpful to determine which MT system one would use for a specific metric. However, those metrics leave the guesswork up to the MT system or CAT tool user and do not attempt to predict which system to use.

In order to properly combine system output, it is necessary to assess the quality of that output. Formal evaluation requires human intervention (human translations or evaluations). In contrast, Quality Estimation (QE) (Specia et al., 2013), is a popular paradigm for automating assessment. QE uses a model to predict the quality of a translation without human intervention. The features that are used in QE are typically corpus-level features and are not based on previous (conflicting) translations from a different MT system. Nonetheless, one could add features to a QE system to perform work similar to ours - we skip the QE step for now as we are focused more on measuring how well a particular MT (or combination of MT) system(s) perform. Others have also performed research by measuring system output to determine the best model to use. Nomoto (2004), for example, use a voted language model based on support vector regression to determine a confidence score of a sentence in the translation output and use the highest scoring sentence as the final output. His approach is similar to ours; but, we use a different mechanism for selecting output based on several models to predict sentence-level quality before translating.

3 Methodology

Our work uses a predictive classifier to determine the best MT system for translation when used as a black box such that no prior knowledge of the internal workings of the MT system is necessary. It will allow any system with the ability to call a $translate()$ method access to sentence-level quality without the use of more complex paradigms such as quality estimation.

Combining several MT systems via our black-box method should achieve higher scores than just using one MT system. FMR (Ortega et al., 2016) is a recent example of a black-box $translate()$ method that uses one MT system, Apertium. Their work assumes no dependency on other parts of the MT system. Here, we use the work from Ortega et. al (2016) to show the advantage of having a mechanism to predict the best MT system to use before actually calling the $translate()$ method.

Our work intrinsically compares 3 open-source MT systems using 3 different classifier models: 1) Recurrent Neural Network (RNN), 2) FastText[3] classification, and 3) Logistic Regression (LR) described in section 5.1.1. Each model is created to predict sentence-level quality using BLEU. Then, we use both BLEU and word-error rate (WER) as a performance measurement to determine which model to use in FMR. WER for our experiments is considered as the word-based edit distance between the reference translation and the system translation often called *Levenshtein distance* (Wagner and Fischer, 1974). Our model is somewhat similar to a Quality Estimation model but based on MT engines alone. The prediction model is part of a bigger system that when given a new sentence s and a set of systems: $\{MT_{01}, MT_{02}, MT_{03}, ...\}$ derives a translation by selecting an MT system based on training data. Our hypothesis is that a system that can determine which MT engine to use before actually having the system translation should perform better and offer the best value for the translator or CAT tool user.[4]

After establishing that **SelecT** can select translation engines in a fashion that is beneficial to the user, we evaluate, with WER, **SelecT**'s choices using a system that uses black-box MT, fuzzy-match repair (Ortega et al., 2016). When the FMR system needs to translate any segment, whether an entire sentence or sub-segment of a sentence, it calls upon **SelecT** to determine which engine to use for the source sentence to be translated. Then, FMR calls its $translate()$ method with the MT engine suggested. We test **SelecT** in this paper using: Apertium, Moses, and Nematus. Our experiments measure WER from FMR when using **SelecT**. We aim to improve upon previous results (Ortega et al., 2016) by choosing the best predicted MT system for each translation in FMR.

4 Descriptions of MT Systems

4.1 Apertium

Apertium (Forcada et al., 2011) is a rule-based MT system employing manually created rules and dictionaries for each language pair. It is a community-based MT system that has a lot of contributors and provides an on-line translation tool [5] free for anyone's use. In addition to a large community base, there's a lot of documentation (Forcada et al., 2009) available that explain how the shallow-transfer MT system works.[6] We chose Apertium as the representative rule-based MT system because

[3] `https://github.com/facebookresearch/fastText/`

[4] Users of subscription MT services would only pay for sentences that SelecT chose to translate with a particular system. Thus they would not have to pay more than once for the same sentence.

[5] `http://apertium.org`

[6] Apertium works best with romance language pairs like ES–PT, ES–FR, etc. (Ortega et al., 2016; Knowles et al., 2018)

it's an open-source translation engine already used in a black-box translation system for FMR (Ortega et al., 2016). In order to align experiments with past work, we use the same version (SVN 64348) and language-pair package: apertium-en-es from Ortega et al. (2016). Apertium implements morphology through its modifiable technique called the lt-toolbox. It takes into account language structure by using part-of-speech tagging and chunking.

4.2 Moses

Moses (Koehn et al., 2007) is our representative phrase-based MT system. Previous black-box MT work (Knowles et al., 2018) found that Moses works well as a comparison MT engine.[7] Moses is the most widely adopted (non-neural) open-source *statistical* MT system. It combines statistical models with phrase tables to determine how to precisely translate unseen words. Moses is a complex system that, in our developmental experiments, performs well on word ordering and specific learned punctuation like "<<" and ">>" often used for translating quotation marks in our data. In several cases, Moses was the only MT system to correctly translate rare punctuation marks differences.

As a phrase-based MT system, Moses generally outperforms most other PBMT systems and is generally considered the de facto system to use for open-source MT (Dugast et al., 2007; Schwenk et al., 2012). It has already been compared to various neural MT systems. In particular, work from Junczys-Dowmunt et. al (2016) directly compares Moses against Nematus as does other work (Knowles et al., 2018). For the EN–ES language pair, BLEU scores reported in the work from Junczys-Dowmunt et. al (2016) were similar (about 1.4 difference).

4.3 Nematus

Nematus[8] is a neural MT system from the University of Edinburgh. It is implemented in Python, and based on the Theano framework (Sennrich et al., 2017). One major advantage that is pertinent to this paper is that Nematus uses byte-pair encoding (BPE) which starts from a character-level segmentation and eventually encodes full words as a single symbol (Sennrich et al., 2016). The potential for Nematus to score well on translations that differ at the character-level instead of at the word level is high.

Previous black-box comparison experiments for FMR (Knowles et al., 2018) also use Nematus. In WMT 2016, Nematus outperformed other MT systems with less complexity for feature engineering, i.e., Nematus requires training on word-embeddings alone while other systems, like Moses, require more complex statistical models and configuration parameters.

4.4 Advantages and Disadvantages

Based on the previous work using the 3 MT systems (Apertium, Moses, and Nematus), we believe that **SelecT** should outperform any single system. Each individual MT system has some particular advantage (or disadvantage) that would provide more information to a model for prediction to use when translating an unseen sentence. For example, Apertium may produce quality translations in some cases where morphology or part-of-speech linguistic features are absolutely necessary; Moses may perform better than Apertium on sentences that have frequent phrases; and Nematus will probably outperform the other systems for most sentences. Nematus should also do particularly well on character replacements and other sentences that require one-word deletion or insertion.

Luong et. al (2014) and Alva-Manchego et. al (2017) show that Moses is conservative with deletions, yet good with punctuation. However, both Apertium and Moses are unlikely to do well with lexical complexity (Luong et al., 2014). Apertium is good at making lexical and morphological distinctions. So, while it has been shown to perform worse on English to Spanish language pairs (Ortega et al., 2014), it is still worthwhile to use as a default system for testing due to its expert, handcrafted, methodology that is backed by an (HMM) (Cutting et al., 1992) which is known to classify parts of speech and morphemes well.

Some types of problems that an MT system may find with the test corpus, DGT-TM 2016,[9] relate to the corpus's parliamentary text. It contains punctuation irregularities and a lot of the segments that, due to its legal register, require a one-to-one alignment where the target (Spanish) words should not have to change much despite the language difference (English to Spanish). In addition, the text

[7] We trained Moses on Europarl V7(Koehn, 2005) and tuned it on WMT12.

[8] https://github.com/EdinburghNLP/nematus

[9] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

contains several hundred out-of-vocabulary (OOV) words which can be hard to cover with any MT system.

In summary, while Nematus is key to high quality translations, we should not dismiss Apertium or Moses since they translate some segments better than Nematus does.

5 Experimentation

5.1 Settings

Our experiments use several corpora and systems based on previous work on black-box MT (Knowles et al., 2018) and FMR (Ortega et al., 2016). Knowles et. al (2018) does a comparison of fuzzy-match repair using the 3 MT systems described in this paper. For both experiments, we use similar data. First we show that MT systems can be successfully selected; then we use the predictor for fuzzy-match repair. However, since we are trying to reproduce settings similar to (Ortega et al., 2016), there are some changes in the systems used.

5.1.1 MT-Experiments

There are 3 predictive models used to select an MT system based on training data. The implementation of each model is described in further detail below and found on Github [10]. All predictive models used the same DGT-2016 TM [11] for training. We divided DGT-2016 into an 80%/10%/10% split for train/dev/test, respectively. The dev set was used for error analysis and to help better understand the oracle (ensemble) settings. After gathering all of the data for statistical analysis, we used our saved models on the unseen test data. We use the EN–ES language pair from DGT-TM 2016 which contains 203,214 total parallel sentences. We lowercased all sentences and tokenized them using the tokenizer from the Moses baseline run.[12]

We test our predictive models on 3 MT systems (Apertium, Moses, and Nematus). The MT systems were similar in nature as far as the corpora used to train them, although, Apertium doesn't actually require training - it's a rule-based MT system. Apertium is a specific EN–ES version (SVN 64348). Our version of Moses mirrors the baseline[13] except for the training corpus, we train Moses using the EN–ES from EUROPARL v7 (Koehn, 2005) and tune, as in the baseline, on WMT12[14]. Our Nematus MT system is trained on Europarl v7 and News Commentary v10 data[15] (WMT13 training data for EN–ES).

Training is done where the best scoring system (according to BLEU) wins. There are 162571 sentences in the training set. In the training set, Apertium scores best on 26426 sentences; Moses scores best on 54372 sentences; and Nematus scores best on 81773 sentences. For our final test set, a perfect score for the **SelecT** system would be: 3441, 6602, and 10278, respectively. Therefore, we are training on what can be considered the "ensemble" system. Final test results report 2 metrics: 1)BLEU and 2) word-error rate (WER). We use 3 different algorithmic models for training:

1. **Bi-Directional Recurrent Neural Network** In the text we refer to this model as *RNN*. The model uses word embeddings created by Gensim[16] from the DGT-TM 2016 corpus with embedding dimensions of 300. Sentences of more than 100 words in length are discarded. The model itself is created using Theano[17] and has a gated recurrent unit (GRU) (Cho et al., 2014) with 300 hidden units as the recurrent neural layer. We use a dropout rate of 0.5 and RELU (Nair and Hinton, 2010) activation. This model is used with hopes that it has the ability to learn spontaneous words and activate clearly for system label classification where other (non-neural) models would not.

2. **FastText Supervised Learner** We chose the FastText [18] supervised model because it is a quick and efficient model that classifies text. For training, we use 25 epochs. For word embeddings we used a 300 dimension vector. The implementation is very straightforward and our command line options are passed such that the n-gram length is 5. All of our labels were passed in-line following the FastText installation instructions.

   For comparison purposes, FastText could be thought of as a neural net with a single hidden layer using bag-of-n-grams representation (we use 5-grams). This is a generaliza-

[10] https://github.com/AdamMeyers/Web-of-Law/EAMT2018

[11] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

[12] http://www.statmt.org/moses/?n=moses.baseline

[13] http://www.statmt.org/moses/?n=Moses.Baseline

[14] http://www.statmt.org/wmt12/dev.tgz

[15] http://www.casmacat.eu/corpus/news-commentary.html

[16] https://radimrehurek.com/gensim/

[17] http://deeplearning.net/software/theano/

[18] `https://github.com/facebookresearch/fastText/`

tion of bag-of-word logistic regression. For classification purposes, FastText works better in terms of classification. Our results show, however, that better classification accuracy does not necessarily result in better translation quality (BLEU).

3. **Logistic Regression** For our Logistic Regression (LR) model we used the popular Python machine learning framework SciKit-Learn v0.19.1 [19]. For sentence representations, SciKit-Learn is used to get bag-of-words (BOW) features and scored via term frequency inverse document frequency (TF-IDF) scores (Salton and Buckley, 1988).

Model training time differs for the 3 models. FastText and logistic regression (generating a bag-of-words representation and features based on TF-IDF features) can both be trained within several minutes (on 12 cores of an Intel Xeon E-2690v2 3.0GHz CPU), while it takes roughly 16 minutes to train the bi-directional recurrent neural network model per epoch (on one NVIDIA P40 GPU). For our purposes during the development stage, the best accuracy for the RNN was observed at 40 epochs. Clearly, in our experiments, the FastText and logistic regression models train faster than the RNN - one may want to consider these times for replication of our work in the future.

5.1.2 FMR Experiments

In order to replicate experiments from Ortega et. al (2016), we use exactly the same settings as they did. They use 1993 test sentences along with a translation memory extracted from DGT-TM 2015. We use an Apertium MT system(Forcada et al., 2011) (SVN 64348) similar to theirs (Ortega et al., 2016).

The other 2 MT systems that are used are Moses and Nematus. For the FMR experiments, we use the MT systems from section 5.1.1 to test on. All 3 systems (Apertium, Moses, and Nematus) make up part of the **SelecT** system that FMR uses when calling its black-box translate method such that the following steps occur:

1. a new source side sub-segment (σ or σ') is proposed for translation from FMR (for more details on FMR consult (Ortega et al., 2014)).

2. σ or σ' is passed as a new sentence to be classified to the **SelecT** system (**SelecT** does not actually run inside of FMR nor does it have knowledge of the internal workings of FMR).

3. the best performing model from previous experiments on **SelecT** (in our experiments it's the FastText model) is used to select whether Apertium, Moses, or Nematus is used to translate the sentence.

4. the black-box component of FMR translates σ or σ' using the selected MT system.

5. the black-box component of FMR returns a new translation τ or τ' respectively.

We use the best performing model (FastText) from our MT experiments to test FMR by allowing it to choose the best MT system when presented a new segment (or sub-segment) from the FMR's *translate()* method call. Results are reported for **SelecT** by measuring the WER produced when using the selected MT systems per sentence.

All systems WER are reported separately and with and without predictive tactics. It is worthwhile to note that there are cases when a fuzzy-match score is not met and the entire sentence (s' from (Ortega et al., 2016)) is translated. In those cases, we *also* use our predictive models from the **SelecT** system to choose an MT system to translate the entire sentence.

6 Results

We provide results of 2 experiments: experiment 1 measures the accuracy of the predictive models in **SelecT** using BLEU and WER as evaluation metrics. Experiment 2 uses **SelecT** as an agnostic predictor to choose an MT system for FMR. For experiment 1, we use 20321 sentences to test the 3 MT systems (Apertium, Moses, and Nematus) with 3 types of classification (RNN, FastText, and Logistic Regression). Table 1 shows how well each system performs in isolation – if we were to use the respective system as the sole translation engine for all 20321 sentences. Table 2 provides provides counts of sentences such that the corresponding model correctly predicts the highest BLEU score. It allows us to review the scores for each of the MT systems (Apertium, Moses, Nematus) at a localized level to show how well each system performs when it out-performs the other systems. For example, using the FastText system as

[19] `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

a predictor, Apertium outperforms Moses on 2283 of the 20321 total sentences.

System	BLEU	WER	Unique Tokens
Apertium	20.96	59.91	16773
Moses	30.05	54.02	21711
Nematus	37.36	51.77	26372

Table 1: BLEU, WER, and unique tokens for 3 MT systems

System	RNN	FT	LR	Ref
Apertium	2855	2283	1798	3441
Moses	6530	6553	5983	6602
Nematus	10936	11485	12540	10278

Table 2: Count of sentences for 3 predictive models

For even more details about our predictive models, we present the accuracy of our models in isolation on the 20321 test sentences. Table 3 shows how accurate each model is in predicting the MT system that would perform best using BLEU as the scoring metric. For example, the RNN **SelecT** system predicted the best MT system to use about 66% of the time.

System	Prec.	Rec.	F1	Acc.
RNN SelecT MT System				
Apertium	61.05	50.65	55.37	65.79%
Moses	59.25	58.60	58.92	
Nematus	70.94	75.48	73.14	
FastText SelecT MT System				
Apertium	70.52	46.79	56.25	68.12%
Moses	60.72	60.27	60.49	
Nematus	71.86	80.30	75.84	
Logistic Regression SelecT MT System				
Apertium	71.30	37.26	48.94	65.05%
Moses	57.60	52.20	54.76	
Nematus	67.71	82.61	74.42	

Table 3: Evaluation of 3 models on 3 MT systems

Lastly, in Table 4, we report system combination scores as follows: 1) the ensemble system, **SelecT**, selects translations based on the predictive model; 2) the upper bound: always choosing the best scoring system; 3) the lower bound: always choosing the worst scoring system.

System	BLEU	WER	Unique Tokens
Best	40.08	46.70	23767
Worst	18.97	63.91	18595
RNN	37.36	49.69	24546
FastText	38.01	49.55	24790
LR	38.03	49.97	24935

Table 4: Comparison of 3 **SelecT** MT systems

Our FastText system. for example, had a 19.04 improvement over the BLEU lower-bound of (90.2% of the potential difference) and a 14.36 improvement over the WER lower-bound (83.4% of the potential difference), in both cases, this is significantly more than the average of the upper and lower bounds (29.53 BLEU score and 55.31 WER). The ensemble system (using FastText) also out-performs the best individual system (Nematus) by .65 Blue and 2.22 WER. The average between the upper and lower bounds is a good baseline to beat, to demonstrate that our system is successful at predicting the correct high-scoring system most of the time. However, being the best system gives the results practical value.

We observe that Nematus is more likely to correctly handle polysemous words (should English *march* be translated to Spanish as *marzo* (the month) or *marcha* (the action)). However, some of Nematus' errors involve seemingly arbitrary translations of words or the addition of arbitrary words. For example, the English ”*identification numbers*” is correctly translated as ”*números de identificación*” by Apertium, but Nematus translates it as *identificación de identificación* (Moses translates it nearly correctly, but leaves off the ”s” in ”*números*”). Similarly, Apartium correctly translates the English ”*saffron*” as *azafrán*, whereas Moses leaves it untranslated (”*saffron*”) and Nematus translates it mysteriously as ”lágrimas de los perros”.

6.1 FMR-based performance

We evaluate our best performing model (FastText) from 5.1.1 on the agnostic black-box MT system from FMR (Ortega et al., 2016). Table 5 shows our approach for 3 different fuzzy-match score thresholds (FMT) —60%, 70% and 80%—. For our experiments, we use a Levenshtein-based word-error rate distance measurement as described earlier. **SelecT** models are used to select translations for all potential segments (s' segments and sub-segments σ and σ' in work from Ortega et. al (2016)) when

		Apertium		**Moses**		**Nematus**		**SelecT**	
	TM	MT	FMR	MT	FMR	MT	FMR	MT	FMR
FMT: 60%									
Error (%)	55.0	65.3	36.5	45.8	29.2	48.6	30.1	44.8	27.9
Er. (%) on matches	20.1	65.3	17.9	45.8	16.2	48.6	17.1	44.8	16.0
# matches	1184	1993	1184	1993	1184	1993	1184	1993	1184
Avg. length	22.6	22.1	22.6	22.1	21.1	22.3	21.3	22.1	22.8
FMT: 70%									
Error (%)	61.0	65.3	38.5	45.8	30.5	48.6	31.15	44.8	29.2
Er. (%) on matches	16.3	65.3	14.6	45.8	13.7	48.6	13.9	44.8	13.5
# matches	828	1993	828	1993	828	1993	828	1993	828
Avg. length	22.4	22.1	22.5	22.1	22.8	22.2	22.8	22.1	22.7
FMT: 80%									
Error (%)	69.7	65.3	42.6	45.8	32.6	48.6	33.7	44.8	31.7
Er. (%) on matches	13.1	65.3	11.9	45.8	11.3	48.6	11.4	44.8	11.2
# matches	660	1993	660	1993	660	1993	660	1993	660
Avg. length	22.3	22.2	22.4	22.1	23.4	22.2	23.4	22.1	22.8

Table 5: Word-Error Rate (WER) evaluation for FMR using **SelecT** and black-box MT

FMR creates a hypothesis $t*$; then, FMR selects the best hypothesis according to the edit-distance between the hypothesis and the reference t'.

Like work from Ortega et. al (2016) we report on 2 error rates: 1) WER computed on the whole test set and 2) WER computed only on the segments for which a translation unit (TU) with a fuzzy-match score above a threshold is found (error on matches). We use the 2 different forms of measurement to better understand how a translator or CAT tool user would use FMR in a production setting since they would typically only see matches. It is also worthwhile to note that the scores for FMR are based on an oracle setting which implies knowledge of the reference translations (t' for each hypothesis ($t*$).

As seen in Table 5, the **SelecT** system performs better than Ortega et. al (2016). In addition to outperforming work by Ortega et. al (2016), it seems to score well when compared to other work by Knowles et. al (2018). An explanation by Knowles et. al (2018) has already been given as to why Moses performs better in certain situations. It's our belief that in addition to previous work from both authors, our prediction system scores well due to the trained knowledge it has gained from DGT-TM 2016 which is similar to DGT-TM 2015, despite the MT systems themselves being trained on Europarl V7. **SelecT** outperforms all systems in both fuzzy-match situations (matched or not). It even performs better when there's no fuzzy-match and the MT system has to translate the entire source segment (s' in Ortega et. al (2016)).

FMR (Ortega et al., 2016) has already shown to be a potential win for improving translator's productivity. The **SelecT** system presented here shows performance gains of as much as 2 points in WER over previous work (Ortega et al., 2016). We believe that the gains presented here, much like points brought up in 5.1.1, are due to Moses and Apertium's phrase-based and rule-based technology that allow it to come somewhat closer to translator's needs at the sub-segment level. Subsegments in FMR are usually shorter and have more punctuation involved (especially in the DGT-TM 2015 corpus); it's the case here that an ensemble system covers more cases than any one MT system tested and could, thus, be more valuable for a translator or CAT-tool user.

7 Conclusion

Our experiments show that **SelecT** can be used to increase performance in black-box MT tools. **SelecT** is agnostic to other processes in a typical MT pipeline and does not require underlying process changes in current black-box MT systems. **SelecT** only requires access to a command-line utility that accepts a sentence as input to select the best MT system. The work presented in section 5.1.1 also helps explain how well various models perform for black-box systems. Baseline MT systems are combined with a predictive model to create a non-traditional ensemble for improving translations from tools using black-box translation. In our experiments, FastText outperformed other models as measured by BLEU and WER. There are surely more prediction models (non-baseline) that could perform better but we leave that for future work.

8 Future Work

We are considering several avenues for future work including trying additional classifiers for choosing the best MT system including a convolutional neural network (CNN). We would also like to try additional MT systems such as OpenMT [20] or Google translate.[21] In particular, it would be nice to demonstrate whether it is as important to combine diverse systems as it is to combine high-performing systems when creating an ensemble. Our classifiers were also very similar to most baseline systems conventionally found on-line. We feel that by training the systems on more in-domain data as presented in previous work (Knowles et al., 2018), we would improve the results. The classifiers could also be trained with more information about the text very similar to the QE tasks presented by Specia et. al (2010). One could also use QE as a corner stone for leveraging systems that would not only predict via sentence-level features; but, could also predict using the other features presented at the post-editing level as done by Chatterjee et. al (2015).

9 Acknowledgements

We thank Rebecca Knowles for providing MT output using Nematus models from Knowles et al. (2018). John E. Ortega is supported by the Universitat d'Alacant and the Spanish government through the EFFORTUNE (TIN2015-69632-R) project. Kyunghyun Cho was partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI) and Samsung Electronics (Improving Deep Learning using Latent Structure).

References

Alva-Manchego, F., J. Bingel, G. Paetzold, C. Scarton, and L. Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 295–305.

Arenas, A. G. 2013. What do professional translators think about post-editing. *The Journal of Specialized Translation*, (19).

Bentivogli, L., A. Bisazza, M. Cettolo, and M. Federico. 2016. Neural versus phrase-based machine translation quality: a case study. *CoRR*, abs/1608.04631.

Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, Antonio Jimeno Y., P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. 2016. Findings of the 2016 conference on machine translation. In *WMT16*, pages 131–198.

Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *WMT17*, pages 169–214.

Chatterjee, Rajen, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the planet of the apes: a comparative study of state-of-the-art methods for mt automatic post-editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 156–161.

Chatterjee, R., J. GC de Souza, M. Negri, and M. Turchi. 2016. The fbk participation in the wmt 2016 automatic post-editing shared task. In *WMT16*, pages 745–750.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cutting, D., J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140.

Dugast, L., J. Senellart, and P. Koehn. 2007. Statistical post-editing on systran's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223.

Forcada, M. L, B. I. Bonev, S Ortiz Rojas, J. P. Ortiz, G R. Sánchez, F S. Martínez, C. Armentano-Oller, M. A Montava, and F. M Tyers. 2009. Documentation of the open-source shallow-transfer machine translation platform apertium. *Online Departament de Llenguatges i Sistemes Informatics Universitat d Alacant, Available: http://xixona. dlsi. ua. es/˜fran/apertium2-documentation.*

Forcada, M. L., Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Frederking, R. and S. Nirenburg. 1994. Three heads are better than one. In *Proceedings of the ANLP*.

[20] `http://opennmt.net/`

[21] `http://translate.google.com`

Heafield, K. and A. Lavie. 2010. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36.

Junczys-Dowmunt, M., T. Dwojak, and H. Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. *arXiv preprint arXiv:1610.01108*.

Knowles, R., J. E Ortega, and P. Koehn. 2018. A comparison of machine translation paradigms for use in black-box fuzzy-match repair. In *AMTA 2018*, volume 1, pages 249–255.

Koehn, P. and R. Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Lavie, A. and A. Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *StatMT '07*, pages 228–231.

Luong, T., I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. 2014. Addressing the rare word problem in neural machine translation. *CoRR*, abs/1410.8206.

Nair, V. and G. E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML-10*, pages 807–814.

Nomoto, T. 2004. Multi-engine machine translation with voted language model. In *ACL '04*.

Ortega, J. E, F. Sánchez-Martınez, and M. L Forcada. 2014. Using any machine translation source for fuzzy-match repair in a computer-aided translation setting. In *AMTA 2014*, volume 1, pages 42–53.

Ortega, J. E., F. Sánchez-Martínez, and M. L. Forcada. 2016. Fuzzy-match repair using black-box machine translation systems: what can be expected? In *AMTA 2016, vol. 1*, pages 27–39.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235.

Salton, G. and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523.

Sánchez-Cartagena, V. M. and A. Toral. 2016. Abumatran at wmt 2016 translation task: Deep learning, morphological segmentation and tuning on character sequences. In *WMT16*.

Schwenk, H., A. Rousseau, and M. Attik. 2012. Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 11–19.

Sennrich, R., B. Haddow, and A. Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Sennrich, R., O. Firat, K. Cho, Alexandra Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. L"aubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Demonstrations at the 15th Conference of the EACL*.

Specia, L., D. Raj, and M. Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

Specia, L., K. Shah, J. G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria.

Wagner, R. A and M. J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

White, J. S. 1995. Approaches to black box mt evaluation. In *Proceedings of Machine Translation Summit V*, volume 10.

Data selection for NMT using Infrequent n-gram Recovery

Zuzanna Parcheta[1] **Germán Sanchis-Trilles**[1] **Francisco Casacuberta**[2]
[1]Sciling S.L., Carrer del Riu 321,Pinedo, 46012, Spain
[2]PRHLT Research Center, Camino de Vera s/n, 46022 Valencia, Spain
{zparcheta, gsanchis}@sciling.com
fcn@prhlt.upv.es

Abstract

Neural Machine Translation (NMT) has achieved promising results comparable with Phrase-Based Statistical Machine Translation (PBSMT). However, to train a neural translation engine, much more powerful machines are required than those required to develop translation engines based on PBSMT. One solution to reduce the training cost of NMT systems is the reduction of the training corpus through data selection (DS) techniques. There are many DS techniques applied in PBSMT which bring good results.

In this work, we show that the data selection technique based on infrequent n-gram occurrence described in (Gascó et al., 2012) commonly used for PBSMT systems also works well for NMT systems. We focus our work on selecting data according to specific corpora using the previously mentioned technique. The specific-domain corpora used for our experiments are IT domain and medical domain. The DS technique significantly reduces the execution time required to train the model between 87% and 93%. Also, it improves translation quality by up to 2.8 BLEU points. The improvements are obtained with just a small fraction of the data that accounts for between 6% and 20% of the total data.

1 Introduction

Until recently, machine translation (MT) systems were based mostly on PBSMT. Today, the state of the art of MT is NMT. It has been shown that neural networks can improve the quality of translations by up to several BLEU points and also make them more fluid (Toral and Sánchez-Cartagena, 2017). However, NMT is computationally much more expensive. To train an NMT engine, much more powerful machines are required than would be used for building translation engines based on PBSMT. For example, NMT engines require more RAM memory, one or several GPUs and storing the models requires more storage capacity. Also, the training time of an NMT system is significantly longer than that of the systems based on PBSMT (Shterionov et al., 2017). One solution to reduce the training cost of NMT systems is the reduction of the training corpus through DS techniques. Bilingual sentence selection (BSS) is a type of DS where the best subset of bilingual sentences from the available parallel corpora is selected and leveraged to train a translation system. To date, many DS techniques are known that are applied to PBSMT systems, bringing very promising results. Some of them not only reduce the training time but also outperform a system where all the bilingual data available is used, given that the selected sentences are better suited to the domain being dealt with.

In this work, we demonstrate that a DS technique commonly used for PBSMT can also yield satisfactory results when applied in NMT systems. To prove a good performance of DS in NMT we select sentences from a large amount of data from different domains with the purpose of

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 219–227
Alacant, Spain, May 2018.

enlarging a small size, in-domain training corpus. The selection of more suitable sentences achieves improvements in translation quality.

2 Related Work

When creating a machine translation system, it is important to select high-quality bilingual data with a domain similar to the one in which the translation system will be used.

There are multiple techniques of DS for PBSMT based on perplexity as (Gao et al., 2002), where the authors use maximum-likelihood based methods to select the lexicon, segment words, filter and adapt the training data, and reduce language model size. In (Moore and Lewis, 2010), data selection is done comparing the cross-entropy according to domain-specific and non-domain specific. In (Axelrod et al., 2011), sentences are selected with a bilingual cross-entropy based method. The selected subset is used to train a small domain-adapted PBSMT system. This domain-adapted system is combined with the real in-domain PBSMT system.

Also, there are techniques based on distributed representations of words. In (Chen et al., 2016) and (Chen and Huang, 2016), sentences are selected using a convolutional neural network. In (Chinea-Rios et al., 2016), a continuous vector-space representation of word sequences is used for selecting the best subset of a bilingual corpus. In (Peris et al., 2017), a new data selection method is developed, based on a neural network classifier.

Other data selection techniques rely on information retrieval based methods. In (Lu et al.), training data is adapted by redistributing the weight of each training sentence pair.

There are also DS techniques which select sentences relying on information from the development and test set. In (Gascó et al., 2012) two data selection techniques are presented: 1) Probabilistic sampling, that introduces new sentences into the in-domain corpus without distorting the original distribution. First, the sentences are selected according to length, then according to probability. The second technique presented in that work is infrequent n-gram recovery. This technique relies on the idea of enforcing model coverage for those n-grams that are present in the (source) test set. In (Biçici and Yuret, 2011), the authors explore the use of a data selection in a transductive scenario. Feature decay algorithms increase the diversity of the training set by devaluing features that are already included.

All commented techniques were initially implemented for PBSMT systems. There are also some techniques designed explicitly for NMT systems. In (Farajian et al., 2017), the authors present an instance-based adaptive NMT approach that effectively handles translation requests from multiple domains in an unsupervised manner, that is without knowing the domain labels. In (Chinea-Rios et al., 2017), the method developed consists in selecting, from a large monolingual pool of sentences in the source language, those instances that are more related to a given test set. Next, this selection is automatically translated and the general neural machine translation system is fine-tuned with this data.

Also, there are some works that compare the effectiveness of data selection techniques in PBSMT and NMT. In (van der Wees et al., 2017), the authors compare the effects of a commonly used data selection approach (bilingual cross entropy) on PBMT and NMT using four different domains. They also introduce dynamic data selection as a way to make data selection profitable for NMT.

3 Infrequent n-gram Recovery

The data selection technique used in this work is called Infrequent n-gram Recovery (Gascó et al., 2012). The main use of this technique is when the in-domain corpus provided is too small to train properly the translation engine. This technique consists on enlarging the in-domain training set by selecting sentences from a non domain-specific pool of sentences to maximise the coverage of n-grams which appear in the test and development set. For this, it is necessary to establish the minimum number of occurrences (t) required for a certain *n*-gram to be considered as infrequent, and also the order n of the n-grams (unigrams, bigrams, 3-grams etc.) that will be considered. The selected sentences will contain n-grams considered infrequent. With that we ensure that the training set will contain all n-grams from test and development set t times, as long as this is possible with the available out of domain dataset. The pool of sentences will be oppositely denoted as the *out-of-domain* corpus.

Sentences in the out-of-domain pool are sorted by their infrequency score in order to select first the sentences which most improve the coverage of n-grams belonging to the in-domain dataset which might be considered infrequent. Let $\mathcal{X}$ be the set of n-grams that appear in the sentences to be translated and $\mathbf{w}$ one of them; $C(\mathbf{w})$ the counts of $\mathbf{w}$ in the source language training set; t the threshold of counts when an n-gram is considered infrequent, and $N(\mathbf{w})$ the counts of $\mathbf{w}$ in the source sentence $\mathbf{f}$ to be scored. The infrequency score of $\mathbf{f}$ is:

$$i(\mathbf{f}) = \sum_{\mathbf{w} \in \mathcal{X}} \min(1, N(\mathbf{w})) \max(0, t - C(\mathbf{w})) \quad (1)$$

It already was demonstrated that the Infrequent n-gram Recovery technique works very well in PBSMT systems improving up to 1 point of BLEU when compared to training with all the data available (in-domain + out-of-domain), while using only 0.5% of total data. The fact, that the Infrequent n-gram Recovery technique works well in PBSMT system does not mean that it will work fine for NMT, since PBSMT and NMT build the translation model in very different ways. PBSMT splits sentences into smaller chunks and looks for similar occurrences in other languages according to a statistical model. The alignment matrix can not be well estimated if words and n-grams appear rarely in the training corpus. Also, the out-of-vocabulary words can not be translated by PBSMT model. The behaviour of NMT systems is different to PBSMT. NMT generates sequence of words in the target language given an input sequence of words in the source language. The translation is done following an encoder–decoder architecture. The encoder represents the input sequence using a word embedding model (Mikolov et al., 2013), and the decoder generates the sentence in the target language word by word (Sutskever et al., 2014). In NMT, it is necessary to adjust hyper-parameters as learning rate, number of hidden layers, and number of epochs. NMT needs to deal with millions of parameters coming from each neural network unit (weights and biases) to adjust the translation model. The best model is then selected according to translation quality on the development set.

Up until now there is no study about the efficiency of Infrequent n-gram Recovery in NMT.

4 Experiments

The experiments were conducted using the OpenNMT (Klein et al., 2017) deep learning framework based in Torch. This toolkit is mainly specialised in sequence-to-sequence models covering a variety of tasks such as machine translation, image to text, and speech recognition.

All experiments were conducted using an NVIDIA GTX 1080 GPU with 8GB of RAM.

To select domain-specific sentences, we need a small size in-domain dataset and an out-of-domain dataset which contains sentences from different domains. Then, we select sentences from the out-of-domain corpus to enlarge the in-domain corpus.

4.1 Experimental setup

We used two in-domain corpora for our experiments: Medical Web Crawl and IT. Medical Web Crawl is a subset of the UFAL Medical Corpus[1], which contains specific medical vocabulary and expressions; the IT corpus[2] contains sentences belonging to the IT domain. Main figures of both corpora are shown in Tables 1 and 2.

Table 1: Medical Web Crawl main figures. k denotes thousands of elements and M denotes millions of elements. $|S|$ stands for number of sentences, $|W|$ for number of running words, and $|V|$ for vocabulary size.

Subset	language	$\lvert S\rvert$	$\lvert W\rvert$	$\lvert V\rvert$
train	English	130k	1.9M	44.0k
	Spanish	130k	2.1M	54.5k
dev	English	806	12.3k	2.9k
	Spanish	806	13.4k	3.5k
test	English	810	12.1k	2.8k
	Spanish	810	13.3k	3.3k

We use two different out-of-domain corpora for each in-domain corpus. In the case of the IT corpus we use Europarl[3] as the out-of-domain dataset. In the case of Medical Web Crawl, we use JRC-Acquis[4] and Europarl jointly. JRC-Acquis is a collection of legislative text of the European Union and Europarl is a parallel corpus extracted from the European Parliament website. The purpose of using two different corpora for each

[1] https://ufal.mff.cuni.cz/ufal_medical_corpus
[2] http://www.statmt.org/wmt16/it-translation-task.html
[3] http://opus.nlpl.eu/Europarl.php
[4] http://opus.nlpl.eu/JRC-Acquis.php

Table 2: IT corpus main figures. k denotes thousands of elements. $|S|$ stands for number of sentences and M denotes millions of elements., $|W|$ for number of running words, and $|V|$ for vocabulary size.

Subset	language	$\|S\|$	$\|W\|$	$\|V\|$
train	English	147.9k	1M	44.4k
	Spanish	147.9k	1M	50.3k
dev	English	1.7k	32.4k	2.9k
	Spanish	1.7k	34k	3.4k
test	English	857	15.6k	2k
	Spanish	857	17.4k	2.4k

domain was to analyse system performance under different conditions: 1) a first condition (IT domain) in which training the system on all the available data (in-domain and out-of-domain data) leads to better results than training it only on the in-domain data; and 2) a second experiment (medical domain) in which training the system on all the available data leads to worse results than training the system on only the in-domain data. These two different scenarios allow us investigate the behaviour of the DS selection technique used in this work in a scenario where similar-domain data is abundant, but also in a scenario where similar-domain data is scarce. In both cases, sentences longer than 40 words were pruned. Main figures of the out-of-domain corpora are shown in Table 3. All data was previously tokenised and lowercased.

Table 3: Out-of-domain corpora main figures. k denotes thousands of elements. $|S|$ stands for number of sentences and M denotes millions of elements., $|W|$ for number of running words, and $|V|$ for vocabulary size.

Corpus	language	$\|S\|$	$\|W\|$	$\|V\|$
Europarl	English	1.7M	32.8M	118k
	Spanish	1.7M	33.9M	167k
JRC +	English	2.2M	41.2M	151k
Europarl	Spanish	2.2M	43M	198k

We conducted data selection experiments using the Infrequent n-gram Recovery technique. For each in-domain dataset (Medical and IT), the experiments were performed considering n-grams with $n \in \{1, \ldots, 5\}$ for computing the infrequency score (Equation 1). For each n-gram we conducted experiments for thresholds $t \in \{10, 20, 30, 40\}$. The count of infrequent n-grams was done on test and development set jointly. The reason for doing so was that the best model in NMT is chosen according to the best BLEU achieved on the development set. To ensure similar conditions in development and in test, it is important to ensure that all n-grams from the test and development sets appear in the training set. The data selected, together with the in-domain corpus, were used to train the reduced model.

We trained a Byte Pair Encoding model (BPE) (Sennrich et al., 2015) on the selected data and we applied the BPE model to training, development and test set. Then, we trained a recurrent neural network (RNN) (Schuster and Paliwal, 1997) with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) on encoder and decoder side, each of them with only one layer because of the high computational cost entailed. We used a global attention layer to improve translation by selectively focusing on parts of the source sentence during translation. We also used a dropout rate of 0.2, and the *adam* (Kingma and Ba, 2014) optimiser with learning rate of 0.0002. The model featured 512 hidden units and 512-dimensional embedding vectors. The training procedure was run for 40 epochs and we selected the best epoch according to the development set.

We considered three different baseline systems against which to compare our DS systems: first, a model trained only with in-domain data; second, a model trained with all data available (in-domain and out-domain corpora jointly); third, a model trained on data selected at random. For this last baseline, we repeated the random selection procedure 5 times, reporting in our experiments the average of those 5 different experiments

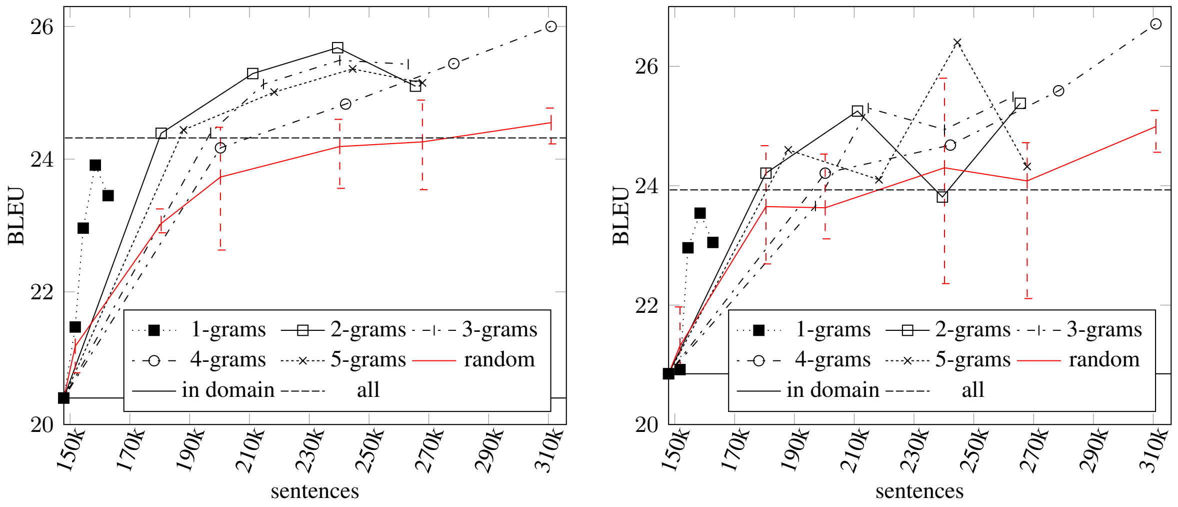
System performance was measured in terms of BLEU (Papineni et al., 2002), which measures n-gram precision with respect to a reference set, with a penalty for sentences that are too short.

4.2 Results

In this section we will analyse the results obtained for both domains. Given that the purpose of evaluating on the IT and medical domain is different, we will analyse the results obtained separately.

4.2.1 IT domain results

In Figure 1, we show BLEU scores for models trained with data selected according to different



(a) IT corpus development set

(b) IT corpus test set

Figure 1: Effect of adding sentences over the BLEU score in IT domain for n-grams $N = \{1, 2, 3, 4, 5\}$ with threshold $t = \{10, 20, 30, 40\}$, where $t = 10$ includes the lowest number of sentences. Figure 1a shows BLEU score for development set and Figure 1b shows BLEU score for test set. Red dashed lines show confidence intervals for random selection.

Table 4: Examples of translated sentences by the best model: 4-grams and t=40. In each example, we show source sentence (src), target sentence (ref), a hypothesis generated by the best model (hyp) and also, a hypothesis with a random model (hyp random). The random model is one of 5 random experiments conducted with the same number of sentences as our best model. This random model was chosen by BLEU score nearest to medium score from all 5 random models.

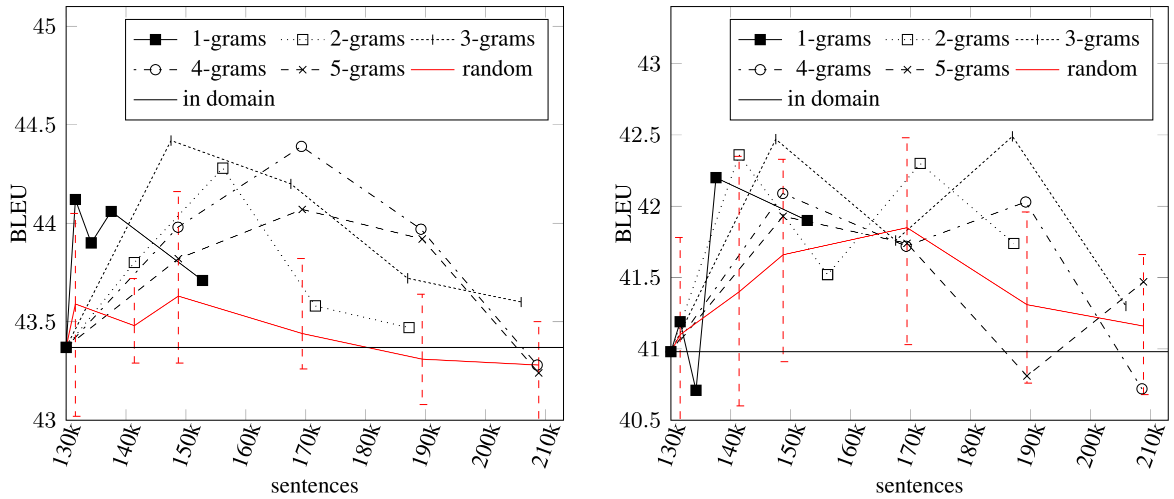
Example 1	
src	try to close and reopen the program.
ref	intente cerrar y abrir de nuevo el programa.
hyp	intentar cerrar y reabrir el programa.
hyp random	intentar cerrar y reabrir el programa.
Example 2	
src	try to shut down your computer, wait a few seconds, and boot it up again.
ref	intente apagar su ordenador, espere unos segundos, y reinícielo de nuevo
hyp	intentar cerrar su equipo, espere unos segundos, y la arranque de nuevo.
hyp random	intentar cerrar su equipo , espere unos pocos segundos y su arranque de nuevo.
Example 3	
src	click the apple icon, then select shut down.
trg	haga clic en el icono de apple y seleccione apagar.
hyp	haga clic en el icono de apple, luego seleccione cierre.
hyp random	pulse en el icono de apple cerrar.
Example 4	
src	someone probably reported you for copyright infringement.
trg	es probable que alguien haya informado al servicio de la infraccin de copyright.
hyp	alguien ha informado probablemente por infraccin de derechos de autor.
hyp random	alguien probablemente ha informado de sus derechos de autor.

order of n-grams and different threshold t. Also, we include the score obtained by a model trained only with in-domain data, and the score obtained by a model trained with all available data. Moreover, we show the average score of all 5 random models, with confidence intervals.

The best model obtained for the IT domain, according to the development set, is the model trained with data selected with n-grams up to order 4, with t=40. Our best model, obtained after epoch 7, reaches 26.7 BLEU on the test set. As described in Section 4.1, we compare our system against three different baselines:

1) Only in-domain data: The model trained only with in-domain data achieves 20.9 BLEU on the test set. Our system is able to improve this score by 5.8 BLEU points.

2) All data: The model trained with all data (in-domain and out-of-domain jointly) achieves 23.9 BLEU. Our system is able to improve this score by 2.8 BLEU points.



(a) Medical corpus development set

(b) Medical corpus test set

Figure 2: Effect of adding sentences over the BLEU score in medical domain for n-grams $N = \{1, 2, 3, 4, 5\}$ with threshold $t = \{10, 20, 30, 40\}$, where $t = 10$ includes the lowest number of sentences. Figure 2a shows BLEU score for development set and Figure 2b shows BLEU score for test set. Red dashed lines show confidence intervals for random selection.

Table 5: Examples of translated sentences by the best model: 3-grams and t=10. In each example, we show source sentence (src), target sentence (ref), a hypothesis generated by the best model (hyp) and also, a hypothesis with a random model (hyp random). The random model is one of 5 random experiments conducted with the same number of sentences as our best model. This random model was chosen by BLEU score nearest to medium score from all 5 random models.

Example 1	
src	wash hands and arms thoroughly after cleaning aquariums . or , wear rubber gloves when cleaning
ref	lávese muy bien las manos y los brazos después de limpiar acuarios o utilice guantes de caucho al realizar la limpieza .
hyp	lávese bien las manos y los brazos completamente después de limpiar los acuarios de limpieza o, use guantes de goma al limpieza.
hyp random	lávese bien las manos y los brazos bien después de limpiar los guantes de venta libre .
Example 2	
src	mellaril overdose ; hydrochloride - thioridazine overdose
ref	sobredosis de mellaril ; sobredosis de hidrocloruro de tioridazina
hyp	sobredosis de mogyil ; sobredosis de troridazina
hyp random	sobredosis de molcio
Example 3	
src	histamine h2 receptor blockers
trg	bloqueadores de los receptores h2 de la histamina .
hyp	bloqueadores h2 de la histamina los receptores de la histamina
hyp random	bloqueadores de los 2 bloqueadores
Example 4	
src	more than 200,000 had to go to the emergency room
trg	más de 200,000 acudieron a salas de emergencias,
hyp	más de 200.000 acudieron a la sala de urgencias
hyp random	más de 9,000 se sometieron a la sala de urgencias

3) Random selection: The average of scores achieved by the 5 random selections on the out-of-domain corpus is 25.0 points of BLEU. Our best model is able to improve this score by about 1.7 points of BLEU. Also, our model is also able to improve over the best of the models obtained with random selection by 1.4 points of BLEU.

The results described above are promising, since we are able to reach improvements in translation quality by selecting only 163k sentences, which represents 20% of all data available, and reduction of training size also implies reduction in model size and execution time: training a model with all the data available takes 10 days 6 hours, compared to 33 hours for the model trained with selected data using 4-grams and t=40, which implies a reduction of computational time by 87%.

Analysing the random selection score in

Figure 1a, it can be seen that the more sentences added in the random setting, the better the score in development. However, this is not so clear in test conditions (1b). In the case of the test set, the plot shows much more noise in the case of random selection.

Examples of translations generated by our model are shown in the Table 4. To compare the quality of the translations generated we show source and target sentences, which correspond to the reference translation. Also, we include the translation obtained by the best random selection, with a comparable number of selected sentences. We can see that the hypotheses of our model (hyp) and the hypothesis of the random model (hyp random) are pretty similar. In Example 1, both hypotheses are the same, and they are perfectly understandable synonyms of the reference translation. In Example 2, the hypotheses of random selection is mostly correct, but the use of the wrong article makes it difficult to understand. In Examples 3 and 4, our model generates a perfect translation. In contrast, the hypotheses generated by the random model have missed words in some cases, and in other present word substitutions that imply that the translation is disfluent and sometimes unable to convey the appropriate meaning.

4.2.2 Medical domain results

In Figure 2, we show the results for the medical domain. The score achieved by the model trained with all data is much lower than the score of the model trained only on in-domain data. For this reason, and for clarity purposes, we did not include the score of the system trained on all the data available. As in the case of the IT domain, we include the score of different systems obtained by selecting data with different order of n-grams and different thresholds t. Moreover, we show the average score of all 5 random selections, with confidence intervals.

In the case of the medical domain, the model trained only with in-domain data achieves 41 BLEU and leads to improvements over the model trained on all the data available, which reaches only 35 BLEU. It supports the hypothesis from (Gascó et al., 2012) that more data not always yields better results.

In case of the medical domain, the best model is trained on a selection obtained by n-grams up to order 3, with threshold t=10, after 19 epochs. This model achieves 42.5 BLEU on the test set with only 41.6K sentences added, which represents only 6% of all data. Our model achieves the following improvements over each of the three baselines described:

1) 1.5 points of BLEU over in-domain

2) 7.5 points of BLEU over out-of-domain

3) 0.8 points of BLEU over the average of scores of the 5 models trained with randomly selected data. Also, the system trained with Infrequent n-grams also improves by 0.1 BLEU over the best system obtained with random selection.

Observing the random-selection curve in Figure 2a, we realise that the more sentences added at random, the worse the BLEU score. We understand this is an evidence that signals that including sentences from an out-of-domain corpus leads to having the in-domain information overwhelmed, yielding a model which is not well suited for the specific domain at hand.

In the case of the development set, BLEU tends to degrade as soon as we add sentences after threshold *t=10* or *t=20*. However, in the case of the test set (Figure 2b), the plot is very noisy, and no clear pattern can be observed, both in the case of random selection and in the case of Infrequent n-gram selection.

It must be noted that training the system on all the data available took 12 days. In contrast, training the system with the selected data only took 17 hours, which entails a reduction of 93%. In Table 5, we show some examples of translations generated by our best model (3-grams with threshold *t=10*). Although a lot of sentences translated by our model and by the random model are very similar, we find some differences which lead us to think that our model generates better quality translations. In Examples 1, 2 and 3, shown in Table 5, the translations generated by random selection present some disfluencies. This model reorders and misses words causing the sentences to not be understandable. In Example 4, we can see that random selection translates a number incorrectly.

5 Conclusions

PBMT and NMT estimate the translation model in a different way. PBMT estimates the parameters using statistical models and use word alignments to generate the translation. Instead, NMT features an encoder-decoder architecture. The encoder represents a sequence of input words mapping them to vectors of real numbers and then the decoder generates the output sequence in a word-by-word basis.

In our work, we show that Infrequent n-gram Recovery brings very satisfactory results when applied to NMT. We demonstrate that, by selecting a subset of data more suitable to a specific in-domain corpus, we can get a model whose quality can improve the quality of a model trained with all the data available (in-domain and out-of-domain data jointly). Such was the case with the IT corpus. In contrast, a less usual case is when the model trained with all data performs worse than one trained with only in-domain data. This was the case with the medical domain dataset. It can be due to very specific vocabulary appearing in the in-domain corpus, and such vocabulary not being frequent in the out-of-domain data. This entails that including sentences from different domains lead to worse translation quality. Despite this fact, the technique described manages to select only sentences that lead to improvements over the translation quality achieved by a system trained only with in-domain data.

In our experiments, we achieve improvements of up to 1.7 BLEU points over a model trained with a random selection of data. In the case of the IT corpus, we improved translation quality by about 2.8 points of BLEU when compared to a model trained on all the data available.

Another important issue is the reduction of execution time. By reducing the amount of training data, we achieved a reduction in execution time between 87% and 93%. We understand that this reduction is very important in the case of NMT, since training an NMT system can take up to several weeks. We demonstrate that with adequate DS, we can reduce execution time from 11 days to 17 hours, while simultaneously improving the translation quality achieved by a model trained with all the data available.

Acknowledgments

Work partially supported by MINECO under grant DI-15-08169 and by Sciling under its R+D programme.

References

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proc. of EMNLP*, pages 355–362.

Biçici, E. and Yuret, D. (2011). Instance selection for machine translation using feature decay algorithms. In *Proc. of WMT*, pages 272–283. ACL.

Chen, B. and Huang, F. (2016). Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proc. of SIGNLL-CoNLL*, pages 314–323.

Chen, B., Kuhn, R., Foster, G., Cherry, C., and Huang, F. (2016). Bilingual methods for adaptive training data selection for machine translation. In *Proc. of AMTA*, pages 93–106.

Chinea-Rios, M., Peris, A., and Casacuberta, F. (2017). Adapting neural machine translation with parallel synthetic data. In *proc. of WMT*, pages 138–147.

Chinea-Rios, M., Sanchis-Trilles, G., and Casacuberta, F. (2016). Bilingual data selection using a continuous vector-space representation. In *Proc. of SPR-SSPR*, pages 95–106. Springer.

Farajian, M. A., Turchi, M., Negri, M., and Federico, M. (2017). Multi-domain neural machine translation through unsupervised adaptation. In *proc. of WMT*, pages 127–137.

Gao, J., Goodman, J., Li, M., and Lee, K.-F. (2002). Toward a unified approach to statistical language modeling for chinese. *ACM*, pages 3–33.

Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., and Casacuberta, F. (2012). Does more data always yield better translations? In *Proc. of EACL*, pages 152–161.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, pages 1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *arXiv preprints*, arXiv:1701.02810.

Lu, Y., Huang, J., and Liu, Q. Improving statistical machine translation performance by training data selection and optimization. In *Proc. of EMNLP-CoNLL*, pages 343–350.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprints*, arXiv:1310.4546.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proc. of ACL*, pages 220–224.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Peris, Á., Chinea-Ríos, M., and Casacuberta, F. (2017). Neural networks classifier for data selection in statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):283–294.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprints*, arXiv:1508.07909.

Shterionov, D., Nagle, P., Casanellas, L., Superbo, R., and ODowd, T. (2017). Empirical evaluation of nmt and pbsmt quality for large-scale translation production. In *Proc. of EAMT*, pages 75–80.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proc. of NIPS*, volume 27, pages 3104–3112.

Toral, A. and Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. *arXiv preprint arXiv:1701.02901*.

van der Wees, M., Bisazza, A., and Monz, C. (2017). Dynamic data selection for neural machine translation. *arXiv preprint arXiv:1708.00712*.

Translating Short Segments with NMT: A Case Study in English-to-Hindi

Shantipriya Parida **Ondřej Bojar**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{parida,bojar}@ufal.mff.cuni.cz

Abstract

This paper presents a case study in translating short image captions of the Visual Genome dataset from English into Hindi using out-of-domain data sets of varying size. We experiment with three NMT models: the shallow and deep sequence-to-sequence and the Transformer model as implemented in Marian toolkit. Phrase-based Moses serves as the baseline.

The results indicate that the Transformer model outperforms others in the large data setting in a number of automatic metrics and manual evaluation, and it also produces the fewest truncated sentences. Transformer training is however very sensitive to the hyperparameters, so it requires more experimenting. The deep sequence-to-sequence model produced more flawless outputs in the small data setting and it was generally more stable, at the cost of more training iterations.

1 Introduction

In recent years, neural machine translation (NMT) systems have been gaining more popularity due to their improved accuracy and even more fluency compared with "classical" statistical machine translation systems such as phrase-based MT (PBMT), see e.g. the shared tasks of WMT and IWSLT (Bojar et al., 2017; Cettolo et al., 2017). The major advantages of NMT include the consideration of the entire sentence, capturing similarity of words, and the capacity to learn complex relationships between languages. At the same time, it has been observed that NMT is more sensitive to the shortage of or noise in the parallel training data (Koehn and Knowles, 2017).

Our goal is to create the Hindi version of Visual Genome (Krishna et al., 2017).[1]

Hindi, with 260 million speakers, is the fourth most widely spoken language on the planet (after Chinese, Spanish and English). Hindi is a morphologically rich language (MRL), with e.g. the gender category being reflected in the forms of nouns, verbs and also adjectives (Sreelekha S and Bhattacharyya, 2017). The structural and morphological differences between English and Hindi result in translation difficulties (Tsarfaty et al., 2010).

Visual Genome is a dataset of images, captions and relations. As such, it is potentially useful for many NLP and image processing applications. The Hindi version would allow to exploit this dataset e.g. to create Hindi image labellers or other practical tools.

The textual part of Visual Genome consists primarily of short sentences or noun phrases that were manually attached to rectangular regions in an input image. In the current version, Visual Genome contains 108K distinct images with 5.4 million such labelled regions in total. On average, an image is thus associated with 50 text segments. Text segments can repeat across images and indeed, when de-duplicated, the set of unique strings reduces to 3.15 million unique segments.

Even with this de-duplication, this set remains too big to be translated manually. It is thus natural to attempt to translate this dataset automatically and in this paper, we are trying to find the best base-

[1] http://visualgenome.org/

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 229–238
Alacant, Spain, May 2018.

line translation. In the future, we want to include also information available in the context of each of the labels: either the text descriptions of nearby regions or directly the visual information in a form of multi-modal translation (Matusov et al., 2017; Calixto et al., 2012; Huang et al., 2016).

The paper is organized as follows. Section 2 reviews related work on neural MT and English-Hindi translation. Section 3 describes our experimental setting: data, models and their parameters. Section 4 provides automatic and manual evaluation of the translations and Section 5 discusses the results in closer detail. We conclude in Section 6.

2 Related Work

Singh et al. (2017) have compared two neural machine translation models, convolutional sequence to sequence (ConvS2S) and recurrent sequence to sequence (RNNS2S) for English↔Hindi machine translation task. They have used the IITB corpus for training (see Section 3.1) and also for development and test data. The RNNS2S model was trained using Nematus (Sennrich et al., 2017) and ConvS2S using Fairseq (Gehring et al., 2017), an open source library developed by Facebook. In their evaluation, ConvS2S was better when targetting English (BLEU scores: RNNS2S: 11.55, ConvS2S: 13.76) but RNNS2S was better when targetting Hindi (BLEU scores: RNNS2S: 12.23, ConvS2S: 11.73). As our experiment scope is limited to English to Hindi translation, we have not tried the ConvS2S.

Wang et al. (2017) use the encoder-decoder framework with attention (Bahdanau et al., 2015) for their submission to the Workshop on Asian Translation (WAT) 2017 shared task and observe considerable gains for English-to-Hindi compared to PBMT. Similarly to other works, they benefit from subword units (Sennrich et al., 2016a) and back-translation (Sennrich et al., 2016b), as well as model ensembling.

Agrawal and Misra Sharma (2017) evaluate English-Hindi translation quality using several variants of RNN-based neural network architecture and basic units (LSTMs, Hochreiter and Schmidhuber, 1997, and GRUs, Cho et al., 2014b), including the attention mechanism by Bahdanau et al. (2015) and more layers in the encoder and decoder. The bi-directional LSTM model with four layers and attention performs best.

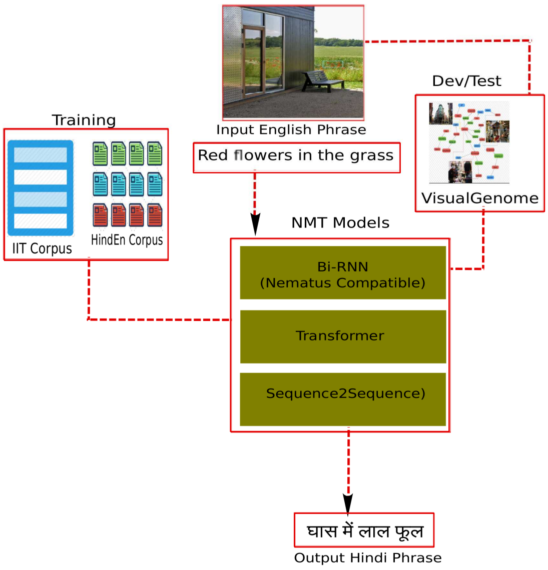


Figure 1: Overall experimental setting.

The early models of NMT have suffered from lower translation quality for long sentences, see e.g. Cho et al. (2014a) and Bahdanau et al. (2015). A recent experiment by Beyer et al. (2017) has however suggested that NMT can perform worse than PBMT also for short segments (insignificantly). It is thus natural to evaluate the effect in our particular setting.

We note that monolingual data plays an important role in boosting the performance of the translation in both PBMT (Brants et al., 2007; Bojar and Tamchyna, 2011) and NMT (Sennrich et al., 2016b; Domhan and Hieber, 2017). We leave these experiments for future work because we would first need to find or select Hindi texts closely matching to the domain of Visual Genome texts.

3 Experiments

The overall framework of our work is shown in Figure 1. The targeted dataset is English text descriptions from Visual Genome but no similar or related data is available in Hindi. So far, we thus used Visual Genome only to select the development and the test set.

We experimented with two parallel corpora as our training data, HindEnCorp and IITB Corpus (see Section 3.1), three NMT models and the PBMT baseline (Section 3.2).

We used the experiment management tool Eman (Bojar and Tamchyna, 2013)[2] for organizing and running the experiments.

[2] http://ufal.mff.cuni.cz/eman

Set	#Sentences	#Tokens	
		En	Hi
Train (HindEnCorp)	273.9k	3.8M	5.6M
Train (IITB)	1492.8k	20.8M	31.4M
Dev (Visual Genome)	898	4519	6219
Test (Visual Genome)	1000	4909	6918

Table 1: Statistics of our data.

Parameter	Bi-RNN	S2S	Transformer
beam-size	12	12	12
dec-cell	gru	lstm	–
dec-cell-base-depth	2	4	–
dec-cell-high-depth	1	2	–
dec-depth	1	4	6
decay-inv	–	–	16000
dim-emb	512	512	512
dim-rnn	1024	1024	1024
dropout-rnn	0.2	0.2	–
dropout-src	0.1	0.1	–
dropout-trg	0.1	0.1	–
early-stopping	10	–	–
enc-cell	gru	lstm	–
enc-cell-depth	1	2	–
enc-depth	1	4	6
enc-type	bidirectional	alternating	–
exponential-smoothing	–	0.0001	–
heads	–	–	8
label-smoothing	–	–	0.1
learning-rate	0.0001	0.0001	0.0003
max-length	50	50	100
normalize	–	–	0.6
optimizer	adam	adam	adam
transformer-dim-ffn	–	–	2048
transformer-dropout	–	–	0.1
transformer-dropout-attention	–	–	0
transformer-postprocess	–	–	dhn
warm-up	–	–	16000

Table 2: Model configurations.

3.1 Dataset Description

This section describes the processing and usage of the training and development data. We have used HindEnCorp (Bojar et al., 2014) as the training dataset which contains 274k parallel sentences. Additionally, we have explored the very recent "IIT Bombay English-Hindi Parallel Corpus" (Kunchukuttan et al., 2018) which is supposedly the largest publicly available English-Hindi parallel corpus. This corpus contain 1.49 million parallel segments and it includes HindEnCorp.

The development and test sentences were extracted from the Visual Genome. The original dataset contains images and their region annotations and several other formally captured types of information (objects, attributes, relationships, region graphs, scene graphs and question answer pairs). We built our dataset by extracting only the region descriptions, which are generally short sentences or phrases. We selected the development and test segments randomly and prepared the corresponding Hindi translation by manually correcting Google Translate outputs.

The training and test sets sizes are shown in Table 1. Note that the token counts considerably differ from those reported in the corpus descriptions. Here we report the token counts as obtained by the Moses tokenizer and used in all our experiments.

3.2 MT Models Tested

One of the current most efficient NMT toolkits is Marian[3] (Junczys-Dowmunt et al., 2016), which is a pure C++ implementation of several popular NMT models. All our experiments thus use Marian models.

3.2.1 Marian's nematus Model (Bi-RNN)

The common baseline NMT architecture is the (shallow) attentional encoder-decoder of Bahdanau et al. (2015). A particularly popular implementation of this model is available in the Nematus toolkit (Sennrich et al., 2017),[4] which adds some implementation differences such as a different initial hidden state, a different RNN cell and several others.

Marian implements both the training and inference with the Nematus (Sennrich et al., 2017) model and in fact, it can load models trained by the original Nematus.

We call this setup "Bi-RNN" in the following and use it only in shallow (depth 1) setting.

3.2.2 Marian's Sequence-to-Sequence (s2s) Model

A more advanced variation of the RNN-based model allows to use deeper layers in both decoder and encoder and it also differs from the original Nematus model in several features, such as a different layer normalization (Sennrich et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017).

We denote this model "S2S" in the following and use it only in the deep (depth 4) setting.

3.2.3 Marian's transformer Model

The Transformer model (Vaswani et al., 2017) has been recently proposed to avoid the expensive training of RNNs, relying on the attention mechanism.

As explored by Popel and Bojar (2018) with the

[3] http://github.com/marian-nmt/marian

[4] http://github.com/EdinburghNLP/nematus

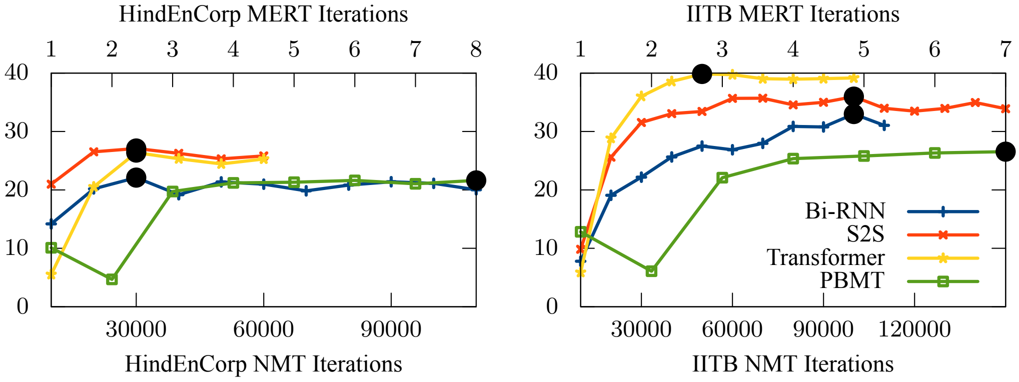


Figure 2: Learning curves in terms of BLEU on dev set. The big black dots indicate which iteration was used for test set translation and evaluation.

original Google implementation,[5] the model can be more difficult to train but it will likely outperform other architectures in both training time and final translation quality. Indeed, we needed to try 9 different configuration settings for Transformer before we got any reasonable performance, compared to just 3 for S2S and 1 for Bi-RNN.

Marian's implementation should be fully compatible with the original Google one.

The configuration parameters used for training of the models are shown in Table 2.

3.2.4 Common Settings

In all NMT experiments, we used the same BPE (Sennrich et al., 2016a), with 30k merges, joint for English and Hindi and extracted from HindEnCorp only. We also tried to extract the BPE from the respective training corpus (i.e. IITB for IITB models) but the performance was lower, perhaps due to domain differences between the corpora. The HindEnCorp BPEs are thus used in all experiments reported here.

3.2.5 Moses PBMT Baseline

For the purposes of comparison, we also train Moses (Koehn et al., 2007) phrase-based MT system with a 5-gram LM and a lexicalized reordering model, trained with the standard MERT optimization towards BLEU. The alignment is based on lowercase tokens, stemmed to the first 4 characters only.

4 Results

Figure 2 presents the learning curves for all the models evaluated on the development set using the BLEU score (Papineni et al., 2002). (PBMT training is displayed in terms of MERT iterations on the secondary x axis.)

		Bi-RNN	S2S	Transf.	PBMT
HindEnCorp	BLEU	20.68	**26.45**	23.91	20.61
	chrF3	32.30	**39.52**	36.36	36.49
	nCDER	34.04	**40.91**	38.26	32.71
	nCharacTER	12.27	18.47	23.12	**29.05**
	nPER	41.76	49.05	47.01	**50.40**
	nTER	29.63	**35.70**	33.52	24.78
IITB Corpus	BLEU	31.78	32.81	**38.31**	25.06
	chrF3	42.63	44.50	**51.08**	43.09
	nCDER	44.49	44.91	**51.78**	37.54
	nCharacTER	-14.76	-47.00	25.07	**37.55**
	nPER	51.86	52.04	**59.60**	55.17
	nTER	40.62	41.44	**49.05**	32.76

Table 3: Results on the test set, multiplied by 100. Best model according to each automatic metric in bold. Metrics with the prefix "n" were flipped (100 − score) to make better scores higher. The negative numbers for nCharacTER happen when the original CharacTER score is over 1.

For NMT, we validated the model every 10000 iterations and ran the training until the cross-entropy has not improved for 10 consecutive validations. For each model, we selected the iteration where the highest BLEU score was reached and translated the test set with this model.

4.1 Automatic Evaluation

Table 3 provides automatic scores of the models in several metrics (Papineni et al., 2002; Snover et al., 2006; Leusch and Ney, 2008; Popović, 2015; Wang et al., 2016).[6] We see that on the smaller HindEn-

[5] http://github.com/tensorflow/tensor2tensor

[6] Note that the exact scores are *heavily* dependent on the tokenization. We collect outputs from all our system after detokenization and tokenize if needed by the metric (chrF3 and CharacTER do not expect tokenized text). We report the scores when Moses tokenizer was used. Using e.g. the Hindi tokenization from IndicNLP, http://github.com/anoopkunchukuttan/indic_nlp_library, leads to sub-

Corp, S2S performs best except in CharacTER and PER where the outputs of PBMT score best. On the larger IITB Corpus, Transformer wins in all metrics except again CharacTER. We suspect that the different evaluation by CharacTER could be an artifact of the Devanagari script used in Hindi.

PER, position-independent error-rate, reflects the overlap of exact word forms used in the reference and the hypothesis, suggesting that PBMT performs reasonably well in terms of preserving words, although the fluency is probably worse.

It should be noted that the automatic scores can be affected by the fact that our test set was created by manual revision of Google Translate outputs. The underlying model of Google Translate is however unknown. Also, we have only one reference translation and it is well known that with more reference translations, automatic evaluations are more reliable (Finch et al., 2004; Bojar et al., 2013).

4.2 Manual Evaluation

To validate the automatic scoring, we manually annotated 100 randomly selected segments as translated by the NMT models.[7]

In this annotation, each annotated segment gets exactly one label from the following set:

Flawless for translations without any error (typesetting issues with diacritic marks due to different tokenization are ignored),

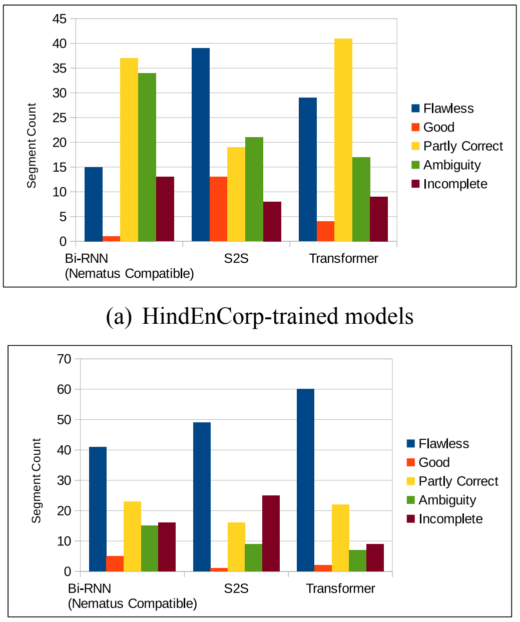
Good for translations which are generally OK and complete but need a small correction,

Partly Correct for cases where a part of the segment is correct but some words are mistranslated,

Ambiguity for segments where the MT system "misunderstood" a word's meaning, and

Incomplete for segments that run well but stop too early, missing some content words. This category also includes the relatively rare cases where the NMT model produced just a single word, unrelated to the source.

The results are summarized in Figure 3.



(a) HindEnCorp-trained models

(b) IITB-trained models

Figure 3: Manual evaluation summary.

The manual annotation generally confirms the automatic scores. On HindEnCorp, S2S has the highest number of Flawless segments and Bi-RNN performs worst, having the majority of outputs only Partly Correct and suffering most from Ambiguity.

On IITB, the performance of all the models is generally much better, with 40–60 of the 100 annotated segments falling into the Flawless category. Transformer is a clear winner here and S2S suffers from surprisingly many Incomplete segments.

Some translation samples are shown in Figure 4.

5 Analysis and Discussion

We assumed that PBMT may perform better on short segments. In order to test this assumption, we divided the 1000 test segments into 5 groups based on the source segment length. Group boundaries were chosen to achieve reasonably balance distribution and at least a minimal size for automatic scoring:

Source length:	1–3	4	5	6	7–12
Segment count:	73	380	282	165	100

Figure 5 plots BLEU scores evaluated on each group of segments separately. We see that our assumption does not hold and that there is no clear tendency in translation quality based on source sentence length. In the small data setting (HindEnCorp), PBMT scores well sentences of length 4 and

stantially lower scores, e.g. BLEU of 7 instead of 20. Fortunately, these BLEU scores correlate very well (Pearson of 0.94) with our scores.

[7] We excluded PBMT from this annotation because its BLEU scores were low; we are now reconsidering this decision given the good performance in PER.

Flawless:

A car on a street

सडक पर एक कार
Gloss: A car on a street

A white and yellow passenger car

एक सफेद और पीला यात ◌् री कार
Gloss: A white and yellow passenger car

White part of the chair

कुर ◌् सी का सफेद भाग
Gloss: White part of the Chair

Partly Correct:

A man wearing white shorts

एक आदमी सफेद शॉर ◌् ट पहनना
Gloss: A man put on white short
(output does not convey the intended
meaning in the target language)

Dog in a lake

इस झील में कुत ◌् ते
Gloss: Dogs in this lake
(grammar error: dog vs. dogs)

Ambiguity:

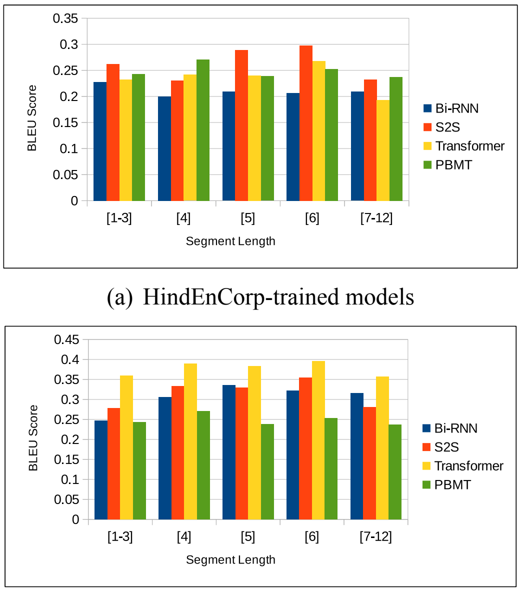
Faucet is above sink

फेसबुक सिंक से ऊपर है
Gloss: Facebook is above sink
(bad translation of the word "Faucet')

Green bean in soup

आत ◌् मा में हरा
Gloss: Spirit in green
(mis-translated words "bean", and "soup")

Figure 4: Sample segment translations and their manual classification.



(a) HindEnCorp-trained models

(b) IITB-trained models

Figure 5: Translation quality for groups of segments based on their source length.

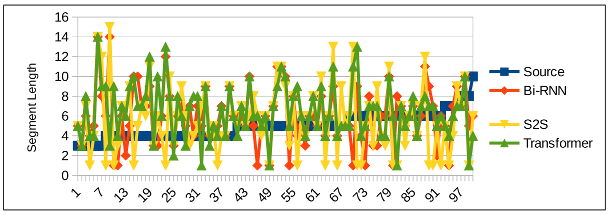


Figure 6: Source and candidate translation lengths for individual segments in the subset of 100 manually-evaluated segments. Segments are sorted by source length. The models were trained on the IITB corpus.

then on sentences over 7 words. In other cases, S2S wins. With the IITB training corpus, Transformer wins and PBMT loses across all lengths.

A generally interesting property of NMT is its ability to correctly predict the sentence length (Shi et al., 2016). We take a look at this by considering both the relation of our candidate translations with the source and with the reference.

Figure 6 plots the length of the translation for individual source segments sorted by length. We see that the target length varies a lot across segments and also different NMT models. In general, outputs are longer than sources but the length of the source is not really followed by any of the models.

We observed on the HindEnCorp training data that some of the NMT models tended to cut off sentences too short in early iterations. To examine this, we checked the difference in length between the candidate and the reference throughout the iterations. The distribution of length differences was however not skewed in any way and the only observable pattern was that the differences get smaller as the training progresses. We plot the differences for the converged runs over the whole 1000 segments in the test set in Figure 7. We see that all the NMT models are very similar, producing output slightly longer (peak at +2) than the reference. The PBMT is optimized well and the peak is located at zero difference between the candidate and reference length. The interesting pattern in NMT outputs of slightly fewer segments with odd differences (+1, +3 and +5) has still to be explained.

6 Conclusion

We have applied the state-of-the-art neural machine translation models and the phrase-based

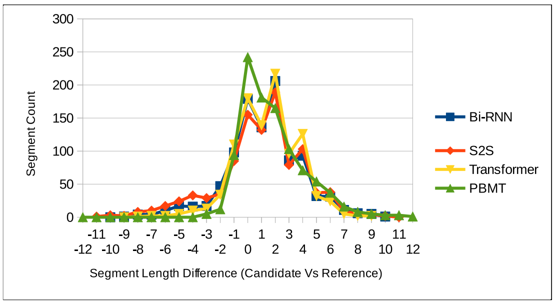


Figure 7: Segment length difference (candidate vs reference) of the IITB-trained models. The positive numbers indicate that candidate is longer than the reference.

baseline to English-to-Hindi translation. Our target domain were relatively short segments appearing in descriptions of image regions in the Visual Genome.

The results indicate that with smaller data (274k parallel segments, 3.8M English tokens), the deep sequence-to-sequence attentional model is the best choice, although the PBMT baseline seemed to perform well in two of the tested automatic metrics, CharacTER and PER. With large parallel data available, Transformer should be preferred and all NMT models clearly outperform PBMT. We have not yet explored the effect of adding monolingual data.

A deeper analysis has not revealed any difference in performance for shorter or longer segments, but the manual annotation suggested that the performance of NMT models varies across individual segments. The overall performance is thus perhaps too crude and it would be suboptimal to decide for a single model.

In the future, we will focus on the possibilities of multi-modal translation (Matusov et al., 2017; Calixto et al., 2012; Huang et al., 2016) to improve translation quality using the Visual Genome images or other contextual information available. Our ultimate plan is to release a machine-translated Hindi version of Visual Genome.

Acknowledgement

This work has been supported by the grants 18-24210S of the Czech Science Foundation, SVV 260 453 and “Progress” Q18+Q48 of Charles University, and using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

We thank Dr. Satyaranjan Dash and Miss Sneha Shrivastav for their support in Development and Test Data preparation.

References

Ruchit Agrawal and Dipti Misra Sharma. Building an Effective MT System for English-Hindi Using RNN’s. *International Journal of Artificial Intelligence & Applications*, 8:45–58, 09 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*, 2015.

Anne Beyer, Vivien Macketanz, Aljoscha Burchardt, and Philip Williams. Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data? In *Proceedings of EAMT User Studies and Project/Product Descriptions*, pages 41–46, Prague, Czech Republic, 2017.

Ondřej Bojar and Aleš Tamchyna. The Design of Eman, an Experiment Manager. *The Prague Bulletin of Mathematical Linguistics*, 99:39–58, 2013. ISSN 0032-6585.

Ondřej Bojar and Aleš Tamchyna. Improving Translation Model by Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. Scratching the Surface of Possible Translations. In *Proc. of TSD 2013*, Lecture Notes in Artificial Intelligence, Berlin / Heidelberg, 2013. Západočeská univerzita v Plzni, Springer Verlag.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. HindEnCorp — Hindi-English and Hindi-only Corpus for Machine Translation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 3550–3555, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large Language Models in Machine Translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007.

Iacer Calixto, Teófilo Emídio de Campos, and Lucia Specia. Images as Context in Statistical Machine Translation. In *In The 2nd Annual Meeting of the EPSRC Network on Vision & Language (VL'12)*, Sheffield, UK, 2012. EPSRC Vision and Language Network.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, pages 2–14, Tokyo, Japan, 2017.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics.

Tobias Domhan and Felix Hieber. Using Target-side Monolingual Data for Neural Machine Translation through Multi-task Learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1500–1505, 2017.

Andrew M. Finch, Yasuhiro Akiba, and Eiichiro Sumita. How Does Automatic Machine Translation Evaluation Correlate with Human Scoring as the Number of Reference Translations Increases? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC*, 2004.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based Multimodal Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation, WMT*, pages 639–645, 2016.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. An Exploration of Neural Sequence-to-Sequence Architectures for Automatic Post-Editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP*, pages 120–129, 2017.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, 2016.

Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL) Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123(1):32–73, May 2017. ISSN 1573-1405. doi: 10.1007/s11263-016-0981-7.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. The IIT Bombay English-Hindi Parallel Corpus. In *Proceedings of LREC*, 2018. In print.

Gregor Leusch and Hermann Ney. BLEUSP, INVWER, CDER: Three improved MT evaluation measures. In *NIST Metrics for Machine Translation Challenge*, Waikiki, Honolulu, Hawaii, October 2008.

Evgeny Matusov, Andy Way, Iacer Calixto, Daniel Stein, Pintu Lohar, and Sheila Castilho. Using Images to Improve Machine-Translating E-Commerce Product Listings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, pages 637–643, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Martin Popel and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, 2018.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*, 2016a.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*, 2016b.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL; Software Demonstrations*, pages 65–68, 2017.

Xing Shi, Kevin Knight, and Deniz Yuret. Why Neural Translations are the Right Length. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282, Austin, Texas, November 2016. Association for Computational Linguistics.

Sandhya Singh, Ritesh Panjwani, Anoop Kunchukuttan, and Pushpak Bhattacharyya. Comparing Recurrent and Convolutional Architectures for English-Hindi Neural Machine Translation. In *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP*, pages 167–170, 2017.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings AMTA*, pages 223–231, August 2006.

Sreelekha S and Pushpak Bhattacharyya. Role of Morphology Injection in SMT: A Case Study from Indian Language Perspective. *ACM Trans. Asian & Low-Resource Lang. Inf. Process.*, 17 (1):1:1–1:31, 2017. doi: 10.1145/3129208.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. Statistical Parsing of Morphologically Rich Languages (SPMRL) What, How and

Whither. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically-Rich Languages, SPMRL@NAACL-HLT*, pages 1–12, 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

Boli Wang, Zhixing Tan, Jinming Hu, Yidong Chen, and Xiaodong Shi. XMU Neural Machine Translation Systems for WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP*, pages 95–98, 2017.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. CharacTER: Translation Edit Rate on Character Level. In *ACL First Conference on Machine Translation (WMT)*, Berlin, Germany, August 2016.

Feature Decay Algorithms for Neural Machine Translation

Alberto Poncelas, Gideon Maillette de Buy Wenniger, Andy Way
ADAPT Centre, School of Computing,
Dublin City University, Dublin, Ireland
{firstname.lastname}@adaptcentre.ie

Abstract

Neural Machine Translation (NMT) systems require a lot of data to be competitive. For this reason, data selection techniques are used only for fine-tuning systems that have been trained with larger amounts of data. In this work we aim to use Feature Decay Algorithms (FDA) data selection techniques not only to fine-tune a system but also to build a complete system with less data. Our findings reveal that it is possible to find a subset of sentence pairs, that outperforms by 1.11 BLEU points the full training corpus, when used for training a German-English NMT system .

1 Introduction

In Statistical Machine Translation (SMT) it has been shown that having more data does not always lead to better results [Ozdowska and Way, 2009]. In fact, performance can increase by limiting the training data to a smaller but more relevant set [Eetemadi et al., 2015]. Neural Machine Translation (NMT) models in contrast are data-hungry, and perform better only with large amounts of available training data, in some cases underperforming SMT when low amounts of data are available [Östling and Tiedemann, 2017; Dowling et al., 2018]. However, the amount of training data required to make NMT work really well depends a lot on the domain of the training data and test set, and possibly also how similar they are. For certain training domains such as TED talks [Bentivogli et al., 2016] it has already been shown that even with fairly limited training sizes NMT can already outperform SMT by a large margin.

Larger training sets also introduce noise and require models to cover a larger number of words, whereas for practical reasons the vocabulary cannot be arbitrarily increased to facilitate these extra words. Consequently, training material that is not relevant for the test set risks wasting limited entries in the vocabulary on source words that are not relevant to the test domain. This is why domain adaptation has proven to be useful in NMT [Chu et al., 2017] by tailoring a model towards in-domain data.

While traditional Machine Translation (MT) approaches perform an inductive learning (i.e. learn a model from translated sentences in order to predict unseen examples), transductive learning approaches aim to identify the best training instance to predict the test set [Bianchini et al., 2016]. Models trained with sentences retrieved by transductive learning methods are tailored towards the test set. This is similar to the way many domain-adaptation methods adapt the training material to be suitable for a specific domain.

In this work, we use Feature Decay Algorithms (FDA), a transductive data selection method that has achieved good results in SMT and apply it to NMT. Our purpose is twofold: i) to question the widely held assumption that in Neural Machine Translation it is always better to use more data; and ii) to explore how a transductive data-selection technique like FDA should be applied in order to build models that outperform those built with all training data.

2 Related Work

Feature Decay Algorithms In our work, in order to extract a subset of the data, we use Feature Decay Algorithms [Biçici and Yuret, 2011; Biçici et al., 2015; Biçici and Yuret, 2015]. This is a method that uses the source side of the test set to select sentences that provide translation examples that are most relevant for this set. Furthermore, FDA aims to maximize the variability of these selected relevant *n*-grams in the training set by decreasing their value as they are being selected.

In order to do that, the features (*n*-grams extracted from the test set) are assigned an initial value, and each sentence of the training data is scored as the normalized (by dividing by the number of words) sum of the values of its *n*-grams. Then, the method iteratively selects the sentence with the highest score and adds it to the set of selected data (which initially is empty). After selecting a sentence, the values of the features contained in it are decreased according to the decay function. By default, the value of a feature f is defined as in (1):

$$decay(f) = init(f)0.5^{C_L(f)} \tag{1}$$

where $init(f)$ is the initial value and $C_L(f)$ is the count of the feature f in selected data.

The score of a sentence s at a particular iteration is the sum of the values of $C_L(f)$ of the features present in s, normalized by the length of s. The score of a sentence, using the default configuration in Equation (1), computed as in (2):

$$score(s) = \frac{\sum_{f \in F_s} 0.5^{C_L(f)}}{\text{\# words in s}} \tag{2}$$

where F_s is the set of features present in s.

FDA has proven to be useful in Statistical Machine Translation (SMT) Biçici [2013]; Poncelas et al. [2016, 2017]. Selecting a small subset of sentences from a parallel corpus using FDA is enough to train SMT systems that perform better than systems trained using the whole parallel corpus.

Neural Machine Translation We use neural machine translation [Kalchbrenner and Blunsom, 2013; Cho et al., 2014] in the form of sequence-to-sequence models [Sutskever et al.] based on recurrent neural networks [Bahdanau et al., 2014; Luong et al., 2015].

Fine-tuning A method of domain adaptation that has been used in NMT is “fine-tuning”, which involves using a pre-built NMT system and training it further for several epochs with smaller amounts of in-domain data.

Most works Luong and Manning [2015]; Freitag and Al-Onaizan [2016] first use general domain data for training a system, and then a different in-domain data set for fine-tuning. Chu et al. [2017] train a system using a resource-rich domain corpus, and then use a small domain corpus to fine-tune the system.

The approach of Li et al. [2016] is the closest to our work, as they use the information of the test set to retrieve the data for tuning. Li et al. [2016] use string similarity measures, such as Levenshtein [Levenshtein, 1966] or the cosine similarity of the average of the word embedding [Mikolov et al., 2013] in order to find sentences that are close to a given sentence of the test set.

An alternative technique of performing fine-tuning is proposed by van der Wees et al. [2017]. They train the model with a dataset that is varied for each epoch, instead of training a model with a fixed training set, and then tuning it with a subset or another dataset for the last epoch. The size of the data is decreased gradually, keeping the sentences that are more in-domain, weighted using Cross-Entropy Difference [Axelrod et al., 2011]. The size of the subset of a training data S at each epoch e is defined as Equation (3):

$$n(e) = \alpha \cdot |S| \cdot \beta^{\lfloor (e-1)/\eta \rfloor} \tag{3}$$

where α is the relative start size, the fraction of training data used for the first epochs (relative start size), β is the fraction of training data kept in the new selection (retention rate), and η is the number of epochs for which the same subset is used.

3 Data selection using the source-side of the test set

Using the source side of test examples is central to machine translation. For example, SMT effectively uses only those phrase-pairs that match the source side of a test sentence. Matching can be done implicitly, inside the decoder and during translation, or explicitly, by filtering the phrase-table with the source-side of the test set before passing it to the decoder.

The usage of the test set source side by FDA is conceptually not different from well-established lazy supervised learning methods such as K-nearest neighbors, and is also not fundamentally different from the usage of source information for matching phrase-pair selection by SMT grammar extractors/decoders [Lopez, 2008].

4 Research questions

Due to the good performance achieved by FDA in SMT, we want to explore whether the improvements also maintain in NMT. Accordingly, the first question we want to answer is:

- Is FDA also useful for selecting a subset of training data to train NMT models that perform better than models trained with the larger (full) training data without any selection?

In NMT there are several possible configurations for applying a data-selection techniques. One method is to build a complete model from scratch using just a subset of the data. Another way is to use fine-tuning to specialize an existing model.

On the top of that, there are several possibilities of how to tune a model: (i) Performing fine-tuning (and even in this option, there are several possibilities as we can choose different epochs of the model to tune); and (ii) perform a gradual tuning, where at each epoch the model is trained in using gradually smaller in-domain subsets

Due to the different configurations available, our second research question is:

- What configuration should be applied so that NMT model benefits the most from FDA techniques?

The test set may not always be accessible when building the NMT model. However, a system tuned for a document of one domain using FDA may be be useful for translating a different one if they share the domain.

- Can a model biased towards one test set using FDA be useful for translating a different test set in the same domain?

5 Experiments

In this work we have constructed a German-to-English NMT system using the Pytorch port [1] of OpenNMT [Klein et al., 2017] to train the models. According to the creators of OpenNMT [2] a good baseline for German-to-English WMT 2015 data is the one built with default parameters (2-layer LSTM with 500 hidden units, vocabulary size of 50002 and 50004 for source and target language, respectively) executed for 13 Epochs. The words in the output that are not in the vocabulary are replaced with the word in the source with the highest attention.

The data sets used in the experiments are based on the ones used in the work of Biçici [2013]:

- *Training data*: The training data provided in the WMT 2015 [Bojar et al., 2015][3] translation task setting a maximum sentence length of 126 words (4.5M sentence pairs, 225M words).

- *Validation data*: 5K randomly sampled sentences from development sets from previous years.

We extract subset of different sizes (100K, 200K, 500K, 1M and 2M sentences) from the training data with FDA using the test set from the WMT 2015 Translation Task. We use the default configuration of FDA (i.e. 3-grams as features, 0.5 as decay factor and 0 as decay exponent of 0). We perform several experiments building different NMT models using the training data and the data extracted with FDA. In order to answer the research questions in Section 4, models are built following different configurations:

- *FDA* experiments: Build NMT models from scratch, using only the output of FDA as training data.

- *BASE12+FDA* experiments: Fine-tune the last epoch of the baseline model with the output of FDA. Since the baseline is run for 13 epochs, we use the model of the 12th epoch.

- *BASE8+FDA* experiments: Fine-tune the the baseline model starting from the 8th epoch. We choose the 8th epoch not only because it is close to the middle stage of the training, but also because it is the point where fine-tuning and convergence of the model is initiated by starting the decay of the learning rate.

- *Gradual fine-tuning* experiments: Perform a gradual fine-tuning where the complete training data is used on the first epochs but gradually smaller sizes of training data are used thereafter. The sentences that are kept for the next iteration are the top sentences retrieved by FDA (being smaller at each epoch). The experiments are performed with the same configuration in the original work of van der Wees et al. [2017], using $\alpha = 0.5$, $\beta = 0.7$ and $\eta = 2$ in Equation (3).

In addition, we are interested in exploring whether the model trained on data retrieved by FDA using one document could also be useful for translating different documents in the same domain. We use the same models (trained with data using the test set of WMT 2015 as seed) for translating a different test set, the namely WMT 2014 [Bojar et al., 2014] news test set, which is in the same domain.

[1] https://github.com/OpenNMT/OpenNMT-py

[2] http://opennmt.net/Models/

[3] http://www.statmt.org/wmt15/translation-task.html

6 Results

In Table 1 and 2 we show several evaluation metrics: BLEU [Papineni et al., 2002], TER [Snover et al., 2006], METEOR [Banerjee and Lavie, 2005] and CHRF3 [Popovic, 2015]. These scores give an estimation of the quality of the output of the experiment when compared to a translated reference. In Table 2 we have also marked in bold the scores that outperform the baseline (Table 1) and computed the statistical significance (marked with an asterisk) with multeval [Clark et al., 2011] for BLEU, TER and METEOR when compared to the baseline at level p=0.01 using Bootstrap Resampling [Koehn, 2004].

	baseline
BLEU	0.2474
TER	0.5525
METEOR	0.2798
CHRF3	48.9473

Table 1: Results of the model trained with all available training data.

In the *baseline* column of Table 2 we see the scores of the translation of the test set (WMT 2015 document) using all training data. In the column *FDA* we present the results of the models built from scratch on different sizes of data retrieved by FDA (different subtables). As expected, an NMT model trained with small sets of data achieves worse results than the baseline. However, we discover that after selecting enough data, the system trained with less data outperforms the baseline. Using just 11% of sentences is enough to obtain better results (500K subtable) that are statistically significant for more than one evaluation metric. We observe the best results when selecting 2 million sentences, which is just 44.6% of the total number of sentences.

If we compare the models which have been fine-tuned (columns *BASE8+FDA*

	FDA	BASE8 +FDA	BASE12 +FDA
100K lines (2%)			
BLEU	0.1951	0.244	0.2458
TER	0.6243	0.5567	0.553
METEOR	0.245	0.2771	0.2793
CHRF3	42.9756	48.5617	48.7841
200K lines (4%)			
BLEU	0.2304	0.2445	**0.2479**
TER	0.5788	0.5562	**0.5523**
METEOR	0.2722	0.2773	**0.2804**
CHRF3	47.2747	48.5487	**49.0209**
500K lines (11%)			
BLEU	**0.2517***	**0.2478**	**0.2487**
TER	0.5601	0.5536	**0.5518**
METEOR	**0.2886***	0.2797	**0.2805**
CHRF3	**49.8314**	48.8575	**49.0866**
1M lines (22.3%)			
BLEU	**0.2560***	**0.2480**	**0.2475**
TER	**0.5497**	0.5533	**0.5524**
METEOR	**0.2886***	0.279	**0.2801**
CHRF3	**50.0932**	48.8372	48.9158
2M lines (44.6%)			
BLEU	**0.2585***	**0.2484**	0.2472
TER	**0.5454***	0.5543	**0.5522**
METEOR	**0.2894***	0.2795	**0.2802**
CHRF3	**50.0950**	48.8752	48.9247
Gradual fine-tuning			
BLEU	**0.2478**	-	-
TER	0.5588	-	-
METEOR	0.2798	-	-
CHRF3	48.8834	-	-

Table 2: Comparison of results of system trained in different sizes of training data retrieved by FDA

	baseline
BLEU	0.2502
TER	0.5558
METEOR	0.2824
CHRF3	49.5967

Table 3: Results of the model trained with all available training data using a different test set (WMT 2014 test set).

and *BASE12+FDA*), the scores obtained in *BASE12+FDA* experiments are better than *BASE8+FDA*. Almost all the evaluation metrics (the only exception is the BLEU score in subtable of 1M lines) are better when the fune-tuning is applied in the last epoch rather than in earlier stages. The *BASE12+FDA* experiment performs better than the baseline when using subsets of more than 200K sentences (we see in column *BASE12+FDA* that most of the scores are in bold). However none of them are statistically significant better than the baseline.

In the last subtable of Table 2 we show the performance of the model built using gradual fine-tuning. Even if it obtains a higher BLEU score the output is not statistically significantly better than the baseline at level p=0.01 for any of the metric.

We have seen that models trained with a subset of data perform better than those trained with all the data. As models built from scratch are not required to extract the words from the whole training data but only from the subset of sentences pairs relevant to the test set source, these are able to focus the limited vocabulary space more on those words that are relevant for the test set source. Tuning approaches in contrast preserve the initial vocabulary, which means they do not benefit from the more focused vocabulary training from scratch using FDA allows, which is one of the principles behind the working of FDA.

6.1 Further analysis: generalisation to additional test sets within the same domain

In order to explore whether the models built are also useful for translating another test set, we present Table 4. Here we see that the only scores that are statistically significantly better (marked with an asterisk) than the baseline (Table 3) are those of the *FDA* experiment when 2M sentences are selected. These results are consistent with those observed in Table 2.

Training models with smaller in-domain data sets achieves better results. In addition, fine-tuning applied in the last epoch causes the results to improve, as in Table 2. However, while we can still see improvements over the baseline for *BASE12+FDA* (numbers in bold in Table 4, column *BASE12+FDA* when 500K sentences or more are selected), none of these improvements are observed for the *BASE8+FDA* configuration/column in Table 4. Furthermore none of these improvements are statistically significant.

The main difference with Table 2 is that more training data is necessary to achieve results that are better than the baseline. This is because in this set of experiments, the vocabulary is not directly obtained from the test set but from a document in the same domain.

Note that in this set of experiments the seed used to extract in-domain data is the WMT 2015 test set which contains only 2169 lines. In future work, we want to explore whether the results can improve if we use documents with more sentences as seed.

As we argued in the introduction, using the source side of the test set is used even implicitly for fragment selection by all data-oriented (fragment based) methods, including SMT, though this may not be widely realized by practitioners in the field. But these results show that FDA can give improvements even if we omit the direct use of the source side of the test set, as is normally done by FDA.

	FDA	BASE8 +FDA	BASE12 +FDA
100K lines (2%)			
BLEU	0.1625	0.2419	0.2489
TER	0.6633	0.5623	0.5563
METEOR	0.2185	0.279	0.282
CHRF3	39.7603	48.9277	49.4858
200K lines (4%)			
BLEU	0.1982	0.2432	0.2501
TER	0.6157	0.5625	0.5566
METEOR	0.2483	0.2786	0.282
CHRF3	44.1265	48.7811	49.4807
500K lines (11%)			
BLEU	0.2307	0.2478	**0.2502**
TER	0.5759	0.5582	**0.5555**
METEOR	0.2711	0.2813	**0.2830**
CHRF3	47.752	49.2136	**49.6680**
1M lines (22.3%)			
BLEU	0.2458	0.2484	**0.2504**
TER	0.5662	0.558	0.5559
METEOR	0.2797	0.2814	**0.2828**
CHRF3	48.8866	49.2997	49.5829
2M lines (44.6%)			
BLEU	**0.2530***	0.2491	0.2501
TER	**0.5553**	0.556	0.5549
METEOR	**0.2849***	0.282	**0.2826**
CHRF3	**49.8117**	49.3921	49.5804
Gradual fine-tuning			
BLEU	0.245	-	-
TER	0.5644	-	-
METEOR	0.2787	-	-
CHRF3	48.8506	-	-

Table 4: Comparison of results of system trained in different sizes of training data retrieved by FDA using a different test set (WMT 2014 test set).

7 Conclusion and Future Work

In this work we have discovered that using FDA, it is possible to find a subset of data that can be used to train an NMT model that achieves better results than a model trained with all data. In particular, our best model, trained on 44.6% of the data improves over the baseline trained on the full training set, while also giving significant improvements on other metrics. Besides the significant improvement in translation quality, this also implies (in the chosen training regime, with 13 epochs and FDA after 8 epochs) a linear reduction in training time compared to the baseline system. For example, by reducing the training data by half for the last 8 epochs we use only 81% of the original training time[4].

In future work, we want to study the impact of the differences in vocabulary in each experiment. We also want to compare these results to different data-selection techniques or different variants of FDA (either using different values in the parameters, or different variants of the algorithm such us the one proposed in Poncelas et al. [2016, 2017]).

Acknowledgements

This research has been supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

This work has also received funding from the European Union's Horizon 2020 research and innovation programme under the

[4] If the last 5 epochs are trained using 50% of the data, the training time is $((8 * 1) + (5 * 0.5))/13 = 0.81$

Marie Skłodowska-Curie grant agreement No 713567.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, Ann Arbor, Michigan, 2005.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas, 2016.

Monica Bianchini, Anas Belahcen, and Franco Scarselli. A comparative study of inductive and transductive learning with feedforward neural networks. In *Conference of the Italian Association for Artificial Intelligence*, pages 283–293, Genova, Italy, 2016. Springer.

Ergun Biçici. Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 78–84, Sofia, Bulgaria, 2013.

Ergun Biçici and Deniz Yuret. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland, 2011.

Ergun Biçici and Deniz Yuret. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):339–350, 2015.

Ergun Biçici, Qun Liu, and Andy Way. ParFDA for fast deployment of accurate statistical machine translation systems, benchmarks, and statistics. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 74–78, Lisbon, Portugal, 2015.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, 2014.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisboa, Portugal, 2015.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua

Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 385–391, Vancouver, Canada, 2017.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, page 176–181, Portland, Oregon, 2011.

Meghan Dowling, Teresa Lynn, Alberto Poncelas, and Andy Way. Smt versus nmt: Preliminary comparisons for irish. *Technologies for MT of Low Resource Languages (LoResMT 2018)*, page 12, 2018.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29 (3-4):189–223, 2015.

Markus Freitag and Yaser Al-Onaizan. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*, 2016.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 67–72, Vancouver, Canada, 2017.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, 2004.

Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. One sentence one model for neural machine translation. *arXiv preprint arXiv:1609.06490*, 2016.

Adam David Lopez. *Machine Translation by Pattern Matching*. PhD thesis, College Park, MD, USA, 2008.

Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, 2015.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119, Daegu, South Korea, 2013.

Robert Östling and Jörg Tiedemann. Neural machine translation for low-resource languages. *arXiv preprint arXiv:1708.05729*, 2017.

Sylwia Ozdowska and Andy Way. Optimal Bilingual Data for French-English PB-SMT. In *Proceedings of the 13th Annual Meeting of the European Association for Machine Translation*, pages 96–103, Barcelona, Spain, 2009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

Alberto Poncelas, Andy Way, and Antonio Toral. Extending feature decay algorithms using alignment entropy. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 170–182, Seville, Spain, 2016.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. Applying n-gram alignment entropy to improve feature decay algorithms. *The Prague Bulletin of Mathematical Linguistics*, 108 (1):245–256, 2017.

Maja Popovic. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, 2015.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, 2006.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, 2017.

Investigating Backtranslation in Neural Machine Translation

Alberto Poncelas, Dimitar Shterionov, Andy Way,
Gideon Maillette de Buy Wenniger and **Peyman Passban**
School of Computing, DCU, ADAPT Centre
{firstname.lastname}@adaptcentre.ie

Abstract

A prerequisite for training corpus-based machine translation (MT) systems – either Statistical MT (SMT) or Neural MT (NMT) – is the availability of high-quality parallel data. This is arguably more important today than ever before, as NMT has been shown in many studies to outperform SMT, but mostly when large parallel corpora are available; in cases where data is limited, SMT can still outperform NMT.

Recently researchers have shown that back-translating monolingual data can be used to create synthetic parallel corpora, which in turn can be used in combination with authentic parallel data to train a high-quality NMT system. Given that large collections of new parallel text become available only quite rarely, backtranslation has become the norm when building state-of-the-art NMT systems, especially in resource-poor scenarios.

However, we assert that there are many unknown factors regarding the actual effects of back-translated data on the translation capabilities of an NMT model. Accordingly, in this work we investigate how using back-translated data as a training corpus – both as a separate standalone dataset as well as combined with human-generated parallel data – affects the performance of an NMT model. We use incrementally larger amounts of back-translated data to train a range of NMT systems for German-to-English, and analyse the resulting translation performance.

1 Introduction

Neural Machine Translation (NMT) [Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015] is a relatively new machine translation (MT) paradigm that has quickly become dominant in both academic and industry MT communities, achieving state-of-the-art results [Bentivogli et al., 2016; Bojar et al., 2016; Junczys-Dowmunt et al., 2016; Wu et al., 2016; Castilho et al., 2017; Shterionov et al., 2017] on a range of language pairs and domains. As a corpus-based paradigm, the translation quality strongly depends on the quality and quantity of the training data provided. In comparison to statistical machine translation (SMT) [Koehn, 2010], NMT typically requires more data to build a system with good translation performance [Koehn and Knowles, 2017].

In many use-cases, however, the amount of good-quality parallel data available is insufficient to reach the translation standard required. In such cases, it has become the norm to resort to back-translating freely available monolingual data [Sennrich et al., 2016b; Belinkov and Bisk, 2017; Domhan and Hieber, 2017] to create an additional synthetic parallel corpus [Sennrich et al., 2016b] for training an NMT model.

In this paper, we assert that this scenario has become the default in NMT without proper consideration of the merits of the approach. For example, Rarrick et al. [2011] present an algorithm for filtering noisy content from Web-scraped parallel corpora, in order to mitigate the "pollut[ion] [of the Web] with increasing amounts of machine-translated content". They note that their algorithm "is capable of identifying machine-

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 249–258
Alacant, Spain, May 2018.

translated content in parallel corpora for a variety of language pairs, and that in some cases it can be very effective in improving the quality of an MT system ... thus challenging the conventional wisdom in natural language processing that 'more data is better data'". Note too that Somers [2005] demonstrates backtranslation (or 'round trip' translation) to be an untrusted means of MT evaluation. In the same vein, Way [2013] notes that in order to show that MT is error-prone, "sites like Translation Party (`http://www.translationparty.com/`) have been set up to demonstrate that continuous use of 'back translation' – that is, start with (say) an English sentence, translate it into (say) French, translate that output back into English, ad nauseum – ends up with a string that differs markedly from that which you started out with".

Surely, then, no-one would argue that building an MT system – whether it be SMT or NMT – with *solely* synthetic data is a good idea; after all, the premise underpinning the paper by Rarrick et al. [2011] was that adding machine-translated data to high-quality human-translated training data *harms* performance. Nonetheless, NMT developers have been seduced into using back-translated data as a means of necessity; there is simply not enough authentic human-translated parallel data available to obtain high-quality results in all scenarios where we would like to deploy NMT. Somewhat surprisingly, despite the inherent problems noted above, adding back-translated data does help improve the quality of NMT output!

In this paper we set out to systematically test from the ground up the merits of back-translated data. We investigate three scenarios: (i) NMT systems trained on 'perfect' human-translated (authentic) data; (ii) using only back-translated (synthetic) data for training NMT systems; and (iii) NMT systems trained on a combination of human-translated and back-translated data. We systematically create multiple training corpora of increasing sizes, using training sets with authentic, synthetic and hybrid (authentic + synthetic) data.

For the hybrid case we increment the back-translated to human-generated data ratio and observe the quality of the resulting NMT systems. We aim to identify to what extent adding synthetic data improves (or harms) the translation capabilities of NMT systems. That is, we investigate whether backtranslation as a core technique in NMT has any limits; given that synthetic data is generated via another imperfect MT system, we hypothesise that NMT trained with 'imperfect' data will – at some point – undo any benefits from the 'perfect' (human-translated) data, and lead the NMT to degrade in performance.[1]

In all our experiments, we exploit data that is widely used in the academic community for researching the quality of MT. The datasets that we use in our experiments all come from the Translation Task of the Tenth Workshop on Machine Translation in 2015 (WMT 2015 [Bojar et al., 2015]).[2] To build our NMT systems we use OpenNMT-py (the pytorch port of OpenNMT [Klein et al., 2017]) with standard settings that allows for easy replicability of our experiments.

The remainder of the paper is structured as follows: Section 2 presents related work on using back-translated and other synthetic data in MT. Section 3 explains how back-translated data affects the training and quality of an NMT system. Our data is described in Section 4, and our experiments are outlined in Section 5. The results are summarised and analysed in Section 6. We conclude in Section 7 with final remarks and future work plans.

2 Related Work

Recent studies have shown different approaches to exploiting monolingual data to improve NMT. Gülçehre et al. [2015] present two approaches to integrate a language model trained on monolingual data into the decoder of an NMT system. Similarly, Domhan and Hieber [2017] focus on improving the decoder with monolingual data. While these studies show improved overall translation quality, they require changing the underlying neural network architecture. In contrast, backtranslation allows one to generate a parallel corpus that, consecutively, can be used for training in a standard NMT implementation as presented by Sennrich et al. [2016b]. Sennrich et al. [2016b] use 4.4M sentence pairs of authentic human-translated parallel data to train a baseline English → German NMT system that is later used to translate 3.6M German and 4.2M English target-side sentences. These are then mixed with the initial data to create human + synthetic parallel corpora which are

[1] Note that this should not be confused with the problem of overfitting, where the NMT system learns the training data very well but fails to generalize, with the result that it performs poorly on unseen data.

[2] `http://www.statmt.org/wmt15/`

then used to train new models. Due to the good results that were obtained, adding synthetic data has become a popular step in the NMT training pipeline [Sennrich et al., 2016c; Di Gangi et al., 2017; Lo et al., 2017].

Karakanta et al. [2018] use back-translated data to improve MT for a low-resource language, namely Belarusian (BE). They transliterate a high-resource language (Russian, RU) into their low-resource language (BE) and train a BE→EN system, which is then used to translate monolingual BE data into EN. Finally, an EN→BE system is trained with that back-translated data.

The work of Park et al. [2017] presents an analysis of models trained only with synthetic data. They train NMT models with parallel corpora composed of: (i) synthetic data in the source-side only; (ii) synthetic data in the target-side only; and (iii) a mixture of parallel sentences of which either the source-side or the target-side is synthetic.

Note too that in contrast to the efforts of Rarrick et al. [2011], backtranslation has been applied successfully in PBSMT. Bojar and Tamchyna [2011] use back-translated data to optimize the translation model of a PBSMT system and show improvements in the overall translation quality for 8 language pairs.

3 Issues involved in creating back-translated parallel data

Intuitively, MT models built using synthetic data should not perform well. A text translated by a machine can contain errors, so a model trained on such data may learn and replicate these mistakes. While Sennrich et al. [2016b] demonstrated that using back-translated data (in combination with human-translated data) during training can have a positive impact on the performance of the model, we hypothesize that the performance of the model will degrade if the synthetic data is overly dominant in the training set, i.e. the benefit of using high-quality authentic parallel data may be outweighed by the synthetic back-translated data.

We investigate our hypothesis through a systematic analysis of NMT models trained on different-sized parallel datasets containing increasing amounts of back-translated data. We acknowledge the plethora of factors that may impact such an analysis, e.g. vocabulary size, learning optimizer, learning rate, total amount of training steps/minibatches, etc. However, with this work we aim to provide a solid experimental baseline NMT set-up that would facilitate the analysis of the impacts of adding synthetic data to the training corpus. Furthermore, our analysis does not aim to compare the best possible systems, but rather NMT systems trained under the same conditions that would allow a fair comparison. In this regard, we train our systems with word-based dictionaries, rather than with dictionaries based on sub-word units e.g., using Byte-Pair Encoding (BPE) [Sennrich et al., 2016a], although the latter case generally leads to higher MT quality. Given two models of the same size (one trained on authentic and one on synthetic data) the same words can be split into sub-words differently. As such, the quality differences could be due to the sub-word units, learned from the specific data rather than the differences in the authentic and synthetic data.

Our evaluation builds a clearer picture of the progressive effects of adding synthetic data to the training corpus of NMT engines. To the best of our knowledge, such an analysis has not been performed at the time of writing.

Furthermore, we compare NMT systems built on authentic-only data to systems built on synthetic-only data and put the two extremes to a test. We hypothesise that only synthetic data will not be enough to train an NMT system with good performance due to the errors mediated by the initial MT system used to generate that data. However, our results are more than a little surprising. We present detailed analysis of our empirical results in Section 6.

4 Data

For the scope of this work, we use the German–English parallel data of the WMT 2015 Translation task [Bojar et al., 2015]. This corpus is shuffled, tokenized, truecased and cleaned (removing sentences of length over 126 words). In total, it contains 4.48M sentence pairs (225M words).

In order to explore the effects of back-translated data, we use human-translated (authentic) and back-translated (synthetic) data in three possible configurations:

- Authentic data only: Models are trained using authentic data only. Such models provide a baseline that any other model can be compared to. This is the baseline scenario for quality of data. Furthermore, such models represent a use-case where an industry partner supplies authentic data to MT engineers in order to build an NMT system.

- Synthetic data Only: Models are built using back-translated data only. Such models represent the case where no parallel data is available but monolingual data can be translated via an existing MT system and provided as a training corpus to a new NMT system. Such cases appear as the other extreme, or the worst-case scenario for quality of data. They reflect resource limitations, either due to the physical unavailability of data, i.e. low-resource languages, or due to economic reasons. Using synthetic data only might also be an option in cases where a high-quality model trained on real data is available, but the translation task is on a very different domain than the training data. In this case using the high-quality model to back-translate domain-specific monolingual target data, and then building a new model with this synthetic training data, might be useful for domain adaptation.
- Hybrid data: Models are built using a base dataset of 1M authentic sentence pairs combined with differing amounts of back-translated data. This is the most interesting scenario (similar to Sennrich et al. [2016b]) which allows us to trace the changes in quality with increases in synthetic-to-authentic data ratio.

All the models that we built are evaluated using the same test set. This test set is provided by WMT 2015 news translation task. It consists of 2169 sentences from the news domain. These sentences have also been tokenized and truecased.

5 Experimental set-up

We train sequence-to-sequence NMT models [Sutskever et al., 2014] based on recurrent neural networks with an attention mechanism [Bahdanau et al., 2015; Luong et al., 2015]. The NMT framework we use is OpenNMT [Klein et al., 2017] and in particular its pytorch[3] port.

Our set-up follows the OpenNMT guidelines,[4] that indicate that the default training configuration is reasonable for training a German-to-English model on WMT 2015 data.

We acknowledge the multitude of parameters and values that one can tweak in the set-up of an NMT system, leading to systems with significantly different performance. Moreover, the choice of these parameters often depends on the training data. In our experiments, however, we have focused on a static NMT set-up, where the different parameters (e.g. the NMT learning optimizer, number of epochs, etc.) are common for all systems we train. The decision on our set-up is based on two factors: (i) by limiting the variability of parameters, we can more easily investigate the effects of back-translated data by directly comparing the translation quality of the resulting NMT systems; and (ii) while certain new architectures such as *Transformer* [Vaswani et al., 2017] or different settings might obtain even better results, our goal here is not to build the absolutely best possible systems, but rather use configurations that are representative of what is used in the field and allow easy replication. Specifically, we use a 2-layer LSTM [Hochreiter et al., 1997] with 500 hidden units, a vocabulary size of 50,002 for the source language and 50,004 for the target language. A model is trained for 13 epochs, using the stochastic gradient descent learning optimizer and a batch size of 64. Any unknown words in the translation are replaced with the word in the source language that has the highest attention.

We first trained a baseline $DE \rightarrow EN$ model on $1,000,000$ parallel sentences of authentic data (*base dataset*) and a baseline $EN \rightarrow DE$ model on the same data set with source and target sides swapped around. The latter model is used for back-translation to create *synthetic dataset*s. We found that using 1M sentences to train the model was sufficient for 'good enough' translations. To determine this, we performed preliminary tests that involve human evaluation alongside automatic metrics (on a random sample of the outputs) with models trained on other data sizes.[5] When performing backtranslation, we also replace any unknown words with the word in English (the source language when performing the backtranslation) having the highest attention. We used this engine to then back-translate different portions of our original data set that we then used as parallel training data in two different scenarios: (i) by itself, i.e. synthetic data only, and (ii) in combination with the authentic data used to train the first engine, i.e. the hybrid models, as defined in Section 4.

[3] http://pytorch.org

[4] http://opennmt.net/Models/

[5] These experiments go beyond the scope of this work and are not included in the current paper.

To make our comparison fair, we defined two cases of authentic data. The first one starts with the first 1,000,000 sentences and grows incrementally (adding 500,000 parallel sentences each time) until it contains 3,500,000 sentences, i.e. ranging between the 1^{st} and the $3{,}500{,}000^{th}$ sentence. We denote these sets as $auth_{0+}$. The *hybr* data sets are composed of the 1^{st} 1,000,000 authentic sentences, combined with back-translated data for each following subset of 500,000 sentences.

In the second case, the authentic data sets start from the $1{,}000{,}000^{th}$ sentence. The first one contains 1,000,000 sentences; the next ones increment with 500,000 additional authentic sentences with the last one ranging between the $1{,}000{,}000^{th}$ to the $4{,}480{,}000^{th}$ sentence. These sets we refer to as $auth_{1+}$. The *synth* data sets are simply the back-translated data sets from the $auth_{1+}$ category.

In this way we compare engines trained on exactly the same original data – $auth_{0+}$ to *hybr* and $auth_{1+}$ to *synth* – which in one case has been partially or fully back-translated.

In Table 1 we present the percentage of tokens (words, numbers and other symbols) of the test set that are covered by the vocabularies we use to build our models.

data size	$auth_{0+}$	*hybr*	$auth_{1+}$	*synthetic*
1M	67.03%	-	66.35%	60.81%
1.5M	67.15%	66.14%	66.44%	60.93%
2M	67.11%	65.10%	66.41%	60.97%
2.5M	67.25%	64.60%	66.36%	61.03%
3M	67.30%	64.15%	66.47%	60.98%
3.5M	67.25%	63.77%	66.55%	61.01%

Table 1: Coverage of the vocabularies (the top-50000 words) on the tokens in the test set.

6 Results

Tables 2 and 3 show the evaluation scores of the models we trained for the authentic-to-hybrid and authentic-to-synthetic cases, respectively. We use a number of common evaluation metrics – BLEU [Papineni et al., 2002], TER [Snover et al., 2006], METEOR [Banerjee and Lavie, 2005], and CHRF [Popovic, 2015] – to give a more comprehensive estimation of the comparative translation quality. With the exception of TER, the higher the score, the better the translation is estimated to be; for TER, being an error metric, the lower the score, the better the quality. For comparing the models of the same size, we have also computed the statistical significance (marked with an asterisk) using multeval [Clark et al., 2011] for BLEU, TER and METEOR at level p=0.01 using Bootstrap Resampling [Koehn, 2004].

		1M auth.	-
1M lines	BLEU	0.2278	-
	TER↓	0.5748	-
	METEOR	0.269	-
	CHRF1	48.7336	-
		1.5M auth.	1M auth. + 0.5M synth.
1.5M lines	BLEU↑	0.2347	0.2378
	TER↓	0.5702	0.5681
	METEOR↑	0.2735	0.2751
	CHRF1↑	49.2973	49.5145
		2M auth.	1M auth. +1M synth.
2M lines	BLEU↑	0.2382	0.2421
	TER↓	0.5646	0.5644
	METEOR↑	0.2755	0.2771
	CHRF1↑	49.6164	49.6818
		2.5M auth.	1M auth. + 1.5M synth.
2.5M lines	BLEU↑	0.2419	0.242
	TER↓	0.5592	0.5622
	METEOR↑	0.2786	0.2784
	CHRF1↑	50.015	49.8781
		3M auth.	1M auth. + 2M synth.
3M lines	BLEU↑	0.2446	0.2442
	TER↓	0.5572	0.5621
	METEOR↑	0.2792	0.2785
	CHRF1↑	50.1999	49.9244
		3.5M auth.	1M auth. + 2.5M synth.
3.5M lines	BLEU↑	0.2435	0.2413
	TER↓	0.5586	0.5651
	METEOR↑	0.2788	0.277
	CHRF1↑	50.0785	49.584

Table 2: Results of models using human-translated or authentic data and back-translated or synthetic data from the $auth_{0+}$ and *hybr* sets.

In Figures 2 and 1 we illustrate how the BLEU and METEOR scores of our models (trained on authentic, synthetic and hybrid data) change with increases in the training data.

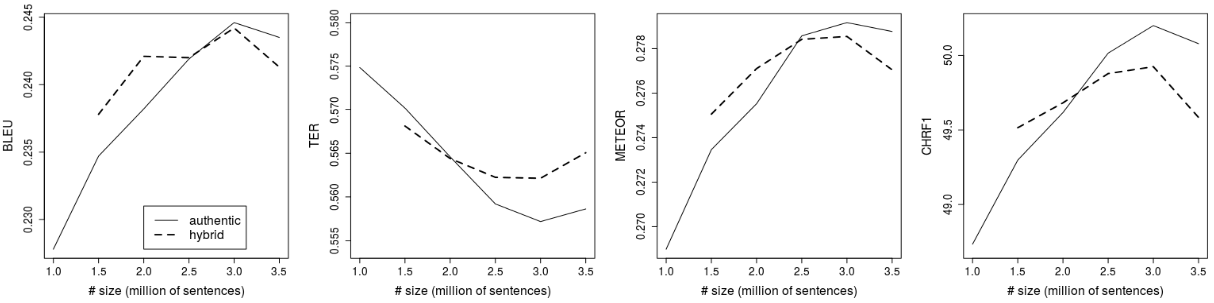


Figure 1: Quality scores of NMT systems trained with different sizes of training data from the *auth*$_{0+}$ and *hybr* sets.

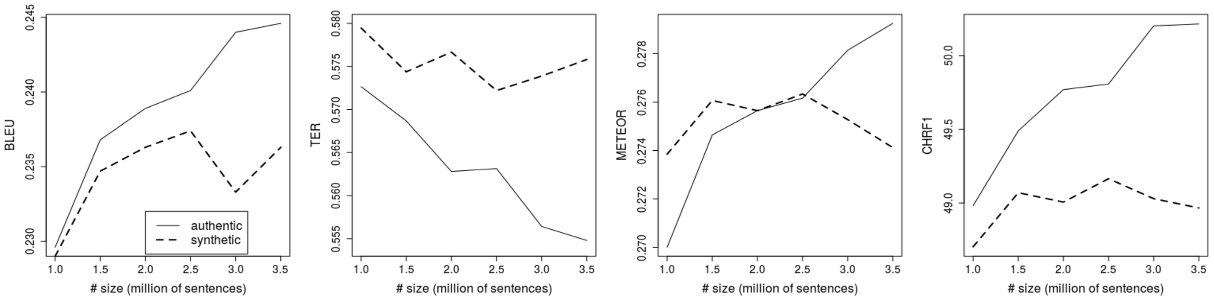


Figure 2: Quality scores of NMT systems trained with different sizes of training data from the *auth*$_{1+}$ and *synth* sets.

6.1 Authentic Data Models

In Tables 2 and 3, we see that, as expected, building NMT systems with increasingly larger amounts of human-translated data improves performance: from a BLEU score of 0.2278 with 1M sentence pairs, to the best score of 0.2446 with 3M sentence pairs. This is an absolute improvement of 0.0168, or 7.4% relative. We do, however, see a slight drop when we build our NMT system with 3.5M sentence pairs. All these findings are corroborated by the other three MT evaluation metrics.

6.2 Hybrid Data Models

According to the results summarised in Table 2 and Figure 1, the benefits of adding back-translated data presented in Sennrich et al. [2016b] are maintained in our experiments. We see that the hybrid model where 0.5M synthetic sentences are added in the training data (i.e. *1M auth + 0.5M synth* column in Table 2) performs better than the model built with 1M human-translated sentences. In fact, the same-sized hybrid model also outperforms the authentic-only model built with 1.5M sentence pairs.

Adding more and more synthetic data to the training set of an NMT systems causes BLEU scores to rise, as expected, with the best combination comprising 3M sentence pairs (1M authentic and 2M synthetic sentence pairs), which achieves a BLEU score of 0.2442, 0.0066 points absolute better than the smallest hybrid model, a relative improvement of 2.8%.

We see in column *hybr* of Table 1 that the coverage of the hybrid models is not as high as for those built with authentic data only, but in all cases they are higher than for the synthetic-only datasets. We observe that the bigger the data set, the lower the coverage is. We expect that as more synthetic data is added, the more its vocabulary starts to dominate, pushing out words that are more frequent in real parallel data, but less frequent in synthetic data. Accordingly, we expect the coverage of hybrid models to tend to converge to the values of the synthetic models.

Figure 1 shows how the quality of the hybrid models increases the more synthetic data is added. For smaller models, the slopes of the hybrid and authentic models are similar. However, the slope becomes less steep for models trained with 2M sentences or more, as in hybrid datasets with 2M sentence pairs half of it contains synthetic data.

6.3 Synthetic Data Models

Earlier in the paper, we suggested that no-one would set out to build an NMT system using solely synthetic data. However, our results show this to

		1M auth.	1M synth.
1M lines	BLEU↑	0.2296	0.2290
	TER↓	0.5726*	0.5795
	METEOR↑	0.2700	0.2738
	CHRF1↑	48.9829	48.7035
		1.5M auth.	1.5M synth.
1.5M lines	BLEU↑	0.2368*	0.2347
	TER↓	0.5687	0.5744
	METEOR↑	0.2746	0.2761
	CHRF1↑	49.4900	49.0705
		2M auth.	2M synth.
2M lines	BLEU↑	0.2389*	0.2363
	TER↓	0.5628*	0.5767
	METEOR↑	0.2756	0.2756
	CHRF1↑	49.7702	49.0069
		2.5M auth.	2.5M synth.
2.5M lines	BLEU↑	0.2401*	0.2374
	TER↓	0.5631*	0.5722
	METEOR↑	0.2762	0.2763
	CHRF1↑	49.8079	49.1656
		3M auth.	3M synth.
3M lines	BLEU↑	0.2440*	0.2333
	TER↓	0.5564*	0.5739
	METEOR↑	0.2781*	0.2753
	CHRF1↑	50.2028	49.0301
		3.5M auth.	3.5M synth.*
3.5M lines	BLEU↑	0.2446*	0.2363
	TER↓	0.5548*	0.5758
	METEOR↑	0.2792*	0.2741
	CHRF1↑	50.2159	48.9671

Table 3: Results of models using human-translated or authentic data and back-translated or synthetic data from the *auth*$_{1+}$ and *synth* sets.

be far from the crazy idea it seemed at the outset (see Table 3 and Figure 2). Using 1M sentence pairs of synthetic-only data (the first of the *synth* data sets), we obtain a BLEU score of 0.229, which continues to rise as we add more synthetic data, achieving the best BLEU score of 0.2363 with 3.5M sentence pairs. This is an absolute improvement of 0.0073, or 3.2% relative. Looking at the other metrics, the picture is rather more mixed; TER, METEOR and CHRF follow a more steady tendency[6].

It is clear, however, that the difference between the quality of engines trained on synthetic and authentic data is rather small. Moreover, the authentic and synthetic data sets of 1,000,000 sentences result in engines where the latter one actually performs better in terms of METEOR. However, even if smaller models built using synthetic data only can perform very close to the level of authentic-only models, it does not appear to be scalable, as the differences in the quality metrics between the two types of engines increase with larger data sizes, i.e. if we look at Figure 2, the quality of the models trained with synthetic data have a relatively lower increase in quality when more back-translated sentences are added.

From column *synth* of Table 1 we notice that the coverage of models built using synthetic data does not increase when more data is added, (all are around 61%). This coverage is much lower than for authentic data models (*auth*$_{1+}$ column), with coverage of more than 66% for all training sizes.

We put this discrepancy in performance down to the limits of the knowledge encoded by the NMT system used for back-translation. In particular, the sentences on the source side are the output of that system, and so (i) the vocabulary of these source-side sentences is always restricted; and (ii) these sentences will contain errors mediated by the initial NMT system. Given enough data, it will reach a steady point and not improve further. We observe this in Figure 2. We can thus conclude that an NMT system trained on synthetic-only data can learn very well the knowledge encoded by the original system used for back-translation, and can even exceed its quality.

It is worth mentioning that models trained on synthetic or on hybrid data outperform the authentic-only models in the lower-sized training data sets. This indicates that in low-resource scenarios it makes sense to exploit back-translation in order to achieve a better NMT system. However, with synthetic-only data, at a given point the performance of the NMT system plateaus, while in the case of hybrid data the quality starts degrading as the synthetic data overpowers the authentic. In our experimental set-up and data we reached this point at a synthetic-to-authentic ratio of 2:1. In the future we will conduct more experiments with different data, data sizes and language pairs, as well as network set-ups to see whether a true tipping point emerges.

[6] The only disagreement of BLEU with the rest of the evaluation metrics is the increment in the translation quality of the model trained using 3.5M synthetic sentences (compared to the model trained using 3M synthetic sentences). However this improvement is not statistically significant at level $p = 0.01$.

We believe this finding will have positive consequences especially for resource-poor scenarios. In particular, we hypothesise that using any existing MT system (or a combination of systems) to translate monolingual data in order to build an NMT system for the intended language direction with that data is likely to result in translation quality similar to that of the initial MT system.

7 Conclusion and Future Work

In this work we studied the performance of NMT German-to-English models when incrementally larger amounts of back-translated (or synthetic) data are used for training. We analysed hybrid NMT models built by adding back-translated data to an initial set of human-translated (or authentic) data, and showed that while translation performance tends to improve when larger amounts of synthetic data are added, performance appears to tail off when the balance is tipped too far in favour of the synthetic data; in our experiments we see a drop in performance of 1.2% for the 3.5M hybrid model compared to the 3M hybrid one. We plan to extend these experiments further in our future work, in order to figure out whether there exists a genuine tipping point, i.e. a ratio between the amount of synthetic and authentic data where the model achieves optimal performance, and beyond which the more synthetic data is added, the worse the NMT quality becomes.

We also built models using synthetic data alone. To our surprise, the performance is quite good; the synthetic-only baseline model achieved quality very close to that of the authentic-only engines. Astonishingly, the synthetic-only engine trained with 1M sentences performs better as scored by METEOR than the authentic-only engine trained on the same amount of data.

We believe our findings have important repercussions for resource-poor scenarios, especially where some prior engine – not necessarily an NMT system – exists for the reverse language direction, as this can be used to create arbitrarily large amounts of back-translated data for bootstrapping an NMT engine for the other language direction. We will investigate this further in ongoing work.

In other future work, we also want to explore the effect of adding artificial data to different language pairs and domains. We envisage the current research as the first contribution to an ongoing investigation of the true merits and limits of back-translation. It may well turn out that adding incrementally larger amounts of back-translated data is less harmful than we expect, but at least doing this from the ground up will hopefully result in a set of principles for NMT practitioners, rather than the rather haphazard state of affairs we see before us today.

Acknowledgements

This research has been supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

This work has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 713567.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, San Diego, CA, USA, 2015. 15pp.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, Ann Arbor, Michigan, 2005.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas, USA, 2016.

Ondrej Bojar and Ales Tamchyna. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011*, pages 330–336, Edinburgh, Scotland, 2011.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias

Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation (wmt16). In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 131–198, Berlin, Germany, 2016.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, 2015.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120, 2017.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1179`.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, page 176–181, Portland, Oregon, 2011.

Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. FBK's participation to the English-to-German News Translation Task of WMT 2017. In *Proceedings of the Second Conference on Machine Translation*, pages 271–275, Copenhagen, Denmark, 2017.

Tobias Domhan and Felix Hieber. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark, 2017.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015.

Sepp Hochreiter, Jürgen Schmidhuber, and Corso Elvezia. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, 2016. 8pp.

Alina Karakanta, Jon Dehdari, and Josef van Genabith. Neural machine translation for low-resource languages without parallel corpora. *Machine Translation*, 32, 2018. 23pp.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, 2017.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, 2004.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada, 2017.

Chi-kiu Lo, Boxing Chen, Colin Cherry, George Foster, Samuel Larkin, Darlene Stewart, and Roland Kuhn. NRC Machine Translation System for WMT 2017. In *Proceedings of the Sec-*

ond Conference on Machine Translation, pages 330–337, Copenhagen, Denmark, 2017.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, 2015.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

Jaehong Park, Byunggook Na, and Sungroh Yoon. Building a neural machine translation system using only synthetic parallel data. *arXiv preprint arXiv:1704.00253*, 2017.

Maja Popovic. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, 2015.

Spencer Rarrick, Chris Quirk, and Will Lewis. Mt detection in web-scraped parallel corpora. In *Proceedings of MT Summit XIII*, pages 422–429, Xiamen, China, 2011.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, 2016a.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, 2016b.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany, 2016c.

Dimitar Shterionov, Pat Nagle, Laura Casanellas, Riccardo Superbo, and Tony O'Dowd. Empirical evaluation of nmt and pbsmt quality for large-scale translation production. In *User track of the 20th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 74–79, Prague, Czech Republic, 2017.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, 2006.

Harold Somers. Round-trip translation: What is it good for? In *Proceedings of the Australasian Language Technology Workshop 2005 (ALTW 2005)*, pages 71–77, Sydney, Australia, 2005.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Montreal, Quebec, Canada, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Thirty-first Annual Conference on Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA., USA, 2017.

Andy Way. Traditional and emerging use-cases for machine translation. In *Proceedings of Translating and the Computer 35*, London, 2013. 12pp.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

Multi-Domain Neural Machine Translation

Sander Tars and **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
tarssander1@gmail.com, fishel@ut.ee

Abstract

We present an approach to neural machine translation (NMT) that supports multiple domains in a single model and allows switching between the domains when translating. The core idea is to treat text domains as distinct languages and use multilingual NMT methods to create multi-domain translation systems; we show that this approach results in significant translation quality gains over fine-tuning. We also explore whether the knowledge of pre-specified text domains is necessary; turns out that it is after all, but also that when it is not known quite high translation quality can be reached, and even higher than with known domains in some cases.

1 Introduction

Data-driven machine translation (MT) systems depends on the text domain of their training data. In a typical in-domain MT scenario the amount of parallel texts from a single domain is not enough to train a good translation system, even more so for neural machine translation (NMT; Bahdanau et al., 2014); thus models are commonly trained on a mixture of parallel texts from different domains and then later fine-tuned to in-domain texts (Luong and Manning, 2015).

In-domain fine-tuning has two main shortcomings: it depends on the availability of sufficient amounts of in-domain data in order to avoid overfitting and it results in degraded performance for all other domains. The latter means that for translating multiple domains one has to run an individual NMT system for each domain.

In this work we treat text domains as distinct languages: for example, instead of English-to-Estonian translation we see it as translating English news to Estonian news. We test two multilingual NMT approaches (Johnson et al., 2016; Östling and Tiedemann, 2017) in a bilingual multi-domain setting and show that both outperform single-domain fine-tuning on all the text domains in our experiments.

However, this only works when the text domain is known both when training and translating. In some cases the text domain of the input segment is unknown – for example, web MT systems have to cope with a variety of text domains. Also, some parallel texts do not have a single domain while they are either a mix of texts from different sources (like crawled corpora) or naturally constitute a highly heterogeneous mix of texts (like subtitles or Wikipedia articles).

We address these issues by replacing known domains with automatically derived ones. At training time we cluster parallel sentences and then applying the multi-domain approach to these clusters. When translating, the input segments are classified as belonging to one of these clusters and translated with this automatically derived information.

In the following we review related work in Section 2, then present our methodology of multi-domain NMT and sentence clustering in Section 3. After that, we describe our experiments in Sections 4 and 5 and discuss the results in Section 6. Section 7 concludes the paper.

2 Related Work

The baseline to which we compare our work is fine-tuning NMT systems to a single text domain

(Luong and Manning, 2015). There, the NMT system is first trained on a mix of parallel texts from different domains and then fine-tuned via continued training on just the in-domain texts. The method shows improved performance on in-domain test data but degrades performance on other domains.

In (Sennrich et al., 2016a) the NMT system is parametrized with one additional input feature (politeness), which is included as part of the input sequence, similarly to one of our two approaches (in our work – the domain tag approach). However, their goal is different from ours.

In (Kobus et al., 2017) additional word features are used for specifying the text domain together with the same approach as (Sennrich et al., 2016a). Although both methods overlap with the first part or our work (domain features and domain tags), they only test these methods on pre-specified domains, while we include automatic domain clustering and identification. Also, they use in-domain trained NMT systems as baselines even for small parallel corpora and do experiments with a different NMT architecture. Finally, their results show very modest improvements, while in our case the improvements are much greater.

Other approaches also define a mixture of domains, for example (Britz et al., 2017; Chen et al., 2016). However, both define custom NMT methods and also limit the experiments to the cases where the text domain is known.

3 Methodology

In the following we describe two different approaches to treating text domains as distinct languages and using multi-lingual methods, resulting in multi-domain NMT models. The first approach is inspired by Google's multilingual NMT (Johnson et al., 2016) and the second one by the cross-lingual language models (Östling and Tiedemann, 2017). Then we describe our methods of unsupervised domain segmentation used in our experiments in comparison with the pre-specified text domains.

3.1 Domain as a Tag

The first approach is based on (Johnson et al., 2016). Their method of multilingual translation is based on training the NMT model on data from multiple language pairs, while appending a token specifying the target language to the beginning of the source sequence. No changes to the NMT architecture are required with this approach. They show that the method improves NMT for all languages involved; as an additional benefit, there is no increase in the number of parameters, since all language pairs are included in the same model.

We adapt the language tag approach to text domains, appending the domain ID to each source sentence; thus, for instance, "How you doin' ?" from OpenSubtitles2016 (Lison and Tiedemann, 2016) becomes "__OpenSubs How you doin' ?".

The described method has two advantages. Firstly, it is independent of the NMT architecture, and scaling to more domains means simply adding data for these domains. We can assign a domain to each sentence pair of the training set sentence pair, or set the domain to "other" for sentences whose domain we cannot or do not want to identify.

Secondly, in a multilingual NMT model, all parameters are implicitly shared by all the language pairs being modeled. This forces the model to generalize across language boundaries during training. It is observed that when language pairs with little available data and language pairs with abundant data are mixed into a single model, translation quality on the low resource language pair is significantly improved.

We expect this to be even more useful for text domains. Traditional tuning to a low-resource domain, or for any specific domain for that matter, would result in a likely over-fitting to that domain. Our approach, where all parameters are shared, learns target domain representations without harming other domains' results while maintaining the ability to generalize also on in-domain translation, because little to no over-fitting will be caused. Furthermore, since domains are much more similar than languages, we expect the parameter sharing to have a stronger effect.

3.2 Domain as a Feature

The second approach is based on (Östling and Tiedemann, 2017) for continuous multilingual language models. The authors propose to use a single RNN model with language vectors that indicate what language is used. As a result each language gets its own embedding, thus ending up with a language model with a predictive distribution $p(x_t|x_{1...t-1}, l)$ which is a continuous function of the language vector l.

In our approach the same idea is implemented via word features of Nematus (Sennrich et al., 2017), with their learned embeddings replacing the language vector of (Östling and Tiedemann, 2017). For example, translating ”This is a sentence .” to the Estonian Wikipedia domain would mean an input of ”This|2wi is|2wi a|2wi sentence|2wi .|2wi”[1]

Having a single language model learn several languages helps similar languages improve each others representations (Östling and Tiedemann, 2017). Also, they point out that this greatly alleviates the problem of sparse data for smaller languages. We expect the same effect for text domains, especially since similarity between different domains of the same languages is higher than between different languages. Moreover, similarly to the domain tag approach, the usage of many domains in one model helps bypass the over-fitting problem of smaller domains.

3.3 Automatic domain tags

Here we define the domain of each of the source–target sentence pair automatically. We take two different approaches to achieve the annotation.

Supervised approach is done only in single domain setting. It involves assigning categories to roughly 10,000 Wikipedia articles, for which it could be done with high certainty. Assigning categories to more articles is problematic, because the categories assigned in Wikipedia can often be misleading in terms of content. Next we tag each sentence with the article category.

After tagging the sentences, we train a FastText (Bojanowski et al., 2016; Joulin et al., 2016) classification model with default settings and apply it to classify the rest of the sentences that were not classified based on the article categories. Test/dev set sentences are tagged using the same FastText model that is used to cluster training data.

Unupervised approach is applied to sentence-split data. In case of multi-domain data we still treat it as a single domain data of which we have no domain structure knowledge. In this approach, we train a model and calculate sentence vectors in an unsupervised manner using sent2vec (Pagliardini et al., 2017). After that, we apply KMeans clustering to identify the clusters in the set of calculated sentence vectors. Finally, we tag each sentence with the label that it was assigned by KMeans. To find the optimal number of clusters, we create several versions with different numbers of clusters.

To tag the test/dev set sentences, we train a FastText (Bojanowski et al., 2016) (Joulin et al., 2016) supervised classification model on the tagged training set. For each of the cluster versions and for each language pair, we train a separate FastText model. The additional benefit of this kind of clustering is that each new input sentence can be efficiently assigned its cluster. Also, because of more potentially homogenous train-set clusters, the new sentence is hypothetically assigned more appropriate domain than it would be assigned in case of the pre-defined domains.

The potential benefit of the unsupervised approach over supervised approach is that it does not assume any prior knowledge of the data and thus the domain structure does not rely on potentially faulty pre-defined domain structure. This in turn allows the multi-domain translation approach to be applied to any data without the knowledge of its domain structure.

4 Experiments with Known Domains

In the experiments we use mixed-domain parallel data consisting of Europarl (Koehn, 2005), OpenSubtitles2016 (Lison and Tiedemann, 2016), parallel data extracted from English-Estonian Wikipedia articles and some more mixed parallel corpora from the OPUS collection (Lison and Tiedemann, 2016). The size of the corpora is shown in Table 1. For each corpus we use a randomly chosen and held-out test set of 3000 parallel sentences.

Corpus	Sents	EN tok	ET tok
Opensubs	10.32	83.57	67.56
Europarl	0.644	17.18	12.82
Wiki	0.135	2.281	2.089
Other	7.972	169.9	143.5
Total	19.07	272.9	225.9

Table 1: Data sizes for the training data. Number of tokens (tok) is given pre-BPE. All of the numbers are given in millions

4.1 Mining Wikipedia for Translations

Wikipedia[2] itself is a big set of articles. The articles have two properties, which are extremely useful from our task point of view. Firstly, the arti-

[1] the ”|” is a special symbol in Nematus for delimiting input features.

[2] http://www.wikipedia.org/

cles have links to the articles of same topic, but in different languages, which makes it easier to find comparable data from which to extract parallel data. Secondly, each article has one or several categories attached to it. This means that hypothetically we can assign domain(s) to at least some of the articles based on these categories.

To extract meaningful text from the Wikipedia XML dumps, we used the WikiExtractor tool[3]. The data is extracted in a way that preserves article and paragraphs boundaries. The extraction is done separately for English and Estonian version.

After extracting text from the dumps, another custom-made solution is applied to detect parallel articles. The number of Wikipedia articles in English is well over 5 million whereas for Estonian it is just over 100 thousand. We keep all Estonian articles and only those English articles that have a parallel article in Estonian articles. This leaves us with roughly 70 thousand English articles.

The parallel articles form a comparable corpus. In case of this comparable corpora we know that the articles are parallel in terms of topics but not in sentences. To extract parallel sentences from parallel articles, we used the LEXACC (Ştefănescu et al., 2012) tool, which is a part of the ACCURAT toolkit (Pinnis et al., 2012; Skadiņa et al., 2012). Parallel sentence identification allows also to maintain the info of article origin, which means that direct domain assigning is possible. The identification process also assigns score to each sequence pair, which allows us to create parallel sets with different grade of purity. The optimal grade of purity produced 340 thousand parallel sentences. The size of Estonian Wikipedia in total is 2.8 million sentences. To the rest 2.5 million sentences back-translation is applied to extend the Wikipedia dataset for EN-ET direction; the back-translated sentences are also filtered based on attention weights (Rikters and Fishel, 2017) with a 50% threshold.

4.2 Technical Settings

We apply BPE segmentation (Sennrich et al., 2016b) in a joint learning scenario, learning from the input and the output, limiting the vocabulary to 65,000 entries. The acquired segmentation mostly corresponds to the linguistic intuition on frequent tokens (which are left intact) and medium-frequency tokens (which are split into compound parts or endings off stems); low-frequency tokens (also names, numeric tokens) are split into letters and letter pairs.

The NMT model we use is encoder-decoder with an attention mechanism (Bahdanau et al., 2014), implemented in Nematus (Sennrich et al., 2017). All settings (like embedding size, number of recurrent layers in encoder and decoder, etc.) are kept at their default values. Batch size in experiments is 50 sequences.

4.3 Results

For the **Baseline** experiment we first train a baseline model on all the datasets are used, and use it for translation. Then in the **Tuned** approach for each dataset separately we fine-tune the **Baseline** model to each corpus separately.

For the comparability of the results, the number of iterations during training (800,000) and input parameters are kept equal for **Baseline**, **Tag**, **Feat**. The tuning of **Baseline** is done for additional 60,000 iterations. One iteration means one batch seen during training.

Tables 2 and 3 show the BLEU scores (Papineni et al., 2002) and the p-values of the statistical significance of their difference for Baseline, fine-tuned baseline, domain tags, and domain feature approaches.

As we can see from the results, both of the additional domain info models perform really well. The domain tag (**Tag**) model outperforms both of its baseline (**Baseline**) and tuned (**Tuned**) counterpart in ET–EN direction. It even goes as far as exceeding the **Tuned** approach by more than 1.0 BLEU in all domains. The same holds, but even more strongly, for the version where we add the domain embedding as an input feature for each word (**Feat**).

For EN–ET direction the results do not show such strong improvements. In this direction both **Tag** and **Feat** outperform **Baseline** for all domains. However, the scoring is quite close to the **Tuned** approach with the results between **Tag** and **Feat** also being closer than in ET–EN case. All in all, the fact that the domain tagging results are essentially on-par with Tuned approach, means it is superior to the **Tuned** approach in practice because of the fact that it requires only one model rather than three.

Table 4 shows an example of the **ET–EN** translations highlighting some improvements. Since the

[3] https://github.com/attardi/wikiextractor

Corp	**Baseline**	**Tuned**	**Tag**	**Feat**
Eu	33.0±0.3	35.4±0.3	36.2±0.3	37.3±0.3
Op	27.9±0.6	28.1±0.6	30.5±0.6	30.3±0.6
Wi	15.3±0.4	15.4±0.4	16.9±0.4	17.7±0.4

Corp	**Baseline**	**Tuned**	**Tag**	**Feat**
Eu	0.0001 / 0.0001	0.009 / 0.0001	- / 0.0001	0.0001 / -
Op	0.0001 / 0.0001	0.0001 / 0.0001	- / 0.1	0.1 / -
Wi	0.0001 / 0.0001	0.0001 / 0.0001	- / 0.001	0.001 / -

Table 2: BLEU scores and p-values for Estonian-English direction. **Baseline** model is trained without domain tags. **Tuned** is achieved by tuning these models with the specific corpus. **Tag** is trained with data that has domain tag prepended to each source sentence. **Feat** is trained with data that has domain embedding added as a feature to each source sequence word. p-values are given for significance against **Tag** and **Feat** respectively, separated with **/**.

Corp	**Baseline**	**Tuned**	**Tag**	**Feat**
Eu	22.5±0.3	25.3±0.3	25.4±0.3	24.9±0.3
Op	24.2±0.6	24.5±0.6	24.8±0.6	25.3±0.6
Wi	11.8±0.4	12.1±0.4	12.5±0.3	12.8±0.4

Corp	**Baseline**	**Tuned**	**Tag**	**Feat**
Eu	0.0001 / 0.0001	0.3 / 0.04	- / 0.04	0.04 / -
Op	0.01 / 0.001	0.09 / 0.03	- / 0.06	0.06 / -
Wi	0.01 / 0.001	0.06 / 0.03	- / 0.14	0.14 / -

Table 3: BLEU scores and p-values for English-Estonian direction. **Baseline** model is trained without domain tags. **Tuned** is achieved by tuning these models with the specific corpus. **Tag** is trained with data that has domain tag prepended to each source sentence. **Feat** is trained with data that has domain embedding added as a feature to each source sequence word. p-values are given for significance against **Tag** and **Feat** respectively, separated with **/**.

quality of **Tuned** is close to **Tag** and **Feat**, we omit it from the comparison since the differences would be highly circumstantial and would not hold much information in small scale.

Src (ET)	*vastuseid saab muidugi olla ainult üks : lõpetada kohe igasugused läbirääkimised Türgiga .*
Ref (EN)	*there is , of course , only one possible response : to immediately cease all negotiations with Turkey .*
Base (EN)	*only one can only be one : stop any negotiations with Turkey immediately .*
Tag (EN)	*the answer , of course , can only be one : stop all the negotiations with Turkey immediately .*
Feat (EN)	*there is , of course , only one answer : to put an end to all negotiations with Turkey immediately .*

Table 4: An example of Europarl corpus translations from Estonian to English using the Baseline, **Tag** and **Feat** methods.

5 Experiments with Automatic Domains

Since the results on the full parallel data show that both of multi-domain approaches are on-par, or superior to the single-domain baseline, we apply the methods in a setting where we do not assume beforehand knowledge of the origin domain of source sentences. Here we take the domain tagging approach: even though domain features show better results, domain tags are more generic and compatible with any NMT architecture.

We experiment with two data settings. In the first one, we have a single heterogeneous text domain. We explore both supervised and unsupervised tagging of single text domain based on sentence vectors.

In the second one, we have texts from several domains but we ignore the pre-specified text domains and replace them with automatic clustering based on sentence embeddings.

5.1 Automatic single-domain tagging

To choose the best setting for unsupervised approach, we do a small sweep for input data versions. We check for best number of clusters by training a model for each number of clusters. The input data for this is the whole Wikipedia corpus. The models are trained for 12 hours, which should be sufficient to make them diverge enough to choose the best number of clusters. We also train a regular model without data clustering for reference.

It is important to note that for this experiment a different test set was used than in the full data experiments. Thus the scores in 5 are not comparable to scores presented earlier.

The initial sweep indicates that the best option for the unsupervised classification is 12 clusters. Also, the 12 hours – 100,000 iterations are already showing the effect that domain tagging has over the regular reference approach, making other clusterings also a viable choice.

Wikipedia Translation Results

In the final experiment, three models were trained:

- Supervised 5-domain source tag model
- Unsupervised 5-domain source tag model
- Unsupervised 12-domain source tag model
- Regular not domain-tagged model

Unsupervised 5-domain model was included to compare the performance of supervised and unsupervised approach with the same amount of domains, giving an indication of the ”goodness” of these cluster assignments. The Unsupervised 12-domain model was included to compare the performance of best unsupervised clustering and the intuitively optimal supervised clustering. Supervised 12-domain model is not presented because we were not able produce such reasonable structure from ET Wikipedia. The results are presented in 6. The models were trained for 48 hours.

As we see in Table 6, the Supervised approach (**Super**) with five clusters slightly outperforms Unsupervised 5-cluster approach (**Usup5**). The best option for Unsupervised clustering (**Usup12**) performs as well as the Supervised approach. The results show that Unsupervised approach is comparable in performance to the Supervised approach, which means that at least in this setting both of the approaches are viable. Even more so, when obtaining labelled data for supervised clustering can often require a lot of additional effort, the unsupervised approach is not chained by the (lack) of pre-existing knowledge about the data.

Most important is the fact that both of the unsupervised cluster versions outperform the regular reference (**Ref**) version where sentence cluster tags were not used. This shows that the unsupervised clustering approach can potentially be used in settings that previously were viewed upon as single clusters. For example OpenSubtitles corpus could be clustered further, to improve the translations.

5.2 Unsupervised multi-domain tagging

Hinging on the fact that domain tagging approach outperformed the traditional tuning approach and on the results that unsupervised Wikipedia dataset clustering produced, the ”traditional” approach of text domains should be given another look. One possible action is to cluster or sub-cluster the existing parallel data to restructure it from the domain point of view.

In addition to the results produced on wikipedia dataset, the hypothesis on why this would work, is that large text domains are probably not very homogenous. Also, different domains have probably pretty big overlap of similar sentences. This would mean that the usual approach of domain tuning or domain tagging does not achieve its true potential, because predefined domains are *de facto* several domains and the same domains are actually present in other predefined domains also.

To check for this property and its potential benefit for NMT, we cluster existing parallel sentences to n clusters in the previously described unsupervised manner, train NMT models with domain tagged sentences, and finally, cluster test set sentences in a supervised manner with a supervised clustering model that is trained on the data that was obtained from unsupervised clustering.

The training is done using Nematus with the same settings as in the initial experiment with domain tags. Firstly, we do the sweep of clusters by training 4, 8, 16, and 32 cluster versions for both EN–ET and ET–EN direction. After that we choose the version that has achieved the best BLEU scores on the dev sets for both of the directions and train it for the same time as in the initial domain tag experiment with full data.

Results of unsupervised multi-domain tagging

To evaluate the model performance, we train supervised FastText classification models on the

NClust	**C4**	**C5**	**C6**	**C8**	**C12**	**Ref**
BLEU	19.7	19.5	19.6	19.5	20.0	17.9

Table 5: BLEU scores for Unsupervised Wikipedia parameter setting.

NClust	**Usup12**	**Usup5**	**Super**	**Ref**
BLEU	26.0±0.4	25.2±0.4	25.8±0.4	23.6±0.4
pU12	-	0.01	0.1	0.0001
pU5	0.01	-	0.03	0.0001
pSup	0.1	0.03	-	0.0001
pRef	0.0001	0.0001	0.0001	-

Table 6: BLEU scores and p-values for test on Wikipedia-only data to compare the effect of Unsupervised clustering (**Usup12, Usup5**), supervised clustering (**Super**) and no-clustering approach (**Ref**). The p-values are shown in respect to the version where the value is **-**.

tagged training data. We apply these models on the test/dev sets to classify the sentences. This means that each of the sets – Opensubs, Europarl, and Wiki – gets actually tags from several clusters, depending on which cluster the FastText model assigns to each of the sentences. This means that for each source test set we create four different versions, each for cluster numbers 4, 8, 16, and 32.

The initial parameter sweep shows that the best option is 16 clusters for both EN–ET 7 and ET–EN 8 directions across all test sets. Hence the final models were both trained with 16 clusters.

Corp	**C4**	**C8**	**C16**	**C32**
Eu	4.13	3.19	**5.94**	4.17
Op	9.41	9.36	**10.80**	10.62
Wi	1.09	0.94	**1.31**	0.81

Table 7: BLEU scores for English-Estonian direction sweep. The model is trained on parallel data that is tagged in unsupervised manner using sent2vec + Kmeans clustering. The dev sets are clustered based on this tagged data using FastText. The best scores for each corpus are presented in **bold**.

Corp	**C4**	**C8**	**C16**	**C32**
Eu	20.48	19.88	**20.82**	18.43
Op	20.05	19.54	**20.17**	20.01
Wi	4.61	4.38	**5.50**	4.32

Table 8: Test set BLEU scores for Estonian-English direction sweep. The model is trained on parallel data that is tagged in unsupervised manner using sent2vec + Kmeans clustering. The dev sets are clustered based on this tagged data using FastText. The best scores for each corpus are presented in **bold**.

In table 9 is shown the OpenSubs test sets cluster structure. The test sets are tagged using FastText models trained on tagged train set. We can see that different train set clusters produce different granularity in test sets also. For **C4, C8** the OpenSubs structure is similar, same holds for other test sets. **C16** vs **C8** however shows a significant difference in test set clustering. Here we see that OpenSubs, which based on content is probably not homogeneous domain, is separated quite granularly in **C16**, producing 3–4 main sub-domains. In **C32** the test set is clustered even further, but based on sweep scores, it could be said that the achieved clustering is already too granular.

Corp	**N1**	**N2**	**N3**	**N4**	**N5**	**N6**
C4	2921	29	-	-	-	-
C8	2907	43	-	-	-	-
C16	1331	1015	398	181	18	7
C32	1137	828	356	293	241	71

Table 9: Cluster structure of FastText tagged English OpenSubs test sets. The test sets are clustered based on tagged train data. The clusters are numbered left to right based on size. Here only top 6 clusters are shown. For C32 $N7 = 11$, $N8 = 6$, $N9 = 3$, $N10 = 2$, $N11 = 2$. Test set structures for Estonian sets are similar.

Considering that our OpenSubs cluster is 10 million sentence pairs in size, we can say that **C16** finds 5 significant sub-domains and one less significant sub-domain inside it. This shows that, at least from sentence vectorizing point of view, there exists more than one domain inside OpenSubs, and similarly in other domains.

When looking at the number of clusters present in Table 9, one could notice that the clusters present is less than number of clusters defined. It should be kept in mind that we have 3 main text sources in training set and fourth mixed-corpus which could be divided into 5-6 parts, so 8-9 text domains in total. Also, some sentences are quite

Corp	N1	N2	N3	N4	N5	N6	N7	N8
Train	4859672	4444177	3704753	2767889	1407228	822225	711526	134301
Corp	**N9**	**N10**	**N11**	**N12**	**N13**	**N14**	**N15**	**N16**
Train	114778	40260	22004	18165	10298	9585	2492	646

Table 10: Cluster structure of KMeans tagged English train set for **C16**. The clusters are numbered left to right based on size. Train set structure for Estonian is similar.

distinct from the others based on full train set cluster structure as we can see from the train set structure of **C16** in Table 10. The clustering and its structure is probably interesting aspect to look into in future work.

The final results, where the 16 cluster models were trained for the same amount of iterations as in the initial full data experiments, are presented in 11 and 12 for EN–ET and ET–EN language pairs respectively.

The results show that the unsupervised clustering approach performs similarly with the pre-defined tag version. The results are evidence that the unsupervised tagging approach can serve as a viable alternative to the traditional pre-defined domain approach. Our hypothesis is that this is caused by the pre-defined domains being less homogenous in content than the unsupervised clustered ”domains”. However, this hypothesis should be investigated further to assert its existence and magnitude. Also, since the clustering approach is pretty much applied out-of-the-box, then improved clustering could provide considerable improvements.

All-in-all, taking into consideration the fact that unsupervised approach allows new sentences to be translated with potentially more appropriate domain assigned to them, the unsupervised tagging approach can be seriously considered as the go-to approach for multi-domain translation models.

6 Discussion

The results from the experiments - EN–ET and ET–EN direction parallel translation, Wikipedia data translation, and unsupervised sentence tagging - show that both of the two chosen multi-domain approaches outperform regular approach of uniform translation and domain-tuning.

This indicates the hypothesis that the parameter sharing effect discussed in Google's zero-shot article would benefit domain translation holds. The translation scores even outperform domain-tuning approach, which could be explained by the same parameter sharing. In tuning we tune the model to translate sentences characteristic to the model we are tuning to. This means that domain-characteristic sentences get translated really well. On the other hand, the not-so-characteristic sentences get neglected. The parameter sharing effect of the multi-domain approach helps negate the negative effect by the support of other domains while still learning to more effectively represent each domain by the additional domain info.

Furthermore, the results indicate that adding domains as an input feature can have even stronger effect on the translation scores. This shows that concatenating the domain feature embedding with word embedding at each timestep - basically remembering the source domain equally throughout the sequence improves model performance. This could be explained by the fact that in tag prepending case, the neural net may ”forget” for longer sequences what the input tag was, making the effect of it weaker.

The results also show that for highly quality dependent settings the domain feature concatenation with word embedding is the more suitable option. However, the differences in scores are not drastically different from the domain tag prepending. This means that for the sake of data simplicity, model simplicity and efficiency the tag prepending approach could prove more reasonable of the two for in-production settings.

Finally, the performance of unsupervised domain tagged model indicates that there is grounds to substitute the pre-defined domain approach with automatically assigned domain approach. The unsupervised certainly serves as an improvement in less homogenous single domain settings, where the effect of the detection of underlying ”domains” was shown on the example of Wikipedia.

No less important are the facts that the unsupervised tagging approach ensures better domain assignment to each new sentence and can efficiently incorporate new data from various small domains to fortify each of the learned ”domain” (clusters).

It has to be taken into account that the unsuper-

Corp	Baseline	Tuned	Tag	Unsup
Eu	22.5±0.3	25.3±0.3	25.4±0.3	24.5±0.3
Op	24.2±0.6	24.5±0.6	24.8±0.6	24.6±0.6
Wi	11.8±0.4	12.1±0.4	12.5±0.3	11.1±0.4

Corp	Baseline	Tuned	Tag	Unsup
Eu	0.0001 / 0.0001	0.3 / 0.03	- / 0.004	0.004 / -
Op	0.01 / 0.03	0.09 / 0.4	- / 0.2	0.2 / -
Wi	0.01 / 0.01	0.06 / 0.005	- / 0.0001	0.0001 / -

Table 11: Test set BLEU scores and p-values for English-Estonian direction. **Baseline** model is trained without domain tags. **Tuned** is achieved by tuning these models with the specific corpus. **Tag** is trained with data that has domain tag prepended to each source sentence. **Unsup** is trained with data that has domain tags assigned to each sentence in an previously described unsupervised manner. p-values are given for significance against **Tag** and **Unsup** respectively, separated with **/**.

Corp	Baseline	Tuned	Tag	Unsup
Eu	33.0±0.3	35.4±0.3	36.2±0.3	36.0±0.3
Op	27.9±0.6	28.1±0.6	30.5±0.6	30.2±0.6
Wi	15.3±0.4	15.4±0.4	16.9±0.4	16.0±0.4

Corp	Baseline	Tuned	Tag	Unsup
Eu	0.0001 / 0.0001	0.009 / 0.01	- / 0.3	0.3 / -
Op	0.0001 / 0.0001	0.0001 / 0.0001	- / 0.1	0.1 / -
Wi	0.0001 / 0.004	0.0001 / 0.009	- / 0.01	0.01 / -

Table 12: Test set BLEU scores and p-values for Estonian-English direction. **Baseline** model is trained without domain tags. **Tuned** is achieved by tuning these models with the specific corpus. **Tag** is trained with data that has domain tag prepended to each source sentence. **Unsup** is trained with data that has domain tags assigned to each sentence in an previously described unsupervised manner. p-values are given for significance against **Tag** and **Unsup** respectively, separated with **/**.

vised clustering performed in these experiments is applied basically in out-of-the-box manner, which means that domain assignments can be improved and thus the translation scores should also improve.

7 Conclusions

In this article we tested two approaches to improve multi-domain neural translation. One approach involves prepending domain tags to source sentences, the other adding domain embeddings as an input feature to each source sentence word. We showed that both ways of adding domain information to source sentences in bilingual neural translation improves translation scores considerably compared to both regular baseline translation and fine-tuning. These improvements in source sentence tagging case can be obtained with mere data manipulation.

We also showed that the domain tagging approach can be successfully coupled with unsupervised sentence clustering to add a ”domain dimension” to a previously single-domain corpus. This approach produces better results as opposed to using the corpus as a single domain. The results indicate that unsupervised or semi-supervised training data clustering can be effectively used to improve neural machine translation.

Finally, to bring the two experiments together, we apply unsupervised domain tagging to full parallel data and show that it can serve as a viable alternative to the pre-defined domain approach.

For future work the clustering in fully unsupervised tagging approach should be improved to see if this gives a visible improvement in translation scores.

Secondly, a more comprehensive sweep on number of clusters should be done. It would be interesting to see for how many clusters the effect still persists. This however would need more extensive computational resources and should probably be done with some model dataset.

The differences of the two approaches - source sentence tagging and adding domain info as an input feature - deserve to be looked into more deeply. More precisely, the result profiles of the two in different cases of domain granularity.

Finally, in this work domains are still treated as nominal values; it would be interesting to explore the estimation of domain embeddings at translation time as continuous values.

Acknowledgements

This work was supported by the Estonian Research Council grant no. 1226. Also we are grateful for the anonymous reviews and their help in improving this article.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Britz, Denny, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of WMT*, pages 118–126, Copenhagen, Denmark.

Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *CoRR*, abs/1607.01628.

Johnson, Melvin, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Kobus, Catherine, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of RANLP*, pages 372–378, Varna, Bulgaria.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, volume 5, pages 79–86, Phuket , Thailand.

Lison, Pierre and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of LREC*, pages 923–929, Portorož, Slovenia.

Luong, Minh-Thang and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of IWSLT*, pages 76–79, Da Nang, Vietnam.

Östling, Robert and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of EACL*, pages 644–649, Valencia, Spain.

Pagliardini, Matteo, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *CoRR*, abs/1703.02507.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Pinnis, Mārcis, Radu Ion, Dan Ştefănescu, Fangzhong Su, Inguna Skadiņa, Andrejs Vasiļjevs, and Bogdan Babych. 2012. Accurat toolkit for multi-level alignment and information extraction from comparable corpora. In *Proceedings of ACL*, pages 91–96, Jeju Island, Korea.

Rikters, Matīss and Mark Fishel. 2017. Confidence through attention. In *Proceedings of MT Summit*, pages 299–311, Nagoya, Japan.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of NAACL*, pages 35–40, San Diego, California.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725, Berlin, Germany.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of EACL*, pages 65–68, Valencia, Spain.

Skadiņa, Inguna, Ahmet Aker, Nikos Mastropavlos, Fangzhong Su, Dan TufiÈ™, Mateja Verlic, Andrejs Vasiļjevs, Bogdan Babych, Paul Clough, Robert Gaizauskas, Nikos Glaros, Monica Lestari Paramita, and Mārcis Pinnis. 2012. Collecting and using comparable corpora for statistical machine translation. In *Proceedings of LREC*, pages 438–445, Istanbul, Turkey.

Ştefănescu, Dan, Radu Ion, and Sabine Hunsicker. 2012. Hybrid parallel sentence mining from comparable corpora. In *Proceedings of EACL*, pages 137–144, Trento, Italy.

A Comparison of Different Punctuation Prediction Approaches in a Translation Context

Vincent Vandeghinste
CCL – KU Leuven
vincent@ccl.kuleuven.be

Lyan Verwimp
ESAT-PSI – KU Leuven
lyan.verwimp@esat.kuleuven.be

Joris Pelemans
Apple Inc.
jpelemans@apple.com

Patrick Wambacq
ESAT-PSI – KU Leuven
patrick.wambacq@esat.kuleuven.be

Abstract

We test a series of techniques to predict punctuation and its effect on machine translation (MT) quality. Several techniques for punctuation prediction are compared: language modeling techniques, such as n-grams and long short-term memories (LSTM), sequence labeling LSTMs (unidirectional and bidirectional), and monolingual phrase-based, hierarchical and neural MT. For actual translation, phrase-based, hierarchical and neural MT are investigated. We observe that for punctuation prediction, phrase-based statistical MT and neural MT reach similar results, and are best used as a preprocessing step which is followed by neural MT to perform the actual translation. Implicit punctuation insertion by a dedicated neural MT system, trained on unpunctuated source and punctuated target, yields similar results.

1 Introduction

In speech translation, the first step often consists of automatic speech recognition (ASR). Most ASR systems output an unsegmented stream of words, apart from some form of acoustic segmentation which splits a transcript into so-called *utterances*. Translating this stream of words, using off-the-shelf MT, results in a lower translation quality compared to translating punctuated input, as MT systems are usually trained on properly punctuated and segmented source and target text. End-to-end speech translation systems, that do not suffer from this problem, have recently achieved high-quality results too (Weiss et al., 2017), but these models require infrastructure (in terms of GPUs and training time) that is not available to everyone.

We compare several techniques and approaches for punctuation prediction in a translation context, starting from an input that already contains the correct sentence boundaries. All techniques and approaches are trained on the same dataset, allowing us to fully attribute different results to the specific techniques and approaches used. Thus, the main contribution of this paper is not introducing new methods for punctuation prediction, but a thorough comparison of methods previously used, since extensive comparisons are often lacking in related work. We compare three families of approaches for punctuation prediction: (1) language modeling, (2) sequence modeling, and (3) monolingual MT.

These approaches are combined in three different architectures resulting in translated and punctuated output: (1) *Preprocessing* adds punctuation before translating with a normal MT system, trained on punctuated source and punctuated target data; (2) *Implicit insertion* adds punctuation during MT, which is trained on unpunctuated source and punctuated target data; and (3) *Postprocessing* adds punctuation after MT, which is trained on unpunctuated source and unpunctuated target data. Figure 1 shows these different strategies, together with the *baseline* strategy, in which the unpunctuated data is translated by a regular MT system trained on punctuated source and target data.

2 Related work

In this section we discuss work that explicitly tries to predict punctuation marks like we do. We do not consider sentence boundary prediction.

Punctuation prediction is first described in

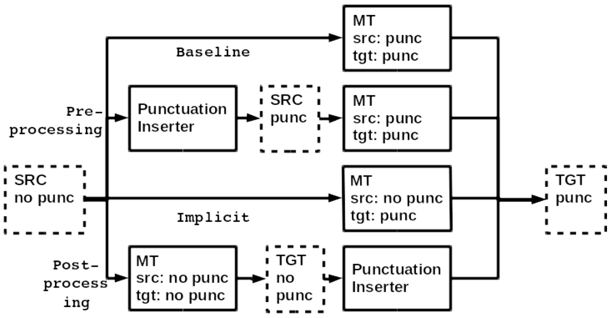


Figure 1: The different punctuation prediction strategies in a translation context.

(Beeferman et al., 1998), who use a lexical hidden Markov model to predict comma insertion in ASR output. Several other models have also been investigated, such as a decision tree classifiers (Kim and Woodland, 2001; Zhang et al., 2002), finite state models and multi-layer perceptrons (Christensen et al., 2001), a maximum entropy model (Huang and Zweig, 2002) and conditional random fields (Lu and Ng, 2010; Ueffing et al., 2013).

Gravano et al. (2009) use a purely text-based n-gram language model but do not compare with previously published methods. Several researchers use recurrent neural networks (RNNs) to tackle the problem as a sequence labeling task. Tilk and Alumäe (2015; 2016) use a two-stage LSTM (Hochreiter and Schmidhuber, 1997) to predict punctuation based on textual and prosodic features. Moró and Szaszák (2017) only use prosodic information to train a bidirectional LSTM while Gale and Parthasarathy (2017) compare several character-level convolutional and LSTM architectures, of which a simple LSTM with delay performs the best, although not consistently better than word-level bidirectional models. Pahuja et al. (2017) train a bidirectional RNN to jointly predict the correlated tasks of punctuation and capitalization.

As far as we know, only Tilk and Alumäe (2016) directly compare unidirectional and bidirectional word-level LSTMs for sequence labeling: even though their unidirectional model is smaller than their bidirectional model[1], the bidirectional one does not consistently outperform the unidirectional one. As we will see in section 4.1, we observe a similar trend.

In the context of MT, Matusov et al. (2006) and Peitz et al. (2011) present the three strategies for punctuation prediction we also use (as shown in figure 1). Lee and Roukos (2006) use a preprocessing approach, and Hassan et al. (2007) present a postprocessing apprach. Peitz et al. (2011) combine the outputs of the different strategies and find that "the translation-based punctuation prediction outperformed the LM based approach as well as implicit method in terms of BLEU and TER on the IWSLT 2011 SLT task". Combining outputs from different approaches through system combination yields even better results (Matusov et al., 2006b).

If we examine the comparisons with previously published methods in related work, we see that some do no compare their approach at all (Beeferman et al., 1998; Huang and Zweig, 2002; Hassan et al., 2007; Moró and Szaszák, 2017), others compare with either n-gram LMs (Kim and Woodland, 2001; Zhang et al., 2002; Lu and Ng, 2010; Peitz et al., 2011; Ueffing et al., 2013; Tilk and Alumäe, 2015; Tilk and Alumäe, 2016), CRF (Gale and Parthasarathy, 2017) or CRF and LSTM sequence labeling (Pahuja et al., 2017). We are not aware of a systematic comparison of MT approaches, n-gram LMs, LSTM LMs and LSTM sequence labeling. Especially a direct comparison of two of the most promising approaches, LSTM sequence labeling and monolingual MT, is lacking.

3 Methodology

We test several methods, keeping the data for training, tuning, and testing constant. Section 3.1 describes the data, section 3.2 discusses the models for punctuation prediction and section 3.3 the bilingual translation models. Finally section 3.4 explains how the quality of the punctuation prediction and translation is measured.

3.1 Data

As training data, we use the Dutch (source) and English (target) components of the Europarl corpus, version 7 (Koehn, 2005). The training data contains 55M words or 2M sentences (per language). As development set and test set we use the data of Vandeghinste et al. (2013). The development set consists of 574 sentences with one reference translation, randomly selected from actual translations made by a language service provider. As test set, we use 500 sentences with three reference translations, made by three different transla-

[1] A hidden size of 100 (Tilk and Alumäe, 2015) vs. 256 (Tilk and Alumäe, 2016), while it is not clear whether both the forward and the backward have 256 units or whether each of them have 128.

tors.[2]

All the data are tokenized, truecased, and depending on the experimental condition, cleaned with the Moses toolkit (Koehn et al., 2007). We compare the full dataset with a dataset in which all sentences longer than 80 words have been removed.[3]

We predict the following punctuation symbols: dot (`.`), comma (`,`), question mark (`?`), exclamation mark (`!`), colon (`:`), semicolon (`;`), opening and closing brackets (`()`), slash (`/`) and dash (`-`). Note that our punctuation set is much larger than most previous work that we are aware of: for example Gale and Parthasarathy (2017) and Pahuja et al. (2017) only focus on predicting periods, commas and question marks.

3.2 Punctuation prediction

We apply punctuation prediction in the preprocessing as well as in the postprocessing punctuation strategy, i.e. in Dutch and in English.

3.2.1 Punctuation prediction using language modeling

We train two types of LMs: n-gram models and LSTMs. The models for preprocessing are trained on Dutch and those for postprocessing on English. The n-gram models are 4-gram LMs (5-grams did not improve the performance) with interpolated modified Kneser-Ney smoothing (Chen and Goodman, 1999), trained with the SRILM toolkit (Stolcke, 2002). We compare the results for using the left context only (forward *fw*) with those for using both the left and the right context (forward + backward *fw+bw*), where both the preceding 3 words and the following 3 words can be used.

The LSTM LMs are trained with TensorFlow (Abadi et al., 2015) and consist of 1 layer of 512 cells, initialized randomly with a uniform distribution between -0.05 and 0.05. They are optimized with Adagrad (Duchi et al., 2011) with a learning rate of 0.1, early stopping is applied if the validation perplexity has not improved 3 times. Otherwise, the maximum number of epochs is 39. We train on batches of size 20 and unroll the network for maximum 35 time steps during backpropagation through time. With respect to regularization, the norms of the gradients are clipped at 5 and we apply 50% dropout (Srivastava et al., 2014) during training. We use sampled softmax (Jean et al., 2014) to speed up training. Due to a lack of resources, we did not apply an exhaustive hyperparameter optimization, but started from settings that have proven to work well for similar datasets.[4]

For punctuation prediction with LMs, we proceed as follows: we train the LMs on punctuated data and test on unpunctuated data. Given a non-punctuated input sentence, we determine the most probable token after every word. If a punctuation symbol is predicted, it is inserted at the current position in the input sentence and the updated sentence is used during the rest of the prediction. We continue the prediction until the end of the sentence is reached, including the position after the last token.

The full vocabulary of the training set consists of approximately 280k words for Dutch and 130k words for English (Dutch has much more compounding than English). Since models with that vocabulary size do not fit on our GPUs and since the large vocabulary also considerably slows down training of the LSTMs, we limit the vocabulary size to 50k. For fair comparison, we report results for n-grams models with the same vocabulary, but also for n-gram models with the full vocabulary in order to investigate the effect of the vocabulary size on the performance. All words not in the vocabulary are mapped to an *unknown-word*-class.

3.2.2 Punctuation prediction using sequence labeling

Besides LSTM LMs, we investigate LSTM sequence labeling ('LSTM seq'): we train an LSTM that takes as input a word and the previous state and predicts in the output whether the word is followed by a punctuation symbol or not ($\langle$*nopunct*$\rangle$-class). There are several advantages to this approach compared to language modeling: firstly, we train the LSTM on unpunctuated text and test it on unpunctuated text, so there is no mismatch in training and test conditions. Secondly, the models are directly optimized for punctuation prediction and they are easier to train since we do not have the large output weight matrix of an LM and we only

[2] These test sets are freely available upon request.

[3] For the development and test set, cleaning does not make any difference, as they are hand-made and are clean to begin with.

[4] We do not use bidirectional LSTM LMs for this task, since during training, the backward LSTM will have seen punctuation symbols following the current token and the model will learn to make use of those symbols. However, for applications such as speech translation, the input for the punctuation prediction model will have no punctuation at all, and hence the model that has learned to make use of subsequent symbols will not be optimal.

have to compute the softmax function over a small number of output classes. Finally, we can train bidirectional LSTMs without causing a mismatch between training input and testing input (see footnote 4). A disadvantage of these models is that the input is not punctuated, and hence the model cannot exploit punctuation in other parts of the sentence that is previously predicted. Note that, as opposed to Tilk and Alumäe (2016), we do not insert end-of-sentence symbols for the LSTM sequence labeling, because this would be an (unfair) advantage for bidirectional models – the probability of seeing an end-of-sentence punctuation mark right before an end-of-sentence symbol is naturally very high.

The hyperparameters of these models are the same as for the LSTM LMs, except that we use a full softmax in the output layer since we do not have to deal with a large vocabulary anymore. The bidirectional LSTMs consist of one forward LSTM of 256 cells and one backward LSTM of 256 cells, in total giving the same amount of LSTM cells and parameters as for the unidirectional LSTM (512).

3.2.3 Punctuation prediction using machine translation

We can model the punctuation prediction as an MT problem, treating the non-punctuated version of our text as source language, and the punctuated version as target language: we build such monolingual MT systems for Dutch (preprocessing) and English (postprocessing).

The phrase-based statistical MT (*PBSMT*) condition uses the Moses decoder (Koehn et al., 2007) in its phrase-based mode, with a 5-gram LM, and *grow-diag-final-and* as phrase alignment criterion. For other parameters we use the default settings. The data is word-aligned using GIZA++ (Och and Ney, 2003). We do not allow reordering, setting the *distortion-limit* to 0. The *PBSMT clean* condition is equal to PBSMT, but removing all sentences longer than 80 words from the training data.

The *Hiero* condition uses Moses in hierarchical mode (Chiang, 2007), with a glue grammar and a maximum phrase length of 5. The other parameters are the same as for the PBSMT condition. All Moses systems are tuned using Minimum Error Rate Training (Och, 2003), maximizing on BLEU.

For the Neural MT (NMT) models, we use the OpenNMT framework (Klein et al., 2017), trained with the default settings, i.e. 500 LSTM cells, *seq2seq* model type, a vocabulary of 50k for both source and target language, a general global attention model (Luong et al., 2015), 13 epochs, a batch size of 64, and optimization through stochastic gradient descent (SGD). The initial learning rate is 1, except for the English model trained with SGD: since the training got stuck in a local minimum, we use 0.9 instead. The learning rate is decreased with a decay factor of 0.7, a beam size of 5, and replacements of *unknowns*, based on the highest attention weight.[5] We also try a variant with optimization Adam (Kingma and Ba, 2014) and a learning rate of 0.0002.

Variants of the systems trained with byte pair encoding have not been included in this study as initial tests only showed worse results than without byte pair encoding.

3.3 Translation Methods

We use the same MT systems as described in section 3.2.3, but now trained on the bilingual version of Europarl. Different from section 3.2.3 is that we now do allow phrase reordering for the phrase-based model, setting the *distortion limit* to 6.

3.4 Evaluation

We measure the quality of *punctuation prediction* with precision, recall and F1-score. The precision over all punctuation symbols is calculated as follows:

$$precision_{all} = \sum_{i \in P} \frac{TP_i}{TP_i + FP_i} \qquad (1)$$

with P the class of all punctuation symbols, TP_i the number of true positives for a certain punctuation symbol and FP_i the number of false positives. Recall is calculated analogously. If a certain punctuation symbol has been predicted but the target is another punctuation symbol, we count this as a false negative.

Additionally, we use three common MT evaluation metrics, i.e. BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Denkowski and Lavie, 2014) with synonyms, comparing the test set with predicted punctuation with the reference text (original text including punctuation). These metrics give us information on the quality of the entire output (and not only the punctuation prediction), which can be an

[5] Replacing the *unknowns* by their most probable aligned source language word.

issue in MT models that allow reordering, such as Hiero and NMT.

We measure the *translation quality* with the same three MT evaluation metrics. Note that, as described in section 3.1, we use an evaluation set with three references, ensuring a higher correlation of BLEU with human judgment, than when only one reference is used.[6]

We perform significance testing by bootstrap resampling for BLEU scores (Koehn, 2004) and F1 scores.[7]

4 Results

Section 4.1 describes the results of punctuation prediction and section 4.2 describes the results of MT of unpunctuated input.

4.1 Punctuation Prediction

4.1.1 Dutch

Table 1 shows the results of the punctuation prediction experiments for Dutch. All MT approaches score significantly better on F1 and BLEU scores ($p < .001$) than the LM approaches. They also score significantly better on F1-score than the LSTM seq approaches ($p < .001$), but only *PBSMT*, *PBSMT clean* and *Hiero* score better on BLEU score ($p < .001$) than any of the LSTM seq approaches. *PBSMT* scores significantly better ($p < .05$) than the other MT approaches on BLEU, but on F1-score it scores only significantly better than *PBSMT clean* ($p < .001$). This difference between BLEU and F1 score can be explained by the fact that the non-PBSMT approaches can reorder the words and perform unwanted transformations other than inserting punctuation (mainly affecting BLEU). This is why we consider the PBSMT approach to punctuation insertion the best approach for this experimental setup.

Of the LM approaches, *n-gram fw+bw* scores significantly better than the other approaches on BLEU and F1 ($p < .001$). Increasing the vocabulary size has only a minor influence on the results: it decreases precision but increases recall, and has no significant effects on BLEU nor on F1. These models tend to overgenerate punctuation, which can be seen from their low precision.

Table 1: Results of punctuation prediction in Dutch

Method	**Prec.**	**Recall**	**F1**	**BLEU**	**TER**	**MET.**
n-gram LMs						
fw 50k	22.63	27.89	24.98	68.69	14.10	87.10
fw full	22.08	28.45	24.86	70.56	13.33	87.69
fw+bw 50k	54.49	78.59	64.36	79.63	8.32	92.88
fw+bw full	53.57	79.15	63.89	80.86	7.78	93.28
LSTM LM fw	44.75	31.83	37.20	83.90	7.97	92.42
LSTM seq						
fw	72.03	11.97	20.53	86.11	8.42	91.12
fw opt	43.70	32.25	37.11	83.45	9.31	90.86
fw+bw	50.23	15.07	23.18	86.84	9.17	90.39
fw+bw opt	41.28	16.34	23.41	86.13	9.74	89.94
PBSMT	92.36	74.93	**82.74**	**94.20**	**2.85**	**97.14**
clean	**93.88**	71.27	81.02	93.76	3.06	96.88
Hiero	83.16	**80.70**	81.92	93.54	3.11	97.16
NMT SGD	84.53	79.30	81.83	85.71	6.88	91.79
Adam	82.43	79.30	80.83	85.04	7.12	91.61

Table 2: Results of punctuation prediction in English

Method	**Prec.**	**Recall**	**F1**	**BLEU**	**TER**	**MET.**
n-gram LM						
fw 50k	12.51	28.31	17.35	50.27	23.04	48.24
fw full	23.69	30.37	26.61	71.96	13.64	54.93
fw+bw 50k	42.62	73.60	53.96	69.21	10.41	53.30
fw+bw full	51.30	79.53	62.35	79.78	8.21	58.87
LSTM fw	35.23	25.10	29.31	80.76	10.37	57.21
LSTM seq						
fw	69.88	7.50	13.54	90.19	8.86	59.75
fw opt	32.88	32.44	32.66	78.79	11.48	56.66
fw+bw	41.53	13.31	20.20	86.34	10.02	58.31
fw+bw opt	39.10	13.83	20.42	85.21	9.93	58.36
PBSMT	86.09	77.15	81.37	**94.76**	**3.12**	**66.12**
clean	83.46	76.77	79.77	93.77	3.45	65.36
Hiero	76.48	79.87	77.97	92.18	3.91	64.32
NMT SGD	**91.41**	82.59	**86.76**	93.53	3.44	64.97
Adam	90.62	**82.63**	86.43	93.78	3.35	65.19

LSTM sequence labeling (*LSTM seq*) does not score better than the LM approaches, mainly because of the low recall. The bidirectional LSTM has a lower precision but a slightly higher recall than the unidirectional LSTM. The *n-gram fw+bw 50k* and *n-gram fw-bw full* methods result in a significantly better F1 score ($p < .001$) than any of the *LSTM seq* methods. In BLEU scores, all *LSTM seq* methods are significantly better than all n-gram LM approaches. This reflects the fact that BLEU is a precision metric. Only the difference between *LSTM fw* and *LSTM seq fw opt* is not significant.

We tested two methods to improve recall for sequence labeling: thresholding for the probability distribution and weighted cross-entropy. Thresholding means that if ⟨*nopunct*⟩ is predicted but the ratio of the probability of the second most probable output over the probability of ⟨*nopunct*⟩ is higher than a certain threshold, we assign the second most probable token as prediction. This method indeed improves the recall of the model but lowers the precision: we report the result after optimizing the threshold for F1 score (‘opt’ in the table). We also

[6] The original BLEU paper (Papineni et al., 2002) also uses multiple references.

[7] We adapted the perl implementation by Mark Fishel for BLEU bootstrap resampling, which is available at https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl to work for F-scores on punctuation insertion.

observe that optimizing for F1 does not result in better quality according to the MT metrics. The optimal threshold for the unidirectional model was 0.3 and for the bidirectional model 0.6. Trading off precision and recall had a much smaller effect on the bidirectional model than on the unidirectional one. Training with weighted cross-entropy, where more weight is given to the punctuation symbols since they are much less frequent than the ⟨*nopunct*⟩-class, has similar effects but has the disadvantage of having to re-train the model and optimize the weights per output class, while the threshold can be optimized during testing.

4.1.2 English

Table 2 shows the results of punctuation prediction for English. As we had three reference sets in the original test set, we present averaged results over punctuation prediction on each of these three sets (we calculate the result for each set separately and average over the three datasets). For BLEU scores we used all three references.

All MT approaches score significantly better than the LM approaches ($p < .001$) They also score significantly better than the *LSTM seq* methods (at least $p < .005$). Similar to punctuation insertion for Dutch, *PBSMT* reaches the best BLEU scores, although not significantly better than *PBSMT clean*, but significantly better than *NMT SGD* and *NMT Adam* (both $p < .05$). With respect to the F1-score, we see that there is no significant difference between *NMT SGD* and *NMT Adam*, but *NMT SGD* scores significantly better ($p < .001$) than the other MT methods. *NMT Adam* scores better than *PBSMT* ($p < 0.05$) and *PBSMT clean* ($p < 0.001$ for two of the three test sets, not significant for the third one), and *Hiero* ($p < .001$).

For LM and sequence labeling, we see similar results as for Dutch, with the exception that limiting the vocabulary to only 50k words decreased the performance much more for English than for Dutch. This might seem surprising given that the Dutch dataset has a much larger vocabulary, but it has many more words that occur only once or a few times (ca. 200k types have a frequency of 5 or less in Dutch, as opposed to ca. 80k in English).

The n-gram LM approaches score much better on F1 score, but they overgenerate, as can be seen from the low precision and lower BLEU scores, when compared to LSTM seq approaches, which seem to undergenerate.

To conclude, we observe that for both Dutch and English the MT approaches work best for punctuation prediction as an isolated task. Since we are mainly interested in punctuation prediction in the context of speech translation, the phrase-based approach is the most promising since it does not cause any reordering of the words, giving the best results according to the MT metrics. We will now examine which approach achieves the best (bilingual) translation quality.

4.2 Translation of unpunctuated input

Table 3 shows the different experimental conditions that are evaluated and will be further explained in the next subsections. The best scores per punctuation strategy are marked in bold, the best scores per translation system are underlined.

4.2.1 Baselines

In the baseline conditions, we train the MT systems on normal, punctuated, tokenized, and truecased source and target text, and tune them on the normal, punctuated, tokenized and truecased development set. We remove all the punctuation from the test set, and let the MT systems translate it. It hence constitutes the *lower bound*.

NMT SGD gets the highest BLEU score, but not significantly better than *PBSMT* and *PBSMT clean*. *Hiero* and *NMT Adam* score significantly worse than the other three conditions ($p < .001$).

4.2.2 Upper Bounds

In the upper bounds conditions, we use the same MT systems as in the baselines, and evaluate them on the normal, punctuated, tokenized and truecased test set, to see how well the MT systems would do with "perfect" input.

Each of the upper bound scores is significantly better ($p < 0.001$) than the same approach in the baseline condition, so using MT without any form of punctuation insertion results in a significant loss in translation quality.

Comparing the different MT systems, *NMT SGD* is significantly better than *PBSMT* ($p < .01$), *PBSMT clean* and *NMT Adam* (both $p < .001$). There is no significant difference between *PBSMT*, *PBSMT clean*, and *NMT Adam*, but all score significantly better than *Hiero* ($p < .001$). Remarkable is the higher METEOR score for *PBSMT*.

Table 3: Results of punctuation insertion + translation.

Punctuation	Translation System														
Insertion	PBSMT			PBSMT clean			Hiero			NMT SGD			Adam		
Method	BLEU	TER	METEOR	BLEU	TER	METEOR	BLEU	TER	METEOR	BLEU	TER	METEOR	BLEU	TER	METEOR
Baselines	43.22	44.69	37.27	42.97	44.37	37.31	39.81	46.69	35.70	**44.01**	**43.80**	**36.40**	38.91	46.85	33.02
Upper bounds	46.19	39.15	**39.08**	46.55	38.80	39.01	42.72	41.59	37.35	**49.13**	**37.66**	38.26	46.30	39.41	37.59
Preprocessing															
n-gram															
fw 50K	36.64	48.53	35.36	35.36	48.11	34.93	35.68	49.35	34.73	36.00	52.76	31.50	34.33	53.05	31.01
fw full	37.63	47.36	36.62	37.46	46.98	36.77	36.19	48.51	35.55	38.76	50.99	33.45	36.60	51.34	32.74
fw+bw 50K	38.98	44.91	36.46	38.25	44.51	36.14	37.97	46.05	35.71	43.51	43.14	35.78	40.43	45.30	34.97
fw+bw full	40.41	43.46	37.81	40.08	43.42	37.83	38.33	45.36	36.40	44.36	42.01	36.68	42.54	43.18	36.45
LSTM fw	39.10	46.08	34.48	38.06	45.52	34.15	37.11	47.24	33.64	38.23	49.25	32.95	35.91	50.85	31.37
LSTM seq															
fw	42.75	44.25	37.22	42.17	44.02	37.15	39.56	46.25	35.76	43.49	43.54	36.19	39.43	46.55	33.43
fw opt	40.44	45.55	36.99	40.29	45.01	37.09	38.88	45.88	35.93	41.81	45.97	35.22	39.21	46.97	34.00
fw+bw	42.01	45.35	37.03	41.57	45.26	36.98	39.96	45.57	35.96	43.78	43.87	36.15	39.69	47.11	33.96
fw+bw opt	41.73	45.83	36.93	10.99	45.91	36.87	40.13	45.40	36.08	43.13	44.94	35.95	39.69	47.11	33.96
PBSMT	45.61	40.26	**38.59**	45.27	40.26	38.56	42.11	42.60	37.01	**47.37**	**39.18**	37.55	45.49	40.79	37.10
clean	45.54	39.98	38.51	45.19	40.27	38.58	41.87	42.70	36.95	47.04	39.40	37.44	45.29	41.05	36.97
Hiero	44.94	40.44	38.57	42.97	44.37	37.32	39.81	46.69	35.70	46.41	39.72	37.12	45.39	40.96	37.08
NMT SGD	44.99	40.49	37.49	44.81	40.56	37.64	40.97	42.90	35.83	47.17	39.35	37.22	45.19	40.77	37.04
Adam	44.65	40.77	37.40	44.53	40.35	37.60	40.94	43.14	35.84	47.05	39.77	37.34	45.23	40.95	37.13
Implicit	44.47	41.65	**38.11**	37.37	44.68	34.77	41.89	42.37	36.78	**47.12**	**38.99**	37.46	44.78	41.24	36.56
Unpunctuated	44.81	42.00	38.55	44.26	41.58	**38.62**	40.27	45.20	36.77	**46.86**	**41.08**	37.55	43.71	43.57	36.52
Postprocessing															
n-gram															
fw 50K	29.62	55.08	35.28	29.05	55.35	35.34	26.75	58.08	33.84	31.36	53.62	34.09	29.23	55.68	33.33
fw full	36.77	48.15	36.47	36.25	47.71	36.45	33.71	50.69	35.02	38.29	47.02	35.22	35.35	49.90	34.32
fw+bw 50K	36.59	46.18	37.08	30.06	45.80	37.26	33.19	49.13	35.54	39.17	45.06	35.97	36.19	47.54	35.05
fw+bw full	38.54	44.62	37.34	38.11	44.74	37.52	34.86	47.72	35.82	41.53	43.47	36.37	38.07	46.33	35.40
LSTM fw	39.67	46.51	36.74	39.04	46.15	36.80	35.81	49.42	35.10	41.78	45.53	35.70	39.20	47.73	34.92
LSTM seq															
fw	41.71	44.90	36.76	41.00	44.43	36.72	37.57	47.64	35.09	42.69	44.72	35.68	39.86	46.69	34.70
fw opt	40.66	46.44	36.43	37.68	46.77	36.57	34.46	50.24	34.98	39.62	46.52	35.40	37.17	48.84	34.48
fw+bw	40.66	46.44	36.43	39.88	46.29	36.40	36.93	49.14	34.81	41.86	45.95	35.34	39.12	47.98	34.34
fw+bw opt	40.51	46.67	36.41	39.66	46.54	36.38	36.71	49.52	34.81	41.70	46.09	35.30	39.01	48.07	34.32
PBSMT	45.10	40.94	38.31	44.75	40.41	38.44	41.11	43.62	36.63	46.73	40.21	37.34	43.58	42.44	36.33
clean	45.05	40.83	38.30	44.73	40.33	38.43	41.06	43.58	36.61	46.64	40.20	37.30	43.60	42.45	36.30
Hiero	44.41	41.61	38.25	43.85	41.11	38.39	40.38	44.27	36.59	46.04	40.73	37.21	43.14	43.08	36.26
NMT SGD	44.87	40.62	38.34	43.95	41.15	38.40	40.65	43.45	36.66	**47.00**	**39.68**	**37.48**	44.17	41.65	36.46
Adam	44.54	40.78	38.10	44.04	40.61	38.25	40.50	43.30	36.50	46.89	**39.68**	37.38	44.21	41.57	36.47

.

4.2.3 Preprocessing

In the preprocessing conditions, we first insert punctuation, as described in section 4.1, before translating. The output of the punctuation insertion is then translated using a regular MT system, trained on punctuated data.

Using LM and LSTM seq as preprocessing approach never helps significantly over the baseline, only in the case of *ngram fw+bw full + NMT Adam* ($p < .001$). Using monolingual MT as preprocessing nearly always helps ($p < .005$), except when using *Hiero* as preprocessing or as translation engine. Whether *PBSMT* or *PBSMT clean* are used as preprocessors does not make a significant difference. When using *NMT SGD* as translation method, the kind of monolingual MT (apart from Hiero) does not play a significant role.

The best preprocessing results (using PBSMT as punctuation inserter) score still significantly lower than the upper bound scores when using the same translation system ($p < .05$ for *PBSMT*, *PBSMT clean* and *NMT Adam*, $p < .01$ for *Hiero* and $p < .001$ for *NMT SGD*).

4.2.4 Implicit Punctuation Insertion

We remove punctuation from the source side of the parallel corpus and train the MT engines on these data, so they should be well-suited to translate source text without punctuation in target text with punctuation.

The score for implicit translation using *NMT SGD* is not significantly worse than preprocessing *PBSMT + NMT SGD*. *NMT SGD* scores significantly better ($p < .005$) than all other implicit punctuation insertion methods.

4.2.5 Unpunctuated

We have tested the MT systems trained on unpunctuated data both in the source and the target, and evaluated against references from which the punctuation is also removed. As we use a different version of the references, we cannot apply significance testing. We present these results as they provide an indication about the maximum score we can expect for the postprocessing approach.

Even without punctuation inserted, it is clear that the scores are much lower than the *Upper bounds* presented earlier. The presence of punctuation thus improves the bilingual translation quality in general.

4.2.6 Postprocessing

In the postprocessing approach, we translate using MT systems trained on unpunctuated data (both source and target), resulting in a translation that does not contain punctuation. The postprocessing step consists of punctuation insertion, similar to the preprocessing punctuation insertion step, but now for English.

Postprocessing with LM and LSTM seq does not yield any improvements over the baseline. With monolingual MT we reach significance in all cases where we use *PBSMT* ($p < .001$), *PBSMT clean* ($p < .001$) and *NMT SGD* ($p < .05$). *NMT Adam* also improves over the baseline ($p < .05$), except when combined with the *Hiero* system.

4.2.7 General results

We note the lack of significant difference between pre- and postprocessing in the cases where punctuation insertion consists of *PBSMT*, *PBSMT clean*, *NMT SGD* or *NMT Adam* and translation consists of *PBSMT*, *PBSMT clean*, *NMT SGD* or *NMT Adam*.

When considering how much we can close the gap between *upper bound* and *baseline* using the best scoring combination of methods for each of the translation systems, we note gap closure of 80% for *PBSMT*, 64% for *PBSMT clean*, 79% for *Hiero*, 66% for *NMT SGD* and 89% for *NMT Adam*.

5 Conclusions and Future Work

We set out to compare different approaches to punctuation prediction in the context of translation. We test several different architectures and methods for punctuation prediction as well as for MT, all trained on the exact same data sets, and evaluate the punctuation prediction quality as a monolingual phenomenon, as well as its effect on MT quality.

While there is a clear deterioration of MT quality when working with unpunctuated input, this gap can be closed for 66% in the case of our best bilingual MT system, NMT, by applying monolingual MT as punctuation insertion, or by using a dedicated implicit insertion MT system.

Whether we use pre- or postprocessing did, in most cases, not result in a significant difference, indicating that the general punctuation prediction quality for Dutch is similar to that of English.

In future work, we would like to develop a similar experiment for *segmentation prediction*, and test the results on real speech signals in order to determine the usefulness of the results in a more realistic setting. A possible improvement would be to use NMT as punctuation prediction model, but constrain the word order with the help of the attention weights, thus combining the advantage of neural MT with the constraints on reordering of PBSMT.

Acknowledgements

This research was done in the context of the SCATE project, funded by the Flemish Agency for Innovation and Entrepreneurship (IWT project 13007).

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow.org*.

Beeferman, Doug, Adam Berger and John Lafferty. 1998. Cyberpunc: A lightweight punctuation annotation system for speech. *IEEE Conference on Acoustics, Speech and Signal Processing*. 689–692.

Kim, Ji-Hwan and P.C. Woodland 2001. The use of prosody in a combined system for punctuation generation and speech recognition. *Proceedings of EuroSpeech*.

Christensen, Heidi, Yoshihiko Gotoh and Steve Renals. 2001. Punctuation Annotation using Statistical Prosody Models. *Proceedings of ISCA Workshop on Prosody in Speech Recognition and Understanding*.

Huang, Jing and Geoffrey Zweig. 2002. Maximum entropy model for punctuation annotation from speech. *Proceedings of ICSLP*.

Gravano, Agustín, Martin Jansche and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. *Proceedings ICASSP*.

Chen, Stanley F. and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 17:359–394.

Chiang, David. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Denkowski, Michael and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. *Proceedings of WMT*. 376–380.

Duchi, John, Elad Hazan and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Gale, William and Sarangarajan Parthasarathy. 2017. Experiments in Character-level Neural Network Models for Punctuation. *Proceedings Interspeech*. 2794–2798.

Hassan, Hany, Yanjun Ma and Andy Way. 2007. Matrex: the DCU machine translation system for IWSLT 2007. *Proceedings IWSLT*. 69–75.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation*, 9(8):1735–1780.

Jean, Sébastien, Kyunghyun Cho, Roland Memisevic and Yoshua Bengio. 2014. On Using Very Large Target Vocabulary for Neural Machine Translation. *arXiv preprint arxiv:1412.2007*.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv preprint arxiv:1701.02810*.

Kingma, Diederik P. and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arxiv:1412.6980*.

Koehn, Philip. 2004. Statistical Significance Tests for Machine Translation Evaluation. *Proceedings EMNLP*. 388–395.

Koehn, Philip. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *Proceedings MT Summit X*. 79–86.

Koehn, Philip, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of ACL Demonstration Sessions*. 177–180.

Lee, Young-suk and Salim Roukos 2006. IBM Spoken Language Translation System. *Proceedings TC-STAR Workshop on Speech-to-Speech Translation*. 13–18.

Lu, Wei and Hwee Tou Ng. 1998. Better punctuation prediction with dynamic conditional random fields. *Proceedings EMNLP.* 177–186.

Luong, Minh-Thang, Hieu Pham and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025.*

Matusov, Evgeny, Arne Mauser and Hermann Ney. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. *Proceedings IWSLT.* 158–165.

Matusov, Evgeny, Nicolas Ueffing and Hermann Ney. 2006. Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment. *Proceedings EACL.* 33–40.

Moró, Anna and György Szaszák. 2017. A phonological phrase sequence modelling approach for resource efficient and robust real-time punctuation recovery. *Proceedings Interspeech.* 558–562.

Och, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proceedings of ACL.* 160–167.

Pahuja, Vardaan, Anirban Laha, Shachar Mirkin, Vikas Raykar, Lili Kotlerman and Guy Lev. 2017. Joint Learning of Correlated Sequence Labeling Tasks Using Bidirectional Recurrent Neural Networks. *Proceedings Interspeech.* 548–552.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings ACL.* 311–318.

Peitz, Stephan, Markus Freitag, Arne Mauser and Hermann Ney. 2011. Modeling Punctuation Prediction as Machine Translation. *Proceedings IWSLT.* 238–245.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of AMTA.* 223–231.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky,Ilya Sutskever and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Stolcke, Andreas. 2002. SRILM an extensible language modeling toolkit. *Proceedings International Conference Spoken Language Processing.* 901–904.

Tilk, Ottokar and Tanel Alumäe. 2015. LSTM for Punctuation Restoration in Speech Transcripts. *Proceedings Interspeech.* 683–687.

Tilk, Ottokar and Tanel Alumäe. 2013. Bidirectional Recurrent Neural Network With Attention Mechanism for Punctuation Restoration Transcripts. *Proceedings Interspeech.* 3047–3051.

Ueffing, Nicola, Maximilian Bisani and Paul Vozila. 2013. Improved models for automatic punctuation prediction for spoken and written text. *Proceedings Interspeech.* 3097–3101.

Vandeghinste, Vincent, Scott Martens, Gideon Kotzé, Jörg Tiedemann, Joachim Van den Bogaert, Koen De Smet, Frank Van Eynde and Gertjan van Noord. 2013. Parse and Corpus-based Machine Translation. *Essential Speech and Language Technology for Dutch*, chapter 17, Peter Spyns and Jan Odijk (eds.), 305–319.

Weiss, Ron, Jan Chorowski, Navdeep Jaitly, Yonghui Wu and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. *Proceedings Interspeech.* 2625–2629.

Zhang, Zhu, Michael Gamon, Simon Corston-Oliver and Eric Ringger. 2002. Intra-sentence punctuation insertion in natural language generation. *Microsoft Technical Report.*

User papers

Integrating MT at Swiss Post's Language Service: preliminary results

Pierrette Bouillon, Sabrina Girletti
FTI/TIM, University of Geneva
Boulevard du Pont-d'Arve 40
1211 Geneva, Switzerland
Pierrette.Bouillon@unige.ch
Sabrina.Girletti@unige.ch

Paula Estrella, Jonathan Mutal
FaMaF, University of Córdoba
Av. Medina Allende s/n
X5000HUA Córdoba, Argentina
pestrella@famaf.unc.edu.ar
jdm0113@famaf.unc.edu.ar

Martina Bellodi, Beatrice Bircher
Swiss Post Ltd
Wankdorfallee 4
3030 Bern, Switzerland
martina.bellodi@post.ch
beatrice.bircher@post.ch

Abstract

This paper presents the preliminary results of an ongoing academia-industry collaboration that aims to integrate MT into the workflow of Swiss Post's Language Service. We describe the evaluations carried out to select an MT tool (commercial or open-source) and assess the suitability of machine translation for post-editing in Swiss Post's various subject areas and language pairs. The goal of this first phase is to provide recommendations with regard to the tool, language pair and most suitable domain for implementing MT.

1 Introduction

Nowadays, the production environments of many companies incorporate MT for various reasons: it might be upon request of a client, an initiative to add new services to a company's assets or an attempt to cut costs and shorten delivery times. The technology can be developed by a third party or in-house, each solution having its own pros and cons.

Swiss Post's Language Service would like to integrate MT in their workflow in different contexts, ranging from gisting to professional post-editing, thereby allowing for reduced turnaround times. Hence, in collaboration with the University of Geneva and one of its partners, the University of Córdoba, a preliminary study was carried out to 1) select an MT engine (open source or commercial) and 2) determine the language pairs and subject areas for which MT would be most suitable. In particular, we focused on assessing the potential suitability of MT sentences for professional post-editing.

The source data used to train and test the different systems for the various language pairs are almost parallel, making it possible to compare results across less-studied pairs. In addition, when designing our experimental setting, we chose to put the focus on users, namely Swiss Post's professional translators, providing them with specific training before involving them in the evaluation process. We are convinced that when reorganizing the traditional workflow of professional translators, it is important to give them an active role in the change in order to foster acceptance and avoid biased evaluation due to reluctant MT users.

The paper is structured as follows: we first describe the available data for the various languages and subject areas (Section 2), then explain how we selected the MT engine (Section 3). We then present how the suitability of the MT for PE was assessed by Swiss Post's in-house translators (Section 4) and discuss the results (Section 4.4), before concluding (Section 5).

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 281–286
Alacant, Spain, May 2018.

2 Data and subject areas

Swiss Post's Language Service primarily translates texts from DE(CH) into FR(CH), IT(CH) and EN(UK). The Service has diverse activities, with specific translation memories (TMs) available in different subject areas: vocational training (denoted *Modulo*), financial services (*PF*), process manuals (*PN*), and annual report (denoted *GB*). In addition, there is a big "master" TM (denoted *MTM*) which includes all the specific TMs, plus additional material. The data are almost parallel across language pairs, meaning that at least 65% of source sentences are shared as training data[1]. Since the volume of translated material is significantly lower for DE-EN, we decided to only consider the "annual report" (*GB*) domain for this language pair. Details on amount of data are shown in Table 1.

TMs	**DE-FR**	**DE-IT**	**DE-EN**
Modulo	99,612	107,128	–
PF	129,694	122,568	–
PN	23,131	23,447	–
GB	38,580	37,721	32,857
MTM	2,558,148	1,929,530	417,817

Table 1: Number of translation units in TMs, per language pair.

The language pairs involved in this project are quite challenging, as they involve highly inflected languages (German, French and Italian). Furthermore, language pairs such as DE-IT and DE-FR are underrepresented in the vast literature on MT, as most of the results deal with English (either as the source or target).

3 MT system selection

3.1 Solutions considered

The first part of the study was devoted to a comparative evaluation between two phrase-based MT engines: the open-source toolkit Moses (Koehn et al., 2007) and the commercial online platform offered by Microsoft (Translator Hub, MTH[2]).

These solutions are common options for a company willing to experiment with MT; one is a third-party platform – which only requires uploading data (and then paying for the deployment and employment of the system) – while the other is an in-house solution, which, on the one hand, allows the entire process to be fully controlled, but on the other hand, requires technical knowledge and computing resources.

3.2 Engine training and evaluation

We followed the training process (corpus tokenization, language and translation model training, tuning and testing on a disjoint set from training) using the tools provided by Moses and MTH[3]. After some experimenting, language models for Moses were trained using KenLM (Heafield, 2011) on 4-grams. For models created in MTH, additional preprocessing was needed before building systems as data had to be anonymized for confidentiality. Therefore, named entities, numbers (belonging to phone numbers, amounts, accounts, etc.), urls and emails were replaced by placeholders in training and test data.

Since there are specific TMs for each subject area and language pair, we tried different combinations in order to obtain the highest automatic scores. Using each specific TM individually (*PN*, *Modulo*, *PF*, *GB*) resulted in a small-sized training set leading to poor automatic scores, so we decided to perform two incremental rounds of training:

- Round 1 - using all TMs together as a mixed training set: in this case we tested them on the different domains to explore how the system performed. Both Moses and MTH models were trained for DE-IT/FR.[4]

- Round 2 - using only the *MTM*: in this case we did not train models in MTH, as previous tests had indicated that the results with Moses were better and we could therefore save on the cost of anonymizing the data.

Models were evaluated automatically using standard metrics BLEU (Papineni et al., 2002)[5] and Word Error Rate (WER), as well as internal human evaluations. Four different test sets, one per

[1] Between DE-FR and DE-IT. The percentage is lower for pairs with EN, as this corpus is significantly smaller than the others.

[2] https://www.microsoft.com/en-us/translator/hub.aspx

[3] For training processes, see:
http://www.statmt.org/moses/?n=Moses.Baseline
https://hubtest.microsofttranslator-int.com/Help/Download/Microsoft%20Translator%20Hub%20User%20Guide.pdf

[4] The DE-EN pair was added to the study in a second phase.

[5] Although MTH provides BLEU scores after training, we report BLEU scores calculated using the script available at ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13.pl.

domain, were used to test the behavior of the engine when confronted with new data (not included in the training corpus). Amount of testing data is shown in Table 2.

Test set	**DE-FR**	**DE-IT**	**DE-EN**
PN	1736		–
Modulo	2034		–
PF	1919	2378	–
GB	1829	1718	704

Table 2: Number of translation units in test set per language and domain. For *Modulo* and *PN*, IT and FR shared exactly the same source sentences, while in the other domains, at least 58% of the corpus was shared. This percentage is lower with EN, since the related corpus was significantly smaller.

3.3 Results of MT engine evaluation

Results for Round 1 of training are shown in Tables 3 and 4: Moses outperforms MTH in all domains and better scores are obtained for *PN*. On the basis of these results, Round 2 of training was implemented; we only trained Moses on *MTM* to avoid having to anonymize data sets. Results improved for all domains (see Table 5).

	Moses		**MTH**	
Test set	WER	BLEU	WER	BLEU
PN	43.93	0.51	55.11	0.36
Modulo	45.94	0.46	60.17	0.31
PF	50.92	0.40	63.84	0.28
GB	58.49	0.34	71.91	0.23

Table 3: Results for DE-FR on mixed training set (all TMs).

	Moses		**MTH**	
Test set	WER	BLEU	WER	BLEU
PN	40.40	0.52	52.68	0.37
Modulo	44.16	0.46	55.55	0.35
PF	46.43	0.43	58.36	0.32
GB	51.94	0.40	62.66	0.31

Table 4: Results for DE-IT on mixed training set (all TMs).

We concluded that we could safely proceed to the human evaluation of suitability for PE (detailed in Section 4) with only Moses trained on *MTM*.

Test set	lang/pair	WER	BLEU
PN	DE-IT	33.01	0.6
	DE-FR	34.39	0.61
Modulo	DE-IT	40.96	0.5
	DE-FR	43.53	0.5
PF	DE-IT	43.07	0.48
	DE-FR	41.14	0.52
GB	DE-IT	47.41	0.45
	DE-FR	54.28	0.39
	DE-EN	34.48	0.62

Table 5: Results for DE-FR/IT/EN on MTM.

4 Human evaluation: suitability of MT for PE

4.1 Goal

The aim of the evaluation was to assess the potential suitability[6] of MT for post-editing in various language pairs and subject areas, from the perspective of Swiss Post's translators. We decided to let the translators assess the quality of the segments first, before involving them in a real post-editing task, in order to give them an idea of expected quality.

4.2 Test data

For the human evaluation, we used four specific test sets. We randomly selected a sample of 250 German sentences per subject area (1000 sentences in total) from the original test sets (described in Table 2), along with their respective target translations in FR, IT and EN. The test sets are completely parallel, meaning that we selected exactly the same 250 source sentences per subject area across the three language pairs. As in the previous evaluation, we only used the subject area "annual report" (*GB*) for the DE-EN pair. The automatic scores for these specific test sets are shown in Table 6.

4.3 Methodology

Eight in-house translators of the Language Service participated in the test team: three for DE-FR and DE-IT, and two for DE-EN. All translators in the test team had been working at Swiss Post's Language Service for at least 6 months, and had 1 to 19 years of translation experience. Before performing the evaluation task, the test team was given a one-day training course on MT and PE, involving both

[6] We also use "usability of MT for PE" as a synonym for "suitability".

Test set	lang/pair	WER	BLEU
PN	DE-IT	35.91	0.58
	DE-FR	35.20	0.59
Modulo	DE-IT	41.88	0.48
	DE-FR	47.52	0.46
PF	DE-IT	47.32	0.41
	DE-FR	47	0.43
GB	DE-IT	47.46	0.43
	DE-FR	58.77	0.34
	DE-EN	41.78	0.51

Table 6: BLEU and WER scores for test set (250 sentences), per domain and language pair.

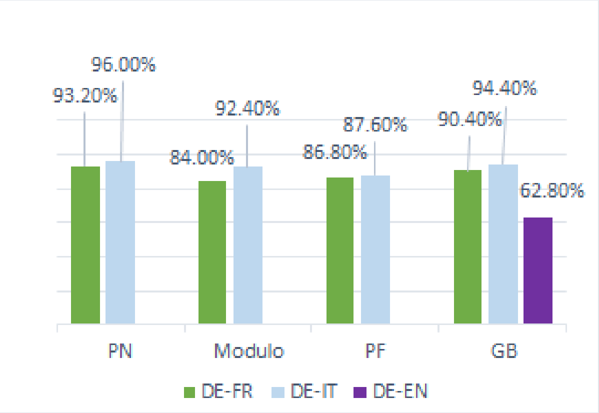


Figure 1: Percentage of machine translated sentences suitable for PE, per domain and language pair.

theory and practical exercises on MT engine training, evaluation and post-editing.

Since the purpose of this human evaluation was to assess the actual suitability of machine-translated sentences for subsequent post-editing by professional translators, we decided to use a customised metric. For each source sentence in the test sets, translators were presented with a raw machine translation and were requested to answer the following question: *"In a post-editing task, would you reuse this translation?"*, with possible answers being *"Yes, I would leave it as it is"* (denoted "Yes"), *"Yes, I would use it with some changes"* (denoted "YwC") and *"No, I would translate from scratch"* (denoted "No"). Since the evaluators were already familiar with the material being evaluated, we did not include any reference translation in our test. However, the translators were aware of the origin (that is, the subject area) of each segment, so that they could evaluate if the terminology used was appropriate.

We are aware that the "YwC" category is too broad, as it comprises all segments requiring minor changes or intensive post-editing, but in this preliminary evaluation we were mostly interested in finding out whether the translators would accept to post-edit the raw MT.

4.4 Results of human evaluation

Figure 1 summarises the results in terms of percentage of sentences suitable for PE, calculated as the sum of all "Yes" and "YwC" majority judgments[7], divided by the total number of sentences. In FR and IT, the results were very encouraging, with between 84% and 96% suitable sentences for each test set. The subject area *PN* obtained the best ratings, for both IT and FR; this result was also confirmed by automatic metrics (see Table 6). The second best domain was "annual report" (*GB*), with IT evaluators assessing a higher percentage of usable sentences than their FR and EN colleagues. However, this contradicts automatic scores, where GB seemed to be the subject area in which MT performed the worst. This calculation somewhat prejudices the scores of *GB* in EN, since there were only two evaluators and we only counted the sentences for which they agreed.

An Inter-Rater Reliability (IRR) analysis was performed to assess consistency among nominal ratings provided by the evaluators. Light's kappa (Light, 1971) and Cohen's kappa for DE-EN were used as an index of IRR. Figures are shown in Table 7.

	DE-FR	DE-IT	DE-EN
PN	0.341	0.549	–
Modulo	0.411	0.547	–
PF	0.412	0.519	–
GB	0.340	0.562	0.430

Table 7: Figures of Light's kappa (DE-FR/IT) and Cohen's kappa (DE-EN).

Overall, the results show moderate agreement among evaluators, with the exception of two domains (*PN* and *GB*) in DE-FR, where agreement is "fair" (Landis and Koch, 1977). Results are therefore more reliable for DE-IT.

Tables 8, 9 and 10 report detailed results per language pair. These results confirm that for DE-FR and depending on the domain, between 20-22% of the segments would not require post-editing at all ("Yes" column) and between 63.6-71.2% would

[7] Majority judgments are judgments on which at least two of the evaluators agree.

DE-FR			
ratings %	Yes	YwC	No
*PN**	22	71.2	5.2
*Modulo**	20.4	63.6	15.6
PF	22	64.8	13.2
*GB**	20	70.4	9.2

Table 8: Detail of ratings, per domain. For test sets *PN*, *Modulo* and *GB*, majority ratings could not be counted for, respectively, 2%, 0.4% and 0.4% of sentences.

DE-IT			
ratings %	Yes	YwC	No
*PN**	32.4	63.6	3.6
Modulo	31.6	60.8	7.6
*PF**	22.8	64.8	12
GB	26.8	67.6	5.6

Table 9: Detail of ratings, per domain. Majority ratings for the test sets *PN* and *PF* could not be counted for 0.4% of sentences.

DE-EN				
ratings %		Yes	YwC	No
GB	min.	14.8%	67.6%	17.6%
	maj.*	9.2%	53.6%	9.2%

Table 10: Detail of minimal (min.) and majority (maj.) ratings, per domain. Since only two evaluators were involved in this task, majority ratings could not be counted for 28% of sentences.

require some post-editing, but could still be used. What remains to be studied is the effort it would take the translators to post-edit those segments in column "YwC" to convert them into a polished final translation. It is worth noting that the amount of segments that would be translated from scratch is minimal. Using the majority judgment, 2% of segments were scored in disagreement (i.e., they received three different scores).

Table 9 shows detailed results for DE-IT. The percentage of sentences in the "Yes" category was even higher than for the DE-FR language pair. In particular, the domain PN had the highest percentage of sentences usable without any modification ("Yes') and the lowest percentage of non-usable sentences ("No') overall.

For the DE-EN language pair, sentences could mostly be used with some changes. An equal percentage of "Yes" and "No" was also reported. However, it is worth noting that 28% of the sentences could not be counted. Since only two EN evaluators participated in the task, the majority judgment became a unanimous judgment, and we were not able to assess whether that third of the segments might be usable for post-editing. That is why, in Table 10, we also report minimal judgments, i.e. we count the times each nominal category received *at least* one score. When adding missing judgments to the count, more sentences are rejected and fewer sentences are accepted without any changes. However, in this particular case, we would need further analysis to confirm if DE-EN produces MT less suitable for post-editing, or if the evaluators are less inclined to use the raw MT output.

Both human and automatic evaluations confirmed that "process manuals" (*PN*) is the best domain for the Language Service to begin implementing MT. We therefore focused on this domain to see what influences the subjective judgments of suitability, and further study if they correlate with objective factors (length of sentences, quality of raw translation).

As shown in Figure 2, translations assessed as "usable without modifications" ("Yes") are clearly shorter than the average length of source sentences in the corpus, while non-usable sentences ("No") are longer. The overall most chosen category, "YwC", comprises sentences that are generally longer than the average source sentence length.

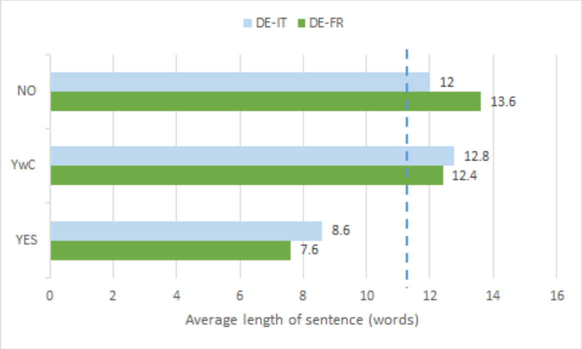


Figure 2: Average length of source sentences evaluated by IT and FR translators for the *PN* test set. Average sentence length in the *PN* source corpus is 11.37 words.

In table 11 we can see that WER scores also vary in line with suitability and that Light's kappa, calculated for each category, is inversely proportional to WER scores ("Yes" > "YwC" > "No").

Finally, we found that the amount of sentences that overlap in each category for both FR and IT is between 42%(IT) and 62% (FR) of "Yes" judgments, versus only between 31% (FR) and 44% (IT) for "No". These latest results are encouraging: they confirm that subjective judgments can be related to objective factors and that, in general, "Yes" judgments are very reliable, while "No" and "YwC" judgments seem to depend more on language, translators' choices and personal opinions.

lang. pair	metrics	**Yes**	**YwC**	**No**
DE-FR	%	22	71.2	5.2
	WER	20.40	37.34	65.16
	Kappa	0.462	0.282	0.235
DE-IT	%	32.4	63.6	3.6
	WER	22.99	42.68	71.29
	Kappa	0.64	0.514	0.339

Table 11: Detail of ratings (%) for *PN* compared to WER scores and Light's kappa (k) figures on the specific set of majority judgments for each category.

5 Conclusion and future work

We have presented the preliminary results of a project that aims to integrate MT into the workflow of Swiss Post's Language Service.

The first part of the study was devoted to choosing between the commercial Microsoft Translator Hub system and an in-house trained Moses solution, both trained using Language Service's material. We decided to proceed with the latter, trained on *MTM*, since automatic scores were systematically better with this system and training configuration. This allowed us to use just one system per language pair.

In the second part of the study, a human evaluation was carried out to assess the percentage of raw MT sentences perceived as suitable for professional post-editing. A sample of Swiss Post Language Service's professional translators was actively involved in this task. The outcomes of the evaluation were overall better for the subject area "process manuals" (*PN*). DE-IT evaluators assessed the highest percentage of usable sentences (with or without changes). More agreement among evaluators was also reported for this language pair. However, we sometimes found contradictions between human results and automatic scores, for instance in DE-EN, likely due to the fact that we only had two evaluators for this language pair. Furthermore, *GB* scored worse with automatic metrics, but was the second best subject area, according to human evaluation. Further investigation is required to discover the reasons behind this inconsistency between human ratings and automatic scores.

All in all, we consider our results to be satisfactory: a percentage of usable sentences ranging from 84% (DE-FR) to 96% (DE-IT) is a good threshold to start working with MT in a professional context. As for DE-EN, the 62.80% obtained suggests that in this case, raw MT output might be suitable, but to a lesser extent, so further work should be done in this direction.

In the next phase, we will carry out a productivity test with the translators, in order to determine if implementing MT into Language Service's workflow could actually be cost effective. These tests will first involve the highest scored domain (*PN*), since we believe that a gentle introduction to MT as new working tool is necessary to make the most of it. Finally, once translators are used to the new workflow, we would like to carry out a comparative evaluation of our PBMT system with the neural baseline we are currently training. This will allow us to compare both translators' productivity and satisfaction when using different MT architectures.

References

Heafield, Kenneth. 2011. KenLM: Faster and smaller language model queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation* pp. 187–197, ACL.

Koehn, Philipp and Hoang, Hieu and Birch, Alexandra and Callison-Burch, Chris and Federico, Marcello and Bertoldi, Nicola and Cowan, Brooke and Shen, Wade and Moran, Christine and Zens, Richard and others. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th ACL*, ACL.

Landis, J Richard and Koch, Gary G. 1977. The measurement of observer agreement for categorical data. *Biometrics*, JSTOR, pp. 159–174.

Light, Richard. 1971. Measures of response agreement for qualitative data: Some generalizations and alternatives. *Psychological bulletin* vol. 76 nr. 5, American Psychological Association.

Papineni, Kishore and Roukos, Salim and Ward, Todd and Zhu, Wei-Jing. 2002. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th ACL*, pp. 311–318, ACL.

Iterative Data Augmentation for Neural Machine Translation: a Low Resource Case Study for English–Telugu

Sandipan Dandapat and Christian Federmann
Microsoft AI & Research
{sadandap,chrife}@microsoft.com

Abstract

Telugu is the fifteenth most commonly spoken language in the world with an estimated reach of 75 million people in the Indian subcontinent. At the same time, it is a severely low resourced language. In this paper, we present work on English–Telugu general domain machine translation (MT) systems using small amounts of parallel data. The baseline statistical (SMT) and neural MT (NMT) systems do not yield acceptable translation quality, mostly due to limited resources. However, the use of synthetic parallel data (generated using back translation, based on an NMT engine) significantly improves translation quality and allows NMT to outperform SMT. We extend back translation and propose a new, iterative data augmentation (IDA) method. Filtering of synthetic data and IDA both further boost translation quality of our final NMT systems, as measured by BLEU scores on all test sets and based on state-of-the-art human evaluation.

1 Introduction

In the past two decades, machine translation (MT) has shown very promising results, most of which have been achieved using data-driven techniques. In recent years, the data-driven paradigm of MT is largely dominated by neural machine translation (NMT) and showing significant success over its predecessor statistical machine translation (SMT) (Bahdanau et al., 2014; Bojar et al., 2017). The performance of any data-driven approach to MT mostly depends on the amount of parallel corpora available to train them. This problem is exacerbated by NMT, which generally needs larger quantities of parallel data and is less robust to noisy data. Unfortunately, large amounts of readily available parallel resources exist only for a small number of languages, e.g., OPUS (Tiedemann and Nygaard, 2004) and Europarl (Koehn, 2005), with only very few sources of Indic language data.

Indic language MT is difficult due to complex linguistic structure and lack of good quality data. Most of the Indic languages are leading languages of the world in terms of number of speakers but are very poorly resourced (i.e., only very little machine-readable parallel text exists) so building a general domain data-driven MT system is a challenging problem. Also, Indic languages do not have enough comparable resources to explore extraction of useful parallel content from the same (Irvine and Callison-Burch, 2013). Lastly, due to the usage of multiple fonts and encodings, a significant portion of the web data cannot be used to extract parallel data for training. Telugu is no exception to this. Lack of large, high quality parallel resources makes the development of general purpose MT systems much harder for Telugu compared to other, resource rich languages, more specifically when building NMT-based models.

One of the major problems with training an NMT system on little data, especially when training an engine for general usage (i.e., not domain specific), is overfitting. Deep neural networks have large parameter spaces and need ample amounts of data in order to generalize adequately; with small amounts of data they tend not to generalize well. We address this issue by learning the optimizer over a smaller number of training steps.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 287–292
Alacant, Spain, May 2018.

In this paper, we describe our English–Telugu (En–Te) general purpose MT system. First, we describe the baseline SMT-based and NMT-based systems trained on 750k parallel sentences. Telugu is a morphologically rich language and, as such, suffers from a high out-of-vocabulary (OOV) rate in a low data scenario. We address data sparsity by augmenting a large amount of synthetic training data (Sennrich et al., 2015), generated using back translation, to iteratively improve the NMT systems. The iterative process uses synthetic data to improve the MT engine and (implicitly) the quality of the synthetic data using the improved MT engine in the reverse direction.

Secondly, we use sub-word representations to reduce the data sparsity problem. This essentially handles Telugu's rich morphology. Furthermore, as translation quality varies across sentences while generating synthetic data, we filter poor quality translation pairs to augment the system only with high quality synthetic parallel data. We observe improved translation quality as a result. The main finding of this work is that the use of iterative data augmentation and filtering of the synthetic data help to improve the translation quality.

The rest of the paper is organized as follows. Section 2 describes the data sets used to build the systems. In Section 3, we describe the baseline SMT and NMT models and their quality. Section 4 provides details on improving NMT models using synthetic data. Section 5 reports the experimental setup and results. We conclude in Section 6.

2 Data Sets

In this work, we use two types of training data: true parallel data, and synthetically generated parallel data using back-translation (Sennrich et al., 2015). In this section, we describe the true English–Telugu parallel data used for system training. The generation of synthetic data is explained in Section 4.1. The full training data contains 750k true parallel sentences along with a larger set of synthetic data (15.4M and 8.2M for En→Te and Te→En, respectively).

The true parallel data includes automatically extracted parallel sentences from the web and from OPUS (Tiedemann and Nygaard, 2004). Many web pages feature content available in multiple languages. Such content includes both sentence or paragraph aligned parallel data (e.g., TED talks' transcriptions) and comparable or noisy-parallel corpora (e.g., cross-lingually linked Wikipedia documents). Once such potential parallel pages between Telugu and English are extracted from the web, a sentence aligner is used to extract sentence aligned parallel text, based on a modified Moore Sentence Aligner (Moore, 2002).

Test Data To the best of our knowledge, there are no publicly available test sets for evaluating Te–En MT systems. Thus, we have created two different test sets to evaluate our systems. Our first test set was created by selecting sentences from news articles. The English source sentences were manually translated into Telugu and validated by human experts. We shall refer this test set as **News**.

In order to understand the performance of our systems w.r.t. state-of-the-art test sets, we have created our second test set using a subset of the WMT 2009 (Callison-Burch et al., 2009) test set for English–French. 1,000 English sentences were randomly selected and manually translated into Telugu by human experts. We call this test set **WMT**. Table 1 summarizes the different data used for training and testing.

Parallel Data	#sentences	#En	#Te
Train	751,609	13.6	10.4
News (test set)	5,000	14.4	10.9
WMT (test set)	1,000	22.8	16.4
Dev	2,500	20.4	14.3
Monoligual Data			
English	8.2m	15.7	–
Telugu	15.4m	–	8.6

Table 1: Number of sentences (#sentences) and average sentence lengths (#En, #Te) for data sets used in this work.

Note that we have created our test sets with a single reference translation. We intend to publicly release the test sets. Monolingual data mentioned in Table 1 is used to build the language models for SMT systems and to generate synthetic parallel data used to train the NMT systems.

3 Baseline Models

The baseline SMT models use a vanilla **phrasal** (Koehn et al., 2003) and a **treelet**[1] (Quirk et al., 2005; Bach et al., 2009) translation model for Te→En and En→Te systems, respectively. We do not use treelet translation system in the Te→En

[1] Extracts treelet translation pairs using source language dependency parse tree and an unsupervised alignment algorithm. This is used for tree-based reordering.

direction due to lack of a Telugu parser. For both phrasal and treelet systems, word alignment is done using GIZA++ (Och and Ney, 2003). We use the target side of the parallel corpus along with additional monolingual target language data to train a 5-gram language model using modified Kneser–Ney smoothing (Kneser and Ney, 1995). Finally, we use MERT (Och, 2003) to estimate the lambda parameters using the held out *Dev* data with a single reference translation.

The baseline NMT model is developed based on the architecture described in (Devlin, 2017). The encoder uses a 3-layer bi-directional RNN (consists of 512 LSTM units). The decoder uses an LSTM layer in the bottom to capture the context and the attention. The LSTM layer is then followed by 5 fully-connected layers applied in each timestep using a ResNet-style skip connection (He et al., 2016). The details of the model and equations are described in (Devlin, 2017). All the models are trained using ADAM optimizer (Kinga and Adam, 2015) with a dropout rate of 0.25. The optimizer uses 100k and 500k steps with a batch size of 1024 for En→Te and Te→En baseline NMT systems, respectively. In the case of Te→En NMT system, source-side Telugu sentences are represented using byte-pair encoding (BPE) (Sennrich et al., 2015) to reduce the data sparsity problem, which uses 50,000 merging operations.

Table 2 summarizes the baseline accuracy of the MT systems on different test sets. We use BLEU (Papineni et al., 2002) score for automatic evaluation of all the systems. It is interesting to note that the baseline SMT systems in general have higher scores for most of the test scenarios compared to the NMT baselines (except for the News test set in the Te→En direction). This essentially indicates that 750k parallel data is not enough to build NMT-based systems with better quality translation compared to corresponding SMT-based systems due to large parameter space of the NMT-based systems. In addition, the absolute BLEU scores achieved by the baseline systems (either NMT or SMT) are quite low, especially in the En→Te direction. We observe that En→Te has much lower BLEU scores compared to Te→En, irrespective of the MT techniques used. This is often the case for morphologically rich, free word order target languages when using automated metrics based on single references.

System	Te→En		En→Te	
	News	WMT	News	WMT
SMT	9.12	8.76	4.99	3.98
NMT	9.13	7.59	4.04	3.26

Table 2: BLEU scores of the baseline systems

4 Improved NMT Models

The baseline experiments in the previous section clearly associate with the fact that NMT models require massive amount of parallel data in order to generalize over the large parameter space of the model (Gu et al., 2018). Researchers have tried different data augmentation techniques (Gulcehre et al., 2015; Cheng et al., 2016) to improve NMT models. Most of the data augmentation techniques try to leverage the use of monolingual data. We adopt the *back-translation* technique proposed by (Sennrich et al., 2015) to improve the quality of the MT system, which has shown notable success in the past. In this direction, we use an iterative data augmentation and filtering strategy to improve translation quality.

4.1 Back-Translation

To improve our models, first, we use back-translation (Sennrich et al., 2015) to increase the use on parallel data. Back-translation uses a reverse translation engine to translate target-side monolingual data and essentially produced the synthetic data to train the system in forward direction. For example, let e_i be an English sentence, and $t'_i = MT_{En \rightarrow Te}(e_i)$ is the translation produced by the $En \rightarrow Te$ MT system. Then the $Te \rightarrow En$ system is trained on $\{t'_i, e_i\}$ data.

We use the monolingual data mentioned in Table 1 to generate the back-translated data. Table 3 summarizes the detail of the synthetic data used to train the NMT systems. Note, after adding synthetic data, we train the ADAM optimizer with 200k steps with a batch size of 4,096.

Corpus	#sentences	#En	#Te
En_{synth}, Te_{mono}	15.4m	11.4	8.6
Te_{synth}, En_{mono}	8.2m	15.7	12.6

Table 3: Synthetic data

4.2 Iterative Data Augmentation

A good quality baseline system (i.e., reverse translation engine) is required to produce good quality synthetic data. The quality of the synthetic data affects the quality of the MT system. Due to the low quality of the baseline systems (cf. Table 2), we plan to improve the quality of the synthetic data iteratively through iterative data augmentation. The detail of our algorithm is given in Algorithm 1. In line 1 and 2 of the algorithm, we build the baseline reverse translation engines ($M^{(0)}$) using only true parallel data (D_{bi}). Line 4 of the algorithm uses the baseline $M^{(0)}_{En\rightarrow Te}$ to produce synthetic parallel data $\langle D'_{Te}, D_{En}\rangle$ which is further used to improve the MT quality in the other direction ($M^{(t)}_{Te\rightarrow En}$) in line 5. Instead of using the baseline engines ($M^{(0)}_{Te\rightarrow En}$), we use the modified $M^{(t-1)}_{Te\rightarrow en}$ engine in line 6 to produce synthetic data $\langle D'_{en}, D_{Te}\rangle$. Finally, in line 7, we improve the $M^{(t)}_{En\rightarrow Te}$ system using the synthetic data produced in line 6. We continue the process until there is no overall gain (average over Δ_{BLEU} in $Te \rightarrow En$ and $Te \rightarrow En$ directions) in BLEU score. This is ensured in line 8 by measuring the change in BLEU score in the dev set between two successive iterations.

```
Algorithm 1 iterativeAugment($D_{En}, D_{Te}, D_{bi}$)
In: Monolingual English corpus $D_{En}$,
Monolingual Telugu corpus $D_{Te}$,
English–Telugu parallel corpus $D_{bi}$
Out: Translation models $M^{(t)}_{Te\rightarrow En}$ and $M^{(t)}_{En\rightarrow Te}$
 1: $M^{(0)}_{En\rightarrow Te} \leftarrow$ baseline En-to-Te NMT system using $D_{bi}$
 2: $M^{(0)}_{Te\rightarrow En} \leftarrow$ baseline Te-to-En NMT system using $D_{bi}$
 3: for $t := 1$ to $T$ do
 4:     $D'_{Te} \leftarrow$ Translate $D_{En}$ to Telugu using $M^{(t-1)}_{En\rightarrow Te}$
 5:     $M^{(t)}_{Te\rightarrow En} \leftarrow D_{bi}$ + $\{D'_{Te}$,$D_{En}\}$
 6:     $D'_{En} \leftarrow$ Translate $D_{Te}$ to English using $M^{(t-1)}_{Te\rightarrow En}$
 7:     $M^{(t)}_{En\rightarrow Te} \leftarrow D_{bi}$ + $\{D'_{En}$,$D_{Te}\}$
 8:     if $\frac{1}{2}(\Delta_{BLEU}$(dev,$M^{(t)}_{Te\rightarrow En}$)+$\Delta_{BLEU}$(dev,$M^{(t)}_{En\rightarrow Te}$))
        $\leq 0$ then
 9:         return $M^{(t-1)}_{Te\rightarrow En}$, $M^{(t-1)}_{En\rightarrow Te}$
10:     end if
11: end for
```

4.3 Data Filtering

Although the quality of the synthetic data improves through the iterative process in the Algorithm 1, we found that the back-translation quality varies widely across sentences. Thus, we filter poor quality back-translated sentences using a pseudo fuzzy match (PFS) score (He et al., 2010) to rank all the back-translated output. For example, in line 6, once the synthetic parallel data (e.g., $\langle D'_{en}, D_{te}\rangle$) is produced using reverse translation engine (e.g., $M^{(t)}_{Te\rightarrow En}$), we further translate the back-translated D'_{en} into Telugu (D''_{te}) using forward translation engine $M^{(t)}_{En\rightarrow Te}$. We measure the PFS between t ($\in D_{te}$) and t'' ($\in D''_{te}$) as shown in Equation 1.

$$PFS = 1 - \frac{EditDistance(t, t'')}{max(|t|, t''|)} \qquad (1)$$

This essentially helps ranking each pair in the synthetic parallel data with higher scores corresponding to better translation quality.

5 Experiments and Results

First, we conducted one experiment to see the effect of choosing SMT and NMT system as the reverse translation engine to produce back-translated data (line 1 and 2 in Algorithm 1). Note that our baseline SMT system has better quality compared to the baseline NMT system (cf. Table 2). However, we found that the use of NMT as the reverse translation engine has better improvement in translation quality compared to using SMT system for back-translation. Table 4 shows the effect of SMT and NMT system as reverse translation engine. In this process we rely on the baseline $M^{(0)}$ (as shown in line 1 and 2 of the Algorithm 1) and do not use any iterative augmentation of data.

System	Te→En		En→Te	
	News	WMT	News	WMT
SMT	12.78	12.26	5.29	4.14
NMT	14.21	13.26	5.71	4.55

Table 4: Effect of MT system type on back translation. NMT achieves higher quality gains compared to SMT.

The accuracies in Table 4 show that the use of synthetic parallel data significantly improves the baseline translation quality (cf. Table 2). The use on SMT as back-translation system gives an average improvement of 3.58 and 4.16 absolute BLEU points for Te→En system over the baseline SMT and NMT system, respectively. Similar observations are found in En→Te directions with 0.23 and 1.07 absolute BLEU point improvement over the baseline SMT and NMT system, respectively.

Furthermore, we found an absolute average BLEU score improvement of 1.22 and 0.41 using NMT for generating back-translated data compared to the SMT reverse translation system, respectively for Te→En and En→Te systems.

We conduct a second experiment based on the iterative data augmentation technique described in Algorithm 1. We shall refer this as **IDA**. Here we do not filter any data based on PFS value (i.e $PFS \geq 0$). Figures 1 and 2 shows the effect

PFS	#data	Te → En	#data	En → Te
≥0	8.2m	15.05	15.4m	6.49
≥0.3	7.3m	15.14	9.8m	6.66
≥0.5	6.3m	**15.22**	6.7m	**6.77**
≥0.7	4.1m	15.20	3.1m	6.57

Table 5: The effect of PFS on News test set

of IDA over the baselines and non-iterative data augmentation (NMTBT) on different test sets for En→Te and Te→En. We found that the algorithm has no improvement after 2nd iteration in both the directions.

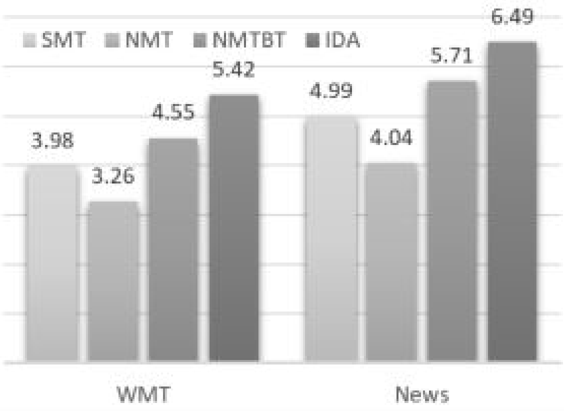


Figure 1: Comparison of BLEU scores for $En \rightarrow Te$. SMT and NMT are baseline systems, NMTBT refers to NMT system with baseline synthetic data.

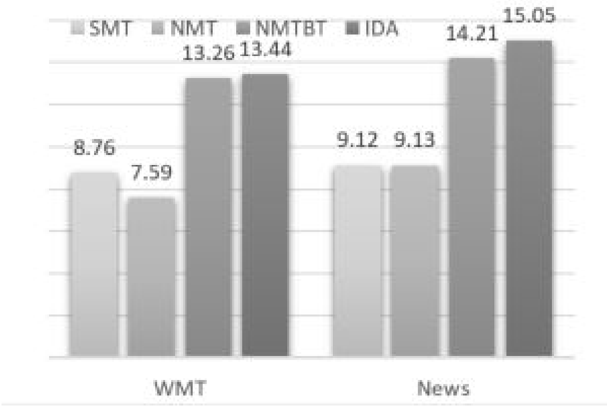


Figure 2: Comparison of BLEU scores for $Te \rightarrow En$

Finally, in our last experiment we show the effect of different PFS threshold for data filtering and their effective impact on BLEU score. Table 5 shows the effect of data filtering using PFS on the News test set. We found that the filtering of data generally improves the translation quality. The best accuracy is achieved when the synthetic data is selected with $PFS \geq 0.5$.

System	Te → En	En → Te
SMT	27.9	35.3
$\text{NMT}_{IDA,PFS\geq0.5}$	56.7	49.6

Table 6: Human evaluation scores on News test set. Based on source-based Direct Assessment. Differences are statistically significant according to Wilcoxon rank sum test with p-level $p \leq 0.05$. Human perceived quality indicates that the NMT system may be good enough for actual general domain use.

5.1 Human Evaluation

In addition to the above automatic evaluations, we performed a manual evaluation of the MT output for both language directions to understand the translation quality from a human perspective. Human evaluation for this research is based on direct assessment. We follow WMT17 (Bojar et al., 2017) and use Appraise (Federmann, 2012), modified to show source sentences instead of reference translations. This adopts the evaluation strategy implemented for IWSLT17 (Cettolo et al., 2017).

For each language direction, five independent annotators evaluated 350 candidate translations on the News test set, randomly drawn from both the baseline SMT (cf. Table 4) and the final NMT system (using IDA and PFS ≥ 0.5). Following direct assessment as implemented at IWSLT17, annotators see the source text and a corresponding candidate translation and are asked to assign a quality score $x \in \{0, 100\}$.

After filtering out annotations used for quality control, we collected an average number of 402 segment scores for SMT, and 399 for NMT. Table 6 shows the average absolute translation quality of the two approaches in both directions. The human evaluation shows statistically significant improvement of 103% and 41% in the absolute scale for Te→En and En→Te NMT systems, respectively, compared to the SMT baseline. We use Wilcoxon rank sum test (Wilcoxon, 1945) with p-level $p \leq 0.05$ to determine statistical significance. All collected data points will be released publicly.

6 Conclusion

We have demonstrated that we can build good quality NMT models with limited resources for a morphologically rich language pair. Contributions of this paper are the definition of iterative data augmentation (IDA) and empirical results showing the effectiveness of back translation and PFS-based data filtering for English–Telugu NMT. The proposed IDA method is much more effective than using baseline back translation by itself.

References

Bach, N., Gao, Q., and Vogel, S. (2009). Source-side dependency tree reordering models with subtree movements and constraints. *Proceedings of the MTSummit-XII, Ottawa, Canada, August. International Association for Machine Translation.*

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proc. of the 2nd Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. (2009). Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. (2017). Overview of the iwslt 2017 evaluation campaign. In *Proc. of IWSLT*, Tokyo, Japan.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596.*

Devlin, J. (2017). Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. *arXiv preprint arXiv:1705.01991.*

Federmann, C. (2012). Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Gu, J., Hassan, H., Devlin, J., and Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368.*

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535.*

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010). Integrating n-best smt outputs into a tm system. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 374–382. Association for Computational Linguistics.

Irvine, A. and Callison-Burch, C. (2013). Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Kinga, D. and Adam, J. B. (2015). A method for stochastic optimization. In *International Conference on Learning Representations (ICLR).*

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54. Association for Computational Linguistics.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the ACL*, pages 160–167. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proc. of the 43rd Annual Meeting of the ACL*, pages 271–279. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709.*

Tiedemann, J. and Nygaard, L. (2004). The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC.*

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.

Toward leveraging Gherkin Controlled Natural Language and Machine Translation for Global Product Information Development

Morgan O'Brien
McAfee, Building 2000 Citygate, Mahon, Cork
mobrien@mcafee.com

Abstract

Machine Translation (MT) already plays an important part in software development process at McAfee where the technology can be leveraged to provide early builds for localization and internationalization testing teams.

Behavior Driven Development (BDD) has been growing in usage as a development methodology in McAfee. Within BDD, the Gherkin Controlled Natural Language (CNL) is a syntax and common terminology set that is used to describe the software or business process in a User Story.

Given there exists this control on the language to describe User Stories for software features using Gherkin, we seek to use Machine Translation to Globalize it at high accuracy and without Post-Editing and reuse it as Product Information. This enables global product information development to happen as part of the Software Development Life Cycle (SDLC) and at low cost.

1 Credits

This document is based on the understanding that commercial Machine Translation systems perform well when used in conjunctions with Controlled Language rules (Roturier, 2004). It uses Gherkin CNL as written by developers and testers of McAfee products. The Machine Translation system used are from the Microsoft Translator Hub. The paper takes input from Information Development teams in McAfee based on the style and standards that they have in place.

2 Introduction

BDD is fast becoming a standard in software development, especially where the User Interface is primarily web based. It aims to satisfy needs of customers in software design by representing the behavior of the user as part of the plan. Using the Gherkin CNL, which is designed to work with BDD frameworks, a business manager who is not a developer can quickly describe how the software should function using examples.

Example based learning has advantages for understanding and retention of information. It enhances the effectiveness of User Stories in Agile software development by reducing ambiguity and enabling non-technical personnel to be involved. It also enhances the ability for a person to retain the information better by the application of a cognitive load to the user as they read. Gherkin could possibly be used as a superior learning asset to traditional information development for complex software processes by virtue that the reader employs more mental effort as they read which helps them retain the information better.

In McAfee, software design is managed through JIRA, a tool for planning, tracking and managing Agile software development. Gherkin descriptions are not currently part of that system and are rarely shared outside of the software Development and Testing teams. By accessing the test code, we get access to the Gherkin which in turn can facilitate the future possibilities such as Information Development. In turn, the ability to quickly leverage this content for international markets at low cost is an attractive possibility for the business.

In this paper, we explore how Gherkin can be organized and stored in JIRA. We test a baseline on how effective Gherkin is with Product Based Machine Translation engines. We then optimize the Gherkin as a better information asset, and in turn optimize the MT to ensure accuracy of message by training glossaries of product and Gherkin terminology.

3 Process

Here we will explain a little about Gherkin and how it will be stored for access. Then we process it through our Product MT engines which are trained on the latest User Interface (UI) translations before running two types of tests on the outputs. We then modify the Gherkin source and re-run the process to test for improvement.

3.1 Gherkin

Gherkin is a ubiquitous language designed to be simple and effective at explaining behaviors carried out on software. Behaviors refer to things the user will do in the Graphical User Interface (GUI). The Gherkin CNL is based on the following concepts:

- A Feature
- A Scenario
- A Background
- A Scenario outline
- The Steps (Given, When, Then, And)
- Examples

There is a specific set of steps to be used for a Gherkin feature or scenario using the “Given”, “When”, “Then” declarations:

- **Given** I experience a specific state
- **And** I experience another starting state
- **When** I do something
- **And** I do something else
- **Then** I will experience an outcome

To reuse the descriptions in Gherkin more effectively, the use of Data Tables is popular. Data Tables allow the test case to be run with a set of variables in the input. The Data Tables we used in our testing are indicative of typical software development content:

- User Interface text
- Usernames and Passwords
- Server Names and Descriptions
- Currency, Numbers and Amounts

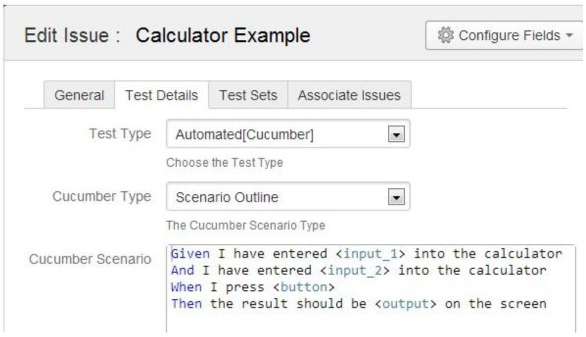
Gherkin authoring standards don’t exist in a structured way within the company to date, but a minimal approach is taken; adhering to the syntax and GUI accuracy. There are 2 types of Gherkin that can be used for different purposes; Imperative and Declarative. Imperative is a detailed description of the behavior expected which has enough specifics to allow test automation code to be written for it, while Declarative is a less detailed higher-level description of the business goals of the software design without thought about the specifics or test code.

Consistency and reusability is of great importance in Gherkin authoring and management to reduce the amount of scenario writing needed.

3.2 JIRA and XRAY

JIRA is a software development tool used by Agile teams. It is designed to streamline the process of Issue and Feature creation and allow global teams to collaborate in their software release process. XRAY is a plugin for JIRA that focuses on the test process by managing the test cases and reporting on their validation in an easy to use dashboard. XRAY allows for the support of Gherkin language in JIRA in multiple ways.

1. Gherkin language is highlighted for known keyword declarations and Data Tables.
2. Gherkin is managed and exportable via a manual or API process in an XML format.
3. The automation code that is bound to the Gherkin test cases can report back on validation again via an API or an importable XML file.



Edit Issue : Calculator Example

Test Type: Automated[Cucumber]
Choose the Test Type

Cucumber Type: Scenario Outline
The Cucumber Scenario Type

Cucumber Scenario:

```
Given I have entered <input_1> into the calculator
And I have entered <input_2> into the calculator
When I press <button>
Then the result should be <output> on the screen
```

Fig 1. Gherkin scenario in XRAY for JIRA

The exportable XML format is key as this offers the opportunity to manage the Gherkin scenario and process it within a localization workflow.

3.3 Translation Quality Tests

To evaluate the success of Machine Translation applied to Gherkin we envisaged tests that are in line with how we currently rate non-Post-Editing translation jobs using MT. The languages we have chosen for this test are Italian, French and Brazilian Portuguese due to the availability of resources to help with the testing. There are 5 complete Gherkin scenarios used in each of the tests, making it 10 scenarios used in all (before and after). It is important to have different scenarios for the before (baseline) and after (future) as familiarity with the process can affect the ability to understand the content during the second set of tests. We chose these tests as they are not exhaustive and provide a quick and useful baseline before pursuing further testing with larger datasets and project participants.

Usability Feedback - from a linguist familiar with products and terminology. The task is to rate the usability and fluency of the sentence in terms of how it can be understood. Usability Feedback is rated from 1 to 5 where 5 is highest quality and 1 is lowest (unusable) quality. We will test this on the source language (English) as well as Brazilian Portuguese, Italian and French. The number generated per language is then the total score divided by the number of strings reviewed (92 Strings were used in the tests).

Cognitive Usage - from an Engineer unfamiliar with the specific product usage but competent in general enterprise software usage. The Cognitive Usability study tests the participant's ability to complete a task with no prior knowledge of the software and is measured on how long it takes to complete the task in minutes. It is measured against the time needed to perform the task using English source. For example, if the task takes 5 minutes in English and 5 minutes in the target language, then the ratio is 1:1 (Same time needed).

Term Type	Example Content
Gherkin	When
Mfe Term	TIE Server
Action	receives a new
Mfe Term	MWG report
none	for the file with
Mfe Term	"Known malicious"
Mfe Term	MWG reputation
none	in
Object/UI	TIE Reputations

Table 1. Term identification - Gherkin segment

3.4 Gherkin Information MT Optimization

Initially our baseline MT systems are not optimized for Gherkin and ultimately the goal is to create an optimized engine. While fluency is sometimes important to understanding, the main goal of Gherkin is quick "In-Process" information development which is cheap to globalize. We focus then on the Keywords and lesser so on the fluency.

The main Action Keywords observed during this test are: Login, Go, Search, Click, See, Set, Accept, Wait, Open, Request, Have, Run, Receive, Request, Reject, Discard.

Gherkin as an information asset must speak in the imperative to direct the user actions. The Gherkin Keywords must be removed to make this possible. This can be done through simple regular expressions (RegEx) on the patterns. This then transforms Gherkin from a User Story description into an instructional asset:

Gherkin as User Story	**Gherkin as Instruction**
When I override the file reputation to "Known Malicious"	Override the file reputation to "Known Malicious"
And I go to "Overrides" tab in the "TIE Reputations" section in ePO	Go to "Overrides" tab in the "TIE Reputations" section in ePO
And I search for the file in the table	Search for the file in the table
And I click on the file in the table	Click on the file in the table

Table 2. Transform Gherkin to an instruction

The process then to move Gherkin from a test asset to an information asset for Machine Translation is like this:

1. Train MT engine with Product Terms
2. Export Gherkin in XML
3. RegEx replaces to the Imperative
4. Machine Translate
5. Publish

3.5 Test Results (before and after)

3.4.1 Usability Study

The Usability Study showed improvements on the understanding and language accuracy for most languages. However, there was a slight drop in accuracy on Italian after optimization was completed.

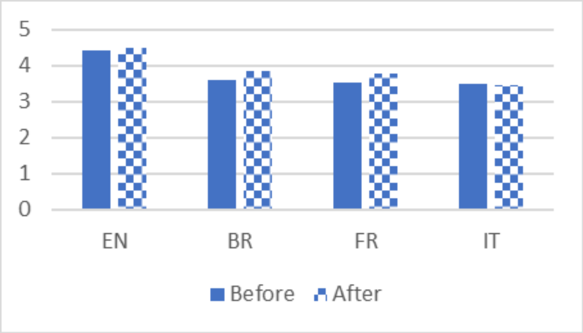


Fig 2. Usability Study averages before/after.

3.4.2 Cognitive Usability

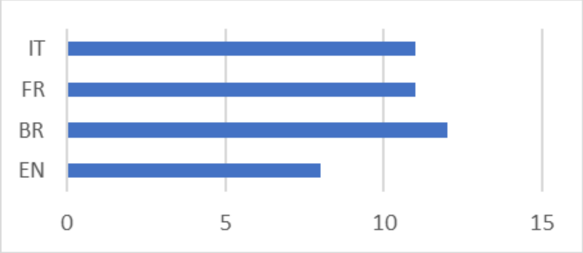


Fig 3. Time to complete tasks baseline.

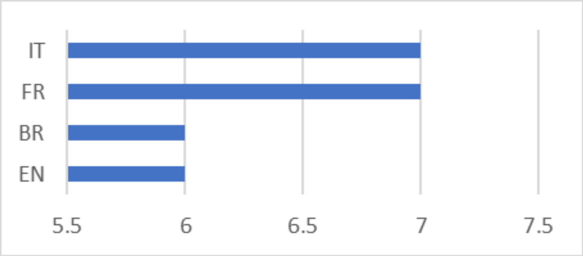


Fig 4. Time to complete after improvements

The Cognitive Study showed improvements across all languages when compared against the source (English) baseline. Before optimization it took between 1.37 to 1.5 times as long to perform the task when compared to following the English source. After optimization this was reduced to 1 to 1.17 times the times, showing that optimization has improved the ability for the user to perform the task in the target languages.

4 Discussion and conclusions

We proposed a method to further leverage an asset currently in use for software development by leveraging Machine Translation and NLP tools such as Regular Expressions. This was done by pre-processing content and optimizing MT engines quickly with one optimization training specifically focused on compliance to terminology. The result is promising showing improvements in many areas based on the analysis of the MT output, and in some cases, is fit for purpose for publishing directly to a customer. In cases where we see drops in usability, the issues stem from the quality of the MT and training/tuning sets. In Italian the UI did not translate well even though the same bitext training content was used in training of all languages. We are confident that training more iterations of the engine to address some of the issues directly would prove useful as issues were predominantly terminology based and may require more weight in the training corpora and or an adjustment of the tuning set.

5 Next Steps

We plan now to expand this test to other languages and improve the current trained MT engines further. In addition, we aim to work with the software development teams to apply more uniform standards to the authoring of Gherkin scenarios and how it is managed as an asset.

Acknowledgements

Acknowledgement is made to the assistance given from Mariana Rolin, Maddalena Benedetto, Sandro Castelletti, Rouslan Placella and Hernan Rey Corvalan from McAfee LLC.

References

Adzic, G. 2009. *Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing*. Neuri, London.

Beck, K. 2002. *Test Driven Development: By Example*. Addison-Wesley, Boston.

Chelimsky, D., Astels, D., Dennis, Z., Hellesoy, A., Helmkamp, B., North, D. 2010. *The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends*. Pragmatic Programmer, New York.

Roturier, J. 2015. *Assessing a set of Controlled Language rules: Can they improve the performance of commercial Machine Translation systems?* Centre for Translation and Textual Studies, Dublin City University.

Sern, L., Salleh, K., Sulaiman, N., Mohamad, M., Yunos, JBM. 2015. *Comparison of Example-based Learning and Problem-based Learning in Engineering Domain*, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.

SpecFlow. 2010. *Pragmatic BDD for .NET* http://specflow.org.

Yagel, R., Sarig, O. 2011. *Can executable specifications close the gap between software requirements and implementation?* Proceedings of SKY 2011 International Workshop on Software Engineering, SciTePress, France.

Implementing a neural machine translation engine for mobile devices: the Lingvanex use case

Zuzanna Parcheta[1] **Germán Sanchis-Trilles**[1] **Aliaksei Rudak**[2] **Siarhei Bratchenia**[2]
[1]Sciling S.L., Carrer del Riu 321,Pinedo, 46012, Spain
[2]Nordicwise LLC, 1 Apriliou, 52, Athienou, Larnaca, 7600, Cyprus
{zparcheta, gsanchis}@sciling.com
{alrudak, s.bratchenya}@lingvanex.com

Abstract

In this paper, we present the challenge entailed by implementing a mobile version of a neural machine translation system, where the goal is to maximise translation quality while minimising model size. We explain the whole process of implementing the translation engine on an English–Spanish example and we describe all the difficulties found and the solutions implemented. The main techniques used in this work are data selection by means of Infrequent n-gram Recovery, appending a special word at the end of each sentence, and generating additional samples without the final punctuation marks. The last two techniques were devised with the purpose of achieving a translation model that generates sentences without the final full stop, or other punctuation marks. Also, in this work, the Infrequent n-gram Recovery was used for the first time to create a new corpus, and not enlarge the in-domain dataset. Finally, we get a small size model with quality good enough to serve for daily use.

1 Introduction

Lingvanex[1] is a trademark for linguistic products made by Nordicwise LLC company. The main focus of the company are translator and dictionary applications that work without internet connection on mobile and desktop platforms.

In collaboration with Sciling[2], an agency specialised in providing end-to-end machine learning solutions, a small-sized translation model from English to Spanish was implemented.

When implementing a mobile translator, it is crucial to understand its purpose. In our case, the purpose was to be able to generate translations on a daily usage scenario, without requiring a Internet connection. This is the typical use case in a travel scenario, where travellers often do not have an internet connection, either because they do not want to assume the cost of a roaming connection, because they do not want to purchase a local SIM card, or even because there is no good connection in the places they are travelling to, such as some countries of Africa. In this scenario, the main purpose of the translation engine is to be able to translate correctly short sentences, composed of common words in a traveller domain, but where other words belonging to e.g. a parliamentary or a medical domain are less frequent. In addition, the model requires to be contained in terms of size, since large models would perform poorly in a mobile device.

In this work, we focus on reducing model size mainly through data selection techniques, until a size of 150MB per model. However, there are other techniques which bring promising results as compressing the NMT model via pruning (See et al., 2016).

Along this article we determine what is the main influence to model size. For that, we conducted experiments comparing model size with total vocabulary size and word embedding size. Also, we compare the model size with different layer number on encoder and decoder

[1] https://lingvanex.com/en/

[2] https://sciling.com/

side, and the size of recurrent layer. Next step is to select data for training the engines through sentence length filtering and leveraging a DS technique. During the implementation of our translation engines we found several problems in the translations generated. We describe each of the problems and we propose appropriate solutions. After implementing these solutions, we evaluate the quality of our final model on a test set, and compare the results with Google's and Microsoft's mobile translators.

2 Data description

The data used to train the translation model was obtained from the OPUS[3] corpus. In total, there were 76M parallel sentences. We also leveraged the Tatoeba corpus for DS described in Section 4. Tatoeba is a free collaborative online database of example sentences geared towards foreign language learners. The development set was also built from the Tatoeba corpus, by selecting 2k random sentence pairs. Main figures of Tatoeba corpus are shown in Table 1. As the test set we create small corpus of more useful sentences in English found in different websites. Also we add some sentences of unigrams and bigrams. In total we selected 86 sentences.

Table 1: Tatoeba main figures. k denotes thousands of elements, $|S|$ stands for number of sentences, $|W|$ for number of running words, and $|V|$ for vocabulary size.

language	$\|S\|$	$\|W\|$	$\|V\|$
English	136k	964k	40k
Spanish	136k	931k	61k

3 Model size dependency

When confronting the model size reduction, the first question that arises is what hyper-parameters have the most influence on model size. Before moving forward and implementing a NMT system, we conducted experiments comparing model size with total vocabulary size and word embedding size (Mikolov et al., 2013). We also compared model size with different number of layers and units per recurrent layer, both on encoder and decoder sides.

To determine how the previously enumerated hyper-parameters affect model size, we trained different models varying these hyper-parameters. In the first experiment, we set the number of units in the recurrent layer to 128, with a single layer on both encoder and decoder sides. We analysed the effect of considering a total combined (source and target) vocabulary size $|V|$ was pruned to $|V| = \{5\text{k}, 10\text{k}, 20\text{k}, 50\text{k}, 100\text{k}\}$, selected according to the most frequent words in the Opus corpus, with souce and target vocabulary size set to $|V|/2$. In addition, we also studied different embedding vector sizes $|\omega| = \{64, 128, 256, 512\}$. The results obtained are shown in Figure 1a.

Next, we analysed the effect of considering different number of hidden units and the number of layers. In this case, we fixed to $|\omega| = 128$. We found that the number of layers, using 128 hidden units, has almost no effect on model size. In Figure 1b, we only show 1 and 4 layers for 128 units. Looking at Figure 1, we can conclude that the number of layers has small effect on model size comparing with number of hidden units and embedding size. Figure 1 can be leveraged to decide on adequate values for these hyper-parameters, once model size has been fixed to 150MB.

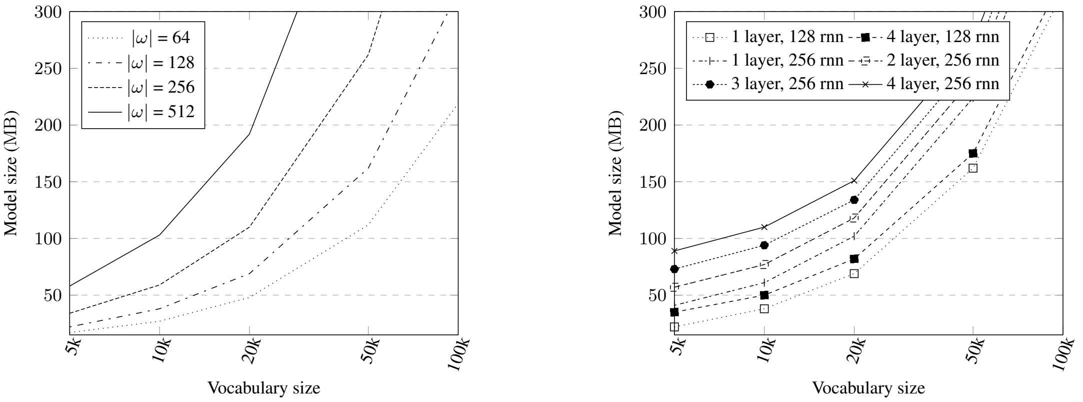
4 Data filtering

Data filtering involved two main steps: first, sentences with length over 20 words were discarded. We did this under the assumption that a mobile translator is mainly designed for translating short sentences. Second, we performed data selection, leveraging Infrequent n-gram Recovery (Gascó et al., 2012). The intuition behind this technique is to select, from the full available bilingual data, those sentences that maximise the coverage of n-grams in a small, domain-specific dataset. The full available bilingual data is sorted by infrequency score of each sentence in order to select first the most informative.

Let be $\mathcal{X}$ the set of n-grams that appear in the sentences to be translated and $\mathbf{w}$ one of them; $C(\mathbf{w})$ denotes the counts of $\mathbf{w}$ in the source language training set; t the threshold of counts when an n-gram is considered infrequent, and $N(\mathbf{w})$ the counts of $\mathbf{w}$ in the source sentence $\mathbf{f}$ to be scored. The infrequency score of $\mathbf{f}$ is:

$$i(\mathbf{f}) = \sum_{\mathbf{w} \in \mathcal{X}} \min(1, N(\mathbf{w})) \max(0, t - C(\mathbf{w})) \quad (1)$$

[3] http://opus.nlpl.eu/



(a) Model size depending of vocabulary size and embedding size. Number of units in the recurrent layer set to 128, and the number of layers is 1.

(b) Model size depending of vocabulary size, number of units in the recurrent layer (*rnn*) and number of layers, with $|\omega| = 128$.

Figure 1: Model size dependency of different parameters. k denotes thousands of elements and MB is an abbreviation for megabyte. The vocabulary size is the sum of source and target vocabulary.

We applied Infrequent n-gram Recovery to the 60M sentences from the Opus corpus as out-of-domain. Intuitively, we selected sentences from the available data until all n-grams, with n up to 5, extracted from the Tatoeba corpus have a maximum of 30 occurrences (if such a thing is possible with the data available). However, applying this technique on the full set of 60M sentences would have led to very long execution time. Hence, we divided the corpus into 6 partitions, and the selection was performed on each one of these partitions. Then, we joined the selections from all 6 partitions and conducted a second selection step on this corpus, since some n-grams could well have $6 \cdot 30$ occurrences. This led to a final selection of 740k sentences. The selected data set presented a vocabulary size of 19.4k words in source and 22.9k on target side. The total (combined) vocabulary was $|V| = 42.4$k. Note that selection was conducted on the tokenised and lowercased corpus.

5 Experimental setup

The system was trained using the OpenNMT (Klein et al., 2017) deep learning framework based in Torch. OpenNMT is mainly focused at developing sequence-to-sequence models covering a variety of tasks such as machine translation, summarisation, image to text, and speech recognition. Byte-pair encoding (BPE) (Sennrich et al., 2015) was trained on the selected training dataset, and then applied to training, development, and test data. In each experiment we trained a recurrent neural network with long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997). We use global attention layer to improve translation by selectively focusing on parts of the source sentence during translation. Also, we use input feeding to feed attentional vectors as inputs to the next time steps to inform the model about past alignment decisions (Luong et al., 2015). However, this option only had a visible effect with 4 or more layers. Training was performed with 50 epochs using the *adam* (Kingma and Ba, 2014) optimiser, with learning rate of 0.0002. Finally, we selected the best model according to higher BLEU (Papineni et al., 2002) score on the development set, and used that model to translate the test set. Given that the test set is very small, we performed a human evaluation to analyse whether the quality obtained was good enough.

6 Results and analysis

We trained different typologies of neural networks observing the conclusions in Section 3. In each of the experiments we varied the hyper-parameters described in Section 3. Since the total combined vocabulary was fixed to $|V| = 42.4$k, from Figure 1 we can infer the combination of hyper-parameters with which the allowed model size will not be exceeded.

Table 2 shows the values of the hyper-parameters used in each experiment, together with the BLEU score obtained by each model and its size.

The best model according to the BLEU score on the development set is the model trained with 2

Table 2: Hyper-parameter values for the different experiments (exp) conducted and results obtained. $|\omega|$ is the size of the word embedding vector, expressed in megabytes.

exp	$\lvert\omega\rvert$	layers	rnn	size	BLEU	
					dev	test
1	128	2	128	146	39.0	26.6
2	128	3	128	151	36.9	22.8
3	128	4	128	155	37.7	21.8
4	64	4	256	206	38.7	23.8
5	256	4	64	203	32.7	21.1

layers, 128 units on recurrent layer, with $|\omega| = 128$. Also, it is the smallest model among those analysed in Table 2.

6.1 Problems found

Analysing the translations from the test set we found 3 different problems. In the following, we describe each of them and propose the corresponding solutions.

6.1.1 Repeated words problem

Analysing the quality of the best model obtained, we noticed that sentences with more than 7 words were translated correctly. However, translations of very short sentences contained repeated words, e.g. *"perro perro perro perro perro perro"*. The hypothesis for explaining this fact could be because of differences between training and test sentence lengths. To understand the validity of this hypothesis, we analysed the histogram of sentence lengths of training set, shown in Figure 2. As seen, the source side of the training data contains a very few amount of sentences shorter than 8 words, in contrast to the target side, where the distribution of sentence length is more uniform. We believe such difference is caused by the sentence selection algorithm used: selection is conducted in the source language and the selection algorithm tends to assign higher scores to longer sentences, since the more n-grams the source sentence contains, the more likely it includes infrequent n-grams. To cope with this fact, we modified the Infrequent n-gram Recovery strategy as follows:

Re-scoring of sentences: To fix the problem of repeated words we decided to modify the sentence selection procedure modifying the Infrequent n-gram Recovery scoring function by adding a normalisation step. In order to normalise such score, we modified Equation 1 as follows:

$$i(\mathbf{f}) = \sum_{\mathbf{w}\in X} \frac{\min(1, N(\mathbf{w}))\max(0, t - C(\mathbf{w}))}{|\mathbf{f}| - \mathbf{w} + 1} \tag{2}$$

where the denominator normalises by the number of n-grams of order n in the sentence. With this normalisation, we avoid the side-effect of sentence length on the infrequency score, ultimately leading to selecting shorter sentences and improving the NMT system's translation of such sentences.

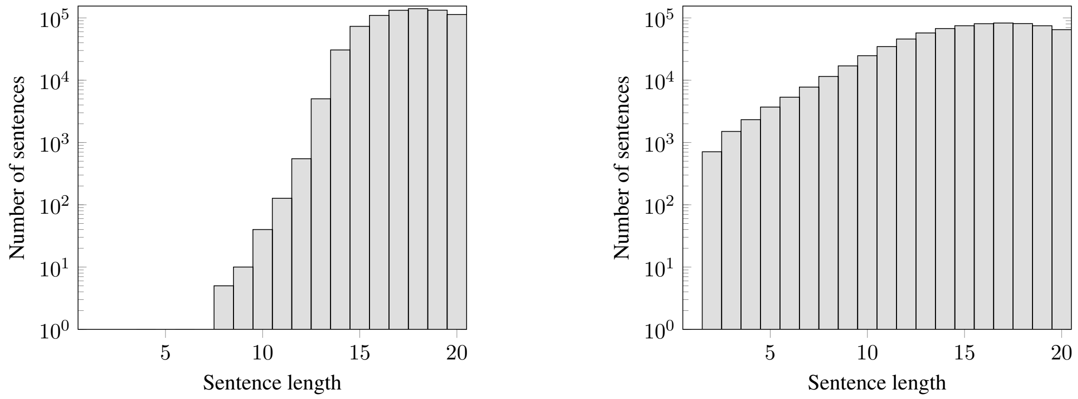
After applying the infrequency score in Equation 2 for selecting the data anew, we obtained 667k sentences. In Table 3 we show the average sentence length in source and target language before and after applying the sentence length normalisation. Average sentence length of Tatoeba is shown for comparison purpose. As shown, we are able to obtain much shorter sentences by including normalisation. The model achieved 36.3 BLEU in development, and 22.8 in test, with a model size of 121MB. Although this score is slightly worse than the one achieved in experiment 1 (Table 2), we believe BLEU is not always the most adequate metric for evaluating translation quality (Shterionov et al., 2017). By manually analysing the hypotheses, we concluded that the repeated words problem had been successfully solved.

Table 3: Average sentence length of Tatoeba and training set before and after applying normalisation in Equation 2.

		source	target
Tatoeba		7.1	6.8
train	before normalisation	17.4	15.1
	after normalisation	10.4	9.0

6.1.2 Punctuation mark expectation

Analysing the hypotheses generated by our new model, we noticed that the model generated wrong translations with very short sentences, e.g. "dog", or "cat", generating surprising translations such as "amor". However, when adding a punctuation mark to the source sentence, e.g. "dog.", the translation was correctly produced. Our first intuition regarding this was that the model was expecting a punctuation mark at the end of each sentence. This intuition was confirmed by the fact that 94% of the sentences in the source language training set had a dot or other



(a) Histogram of source training set. (English)

(b) Histogram of target training set. (Spanish)

Figure 2: Histogram of training set.

punctuation marks at the end of sentence, and one of the more common final words without a punctuation mark was precisely "amor". Hence, the network was confused (i.e., the model was poorly estimated) when such punctuation mark did not appear. For dealing with this problem, we devised two possible solutions:

Special word ending: We append a special word `@@` at the end of each sentence. Then, the model is forced to learn that a sentence will always finish with `@@`, and the fore-last word might or might not be a punctuation mark. This was applied as a pre- and post-processing steps, and will be referred to as *special word ending*. The model trained using special word ending achieved 36.4 BLEU in development, and 26.3 BLEU in test. This model was reached after 21th epochs and its size was of 121MB.

Double corpus: We enlarged the training corpus by concatenating all existing sentences with punctuation mark at the end, but removing such symbols. By doing so, the model is able to learn that a sentence can finish with or without punctuation marks. This time, the model had a size of 156MB, and reached 37.3 BLEU in development, and 25.1 in test.

Both techniques described previously solved the problem of punctuation mark expectation. However, since the double corpus strategy produced a larger model, with lower BLEU score, we decided to employ the special word ending technique.

6.1.3 Missed segments

Further analysing the translations generated by our model, noticed an additional problem: in case the segment being translated was composed of several short sentences, only the first of them was being actually translated. For instance, "Thank you. That was really helpful." was translated into "Gracias." ("Thank you.").

To solve this problem, we decided to apply a preprocessing step, consisting separating segments composed by several sentences into different segments, according to punctuation marks ".", "?" and "!", in the case of English language, and also in "¿" in case of Spanish language. We split 86 sentences from test set into 118, given that most of them were composed by short sentences. After this preprocessing step was performed, the translations were correctly generated, reaching 36.4 BLEU in development, and 33.7 BLEU, this last one being the highest score so far.

7 Final evaluation

Table 4 summarizes the BLEU scores obtained after applying each one of the solutions described in Section 6. After applying the normalised infrequency score, the special word ending, and preprocessing composed sentences, we improved the quality of test set by about 7 BLEU points.

Table 4: Translation quality, as measured by BLEU, after applying each technique described. Size is given in MB.

technique	size	BLEU	
		dev	test
Base model	146	39	26.6
Re-scoring of sentences	121	36.3	22.8
Special word ending	121	36.4	26.3
Double corpus	156	37.3	25.1
Sentence splitting	121	36.4	**33.7**

As final evaluation of our translation system, we compared its quality with Google's and Microsoft's mobile translators. The BLEU score on the test set obtained by each of the analysed translators, alongside with their corresponding model sizes, are shown in Table 5. In general, all translators generate good quality hypotheses, although some small differences could be observed. We noticed that our model was especially accurate when using punctuation marks and capital letters, whereas Google's translator introduced punctuation marks in wrong places. Also, only in a few cases, Google's translator, uses capital letters. We believe this is the reason why Google's translator achieved such a low BLEU score, as compared to the other two systems. However, Google's translator features a much smaller than the other two others. Also, Google's and Microsoft's models are bidirectional, which means that the size of our model should be doubled (2 · 121MB) to be comparable.

Table 5: Translation quality and model size comparison for Google, Microsoft and our best model.

	Google	Microsoft	our system
BLEU	16.7	28	**33.7**
Model size	29MB	234MB	121MB
Both directions	YES	YES	NO

8 Conclusions

In this work, we have presented our approach to developing a small size mobile neural machine translation engine, in the specific case of English–Spanish. We leveraged a data selection technique to select more suitable data for real use of our translator. We have presented some adjustments to the selection algorithm the translation quality obtained. Also, we proposed a solution to deal with the problem of repeated words, and another one for dealing with missed sentence translations within some segments. Finally, we compared the quality of our model with Google's and Microsoft's mobile translator versions. We overcome significantly the BLEU score of both translators, partially due to being able to translate punctuation marks and capital letters correctly. Our model reached a size of 121MB, which is even much smaller than the size we considered initially as acceptable, presenting good translation quality for the specific purpose (travel domain). The translations obtained by our model are perfectly understandable and fluent, and can be used in a scenario where there is no internet connection. In addition, we are still working on improving its quality and on reducing model size even further, using other effective techniques such as weight pruning.

Acknowledgments

Work partially supported by MINECO under grant DI-15-08169 and by Sciling under its R+D programme.

References

Gascó, G. et al. (2012). Does more data always yield better translations? In *Proc. of EACL*, pages 152–161.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, pages 1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980.

Klein, G. et al. (2017). OpenNMT: Open-source toolkit for neural machine translation. *arXiv preprints*, arXiv:1701.02810.

Luong, M. et al. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprints*, arXiv:1508.04025.

Mikolov, T. et al. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprints*, arXiv:1310.4546.

Papineni, K., , et al. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

See, A. et al. (2016). Compression of neural machine translation models via pruning. *arXiv preprints*, arXiv:1606.09274.

Sennrich, R. et al. (2015). Neural machine translation of rare words with subword units. *arXiv preprint*, arXiv:1508.07909.

Shterionov, D. et al. (2017). Empirical evaluation of NMT and PBSMT quality for large-scale translation production. In *Proc. of EAMT*, pages 75–80.

Bootstrapping Multilingual Intent Models via Machine Translation for Dialog Automation

Nicholas Ruiz, Srinivas Bangalore, John Chen
Interactions, LLC
Murray Hill, NJ, USA
{nruiz,sbangalore,jchen}@interactions.com

Abstract

With the resurgence of chat-based dialog systems in consumer and enterprise applications, there has been much success in developing data-driven and rule-based natural language models to understand human intent. Since these models require large amounts of data and in-domain knowledge, expanding an equivalent service into new markets is disrupted by language barriers that inhibit dialog automation.

This paper presents a user study to evaluate the utility of out-of-the-box machine translation technology to (1) rapidly bootstrap multilingual spoken dialog systems and (2) enable existing human analysts to understand foreign language utterances. We additionally evaluate the utility of machine translation in human assisted environments, where a portion of the traffic is processed by analysts. In English→Spanish experiments, we observe a high potential for dialog automation, as well as the potential for human analysts to process foreign language utterances with high accuracy.

1 Introduction

With the present advances in natural language understanding and speech recognition technologies, unprecedented opportunities have been created for realizing natural and sophisticated human-machine conversations to accomplish routine tasks. There has been a resurgence of speech/text-based conversation systems spanning multiple platforms, such as interactive voice recordings, chat, and SMS, owing to the availability of communication platforms that make it convenient to configure human-machine conversations. The potential opportunities of speech/text driven human-machine systems, or *virtual agents*, can only be realized if the user's requests are *understood* by the virtual agent and acted upon appropriately.

For practical applications, such conversational agents and speech/text analytics systems, the meaning of a sentence may be approximated as one or more actionable labels, or *intents*, associated with the input utterance. In such cases, the natural language understanding (NLU) task is modeled as an *intent classification* problem. Although ambiguity is present in natural language, data-driven NLU systems have been successful in modeling user intents in many application domains.

Many commercial and enterprise applications service customers from different geographic locations and varying language proficiencies, requiring multilingual NLU for human-machine interaction. In order to deploy an intent classification system for a new language a new set of labeled training data is conventionally required. This data is often unavailable before a solution is deployed, instead requiring a human-driven dialog system depending on intent analysts or live agents. In time, a sufficient amount of production data may be collected to build a data-driven intent model; however this approach is expensive to operate and ignores valuable knowledge present in other languages that could be used to build an initial model.

In this paper, we evaluate the use of machine translation (MT) as a tool to bridge the knowledge present in one or more intent models for the creation of an intent classifier in a target language. Given MT's capability to translate the content of

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 303–308
Alacant, Spain, May 2018.

utterances in a source language to a target language, our goal is to minimize the number of language proficient intent analysts needed to support a production-scale multilingual dialog system in the absence of target language data.

The remainder of the paper is organized as follows: in Section 2 we discuss the details of the intent classification system and present the NLU model, including our two MT architectures in a multilingual spoken dialog system. In Section 3 we outline our experiment and describe our ASR, MT, and NLU models. In Sections 4-7, we evaluate the ASR, MT, and NLU model performance. In Section 8 we evaluate human agents' ability to label the intents of translated user utterances and summarize our findings in Section 10.

2 NLU for Customer Care

A single utterance is tagged with three types of labels: *intents*, *entities*, and *conversational handlers*. Intents are domain-specific labels such as SALES, TECH ASSISTANCE, and BILLING. *Entity* labels represent the names of products or services mentioned by the user. These include specific models of smart phones or subscription services. Conversational handlers are labels which are similar to speech acts to guide the conversation. For example, LIVE AGENT, CONFUSED, or FOREIGN LANGUAGE. In our experiments, intents and entities are labeled as "session variables" (SV), while conversation handles are partitioned into "task names" (TN) and "event names" (EN). Examples of each in our experiments are shown in Table 1.

A joint SVM classifier is trained by concatenating the TN, EN, and SV labels into a unique label. The model comprises a set of binary SVM classifiers, with each classifier predicting if the input is assigned or not assigned to a particular label type. For a given input utterance x, the joint label is computed as:

$$y^* = \underset{y \in \langle tn, sv, en \rangle}{\arg\max} \; F_y(x, y). \tag{1}$$

The feature set F may comprise *application context*, *conversation context*, and the *utterance*. We use n-gram word-level features in this experiment.

2.1 Confidence Measures

We boost the accuracy of our intent models by using human analysts' predictions on unconfident decisions made by the classifier. We obtain prediction probabilities from the classifier by computing the sigmoid on the scores output for each label by the SVM classifier, computed as:

$$P(y^*) = \frac{1}{1 + \exp(F_y(x, y^*))}. \tag{2}$$

The confidence measure is obtained by computing the ratio of the probabilities of the first and second best labels assigned by the classifier:

$$cf(y^*) = \frac{P(y^*)}{P(y^{*-1})}. \tag{3}$$

The rejection threshold is empirically determined to maximize the accuracy of the human-assisted solution while minimizing human labeling costs.

2.2 Multilingual Bridging via Machine Translation

In the context of dialog systems, MT can be used in one of two ways: (1) translating real-time target

Task Names (TN)	Session Variables (SV)	Event Names (EN)
COMPLAINT ENGLISH FOREIGN NONE QUESTION ...	ACCOUNT action ACTIVATE product ADD service APPOINTMENT type BILLING AUTO PAY BILLING DETAILS PAY BILL service CHANGE ADDRESS DISCONNECT service ...	ANGRY DON'T KNOW GARBLED LIVE AGENT NOISE NO MATCH NONE THANK YOU ...
10	707	18

Table 1: Examples and counts of Spanish intent labels by category and language for the "How may I help you?" dialog state. Concatenating the labels yields 833 distinct intent annotations.

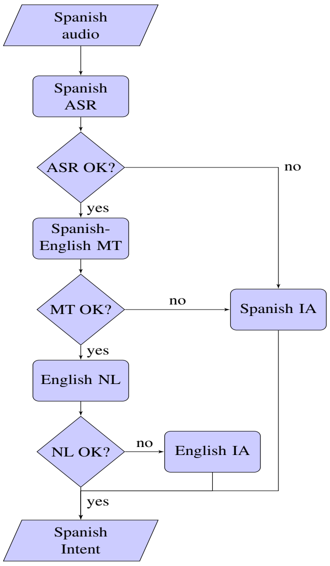


Figure 1: Online Spanish→English bootstrap architecture. Spanish audio is translated into English and processed by an English intent classifier. If ASR or MT scores are not confident, a Spanish intent analyst (IA) labels the segment. If the English intent model is not confident, an English IA labels it.

Data set	# utts	# words	# unique labels
English training	6.5M	40.6M	623
Spanish training	0.8M	4.6M	623
Spanish test (ASR)	1007	4696	178
Spanish test (human)	1007	5153	178

Table 2: Utterance, word, and distinct label counts for the training and test data used in our experiments.

language data into the source language and predicting the intent with a source language intent classification model (cf. Fig. 1); or (2) translating source language data offline into the target language and training a target language intent classification model. In the first scenario, utterances with high translation quality may be processed by an analyst that only speaks the source language, if the NLU confidence score is too low.

3 Experimental setup

We evaluate the efficacy of bootstrapping a Spanish intent classifier using the data and underlying models from an English spoken language dialog system. The data set consists of customer voice responses to the message "How may I help you?" in the customer's native language at the beginning of a phone call to an Interactive Voice Response (IVR) system. We assume that no Spanish intent data is available during training time and evaluate the performance of our bootstrapped Spanish models against an intent model trained with a standard training set. Table 2 lists the data used in our experiment, The training data consists of 5-best ASR hypotheses on audio segments for English and Spanish. We assume that the intent labels covered by the target Spanish model are the same as the English model. Although most of the intent labels overlap one another there is a subset of $\langle tn, sv, en \rangle$ intent triples that do not overlap. We discard the non-training examples with non-overlapping triples from each data set, reducing the number of unique labels in both data sets to 623 (cf. Table 1). As a result, 2.94% of the English training examples and 1.95% of the Spanish training examples are discarded, respectively. Model performance is evaluated on a test set of 1007 Spanish audio segments randomly extracted from production logs and labeled by Spanish-speaking intent analysts (IAs) and verified by a supervisor.

3.1 Automatic Speech Recognition

Our Spanish ASR system consists of an n-gram language model and hybrid DNN acoustic model trained with the cross-entropy criterion followed by the state-level Minimum Bayes Risk (sMBR) objective. We use sequence-training with smoothing and speed-perturb the training data. The acoustic model has general-phone and head-body-tail based digit-specific triphones. The training data consists of 500K training utterances (around 1000 hours of audio) without speech perturbations and about 40K unique vocabulary words.

3.2 Machine Translation

We use a conventional neural machine translation (NMT) sequence-to-sequence encoder-decoder with attention architecture (Bahdanau et al., 2015; Luong and Manning, 2015; Sennrich et al., 2016) commonly used by MT practitioners. The NMT models were trained with parallel English-Spanish data from Europarl v7, CommonCrawl, and WMT News Commentary v8 from the WMT 2013 evaluation campaign (Bojar et al., 2013), as well as the TED talks from IWSLT 2014 (Cettolo et al., 2014). The training data has a shared vocabulary size of 89,500 words after byte-pair encoding (Sennrich et al., 2016). The model is trained for 20 epochs with two bidirectional LSTM encoding and decoding layers with 512 units. In this experiment we assume to have no in-domain parallel data.

For the offline English→Spanish model, we translate the English intent model's training data (6.5 million utterances) into Spanish using our baseline NMT system. The number of words in the translated data set remains roughly the same. The translated outputs are used to train a bootstrapped Spanish intent classifier, using the same training parameters as the native English model. The ASR outputs from the test set are processed by the bootstrapped Spanish intent classifier. For the real-time Spanish→English model, insert punctuation and apply truecasing to the ASR outputs from the test set and translate the outputs with our Spanish→English baseline NMT system. We strip the punctuation and lowercase the machine translated output and pass it through the native English intent classifier.

3.3 Intent classification

Our intent classifiers are trained using an implementation of SVMs in SCIKIT-LEARN[1], using the approach described in Section 2. We evaluate the

[1] http://scikit-learn.org

Translation	BLEU ↑	TER ↓	Length
ASR outputs	42.5	51.2	94.1
ASR outputs (+PE)	48.0	44.6	96.3
Human transcripts	51.8	41.3	111.4

Table 3: Machine translation quality measured in BLEU, TER, and utterance length, evaluated against post-edited translations of human transcripts.

performance of an intent classification model by plotting an *error-rejection curve*, which measures the error rate of the intent classifier as the number of utterances that are processed by the intent analysts increases. For example, a 10% rejection rate corresponds means that only 90% of the test set is evaluated by the model.

We compare the results of each bootstrapped NLU approach with a native Spanish intent model trained on the held-out Spanish training data. We additionally repeat the experiment with the human transcripts to measure the difference in intent classification error that may be explained by ASR. Error-rejection curves for each intent model are shown in Fig. 2 and the scores at 0%, 10%, and 20% rejection are shown in Table 4.

4 ASR performance

We use SCLITE from the NIST Speech Recognition Scoring Toolkit[2] to compute the word error rate (WER) and utterance error rate. After further clean-up and adjudication, we observe a 33.2% WER, with 60.0% of the utterances containing errors (32% substitutions, 22% deletions, 28% insertions). A majority of the substitution errors were confusions between singular and plural (e.g. *problemas→problema*) and articles (e.g. *del→de*). Other errors included phonetic confusions (e.g. *cuenta→fuenta*), named entity misrecognitions (e.g. *HBO→yo*), and a high frequency of dropped articles (e.g. *de*, *la*, *a*) caused by speaker under-articulation. Of these types of errors, the most detrimental are substitutions of named entities, verbs, and nouns that are not lemmatization errors. Another issue driving up the WER score caused by the IVR system truncating audio longer than four seconds to reduce latency.

5 Machine Translation quality

In order to assess the translation quality, we post-edit the translations of the human-transcribed utterances and report the case-insensitive BLEU and TER scores in Table 3. ASR errors increase the required translation edits by 10%, from 41.3% edits to 51.2%. The primary sources of errors are incomplete sentences, lack of punctuation; lexical mistranslations of key words: (e.g. *equipo* (*equipment*)→*team*; *dirección* (*address*)→*direction*; *reclamo* (*complaint*)→ *claim*); ambiguous translations (e.g. *factura→bill/invoice*); and duplicated words during translation (e.g. *payment arrangements→payment payment*). Many of these issues are due to lack of in-domain MT training data and low tolerance of ASR errors. Highly repetitive errors indicate that an automatic post-editing system could substantially improve the translation system's quality. Table 3 also shows that post-edited translations of ASR outputs are substantially worse than those of the human transcripts (41.3% TER difference), showing that ASR errors are exacerbated through translation.

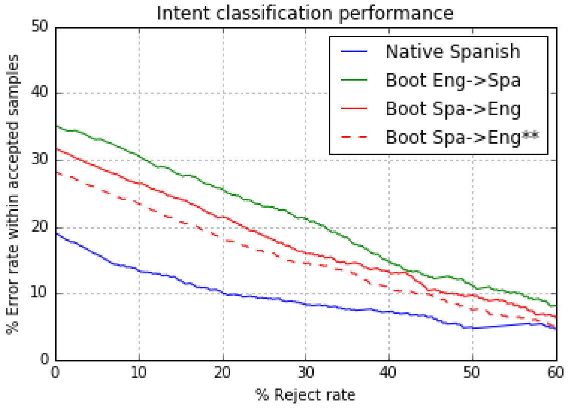


Figure 2: Error-rejection curves for bootstrapped intent classification performance on Spanish ASR outputs, versus a native Spanish model. Results are reported for intent classifier predictions on ASR outputs. Spa→Eng**: performance on post-edited MT outputs.

6 Native NLU performance

Our reference native Spanish intent classification model is trained on ASR outputs since an insufficient amount of human-transcribed intent modeling data is available. From Table 4, we see that at 0% rejection, the native model yields a 19.2% classification error, while the human intent analysts (IAs) yielded a 11.0% error while listening directly to the audio. At the same time, if the IAs are presented with the ASR outputs, they produce an error rate of 24.4%. This demonstrates the intent model's ability to tolerate a certain degree of ASR errors by being trained on ASR errors.

[2] https://www.nist.gov/itl/iad/mig/tools

Configuration	Reject 0%	 10%	 20%
Native Spanish (ASR)	19.2	13.5	10.3
Native Spanish (human)	19.1	12.3	7.4
Boot Eng→Spa (ASR)	35.3	30.7	25.8
Boot Eng→Spa (human)	31.6	26.3	21.4
Boot Spa→Eng* (ASR)	31.8	26.5	21.4
Boot Spa→Eng* (human)	33.4	26.7	20.2
Boot Spa→Eng* (ASR+PE)	28.3	23.6	18.3
Spanish IA (audio)	11.0	10.6	9.5
Spanish IA (human)	14.9	13.1	10.9
Spanish IA (ASR)	24.4	21.6	17.6
English IA (ASR+MT)	33.9	30.6	26.1
English IA (ASR+MT+PE)	25.9	22.8	18.3

Table 4: Intent classification performance by machine learning models and human intent analysts (IAs) at 0%, 10%, and 20% rejection.

Of the 1007 test utterances, 152 utterances have both ASR errors and were classified incorrectly, although they were labeled properly by IAs, comprising 15.1% of the 19.2% intent classification errors. Of the utterances with ASR errors that cause an intent classification error, six were cases where ASR failed to produce a hypothesis, 26 were cases of audio truncation, and 39 utterances containing only ASR substitution errors. The latter cases often caused underspecification during intent classification (e.g. LOWER MY BILL→BILLING PROBLEM; MAKE A PAYMENT→BILLING). In isolation, insertion and deletion ASR errors did not have a significant impact on NLU.

Evaluating NLU performance on human transcripts, we observe improvements that rival human labeling performance at 10% rejection and above. At 20% rejection, the model significantly outperforms human labeling.

7 Bootstrapped NLU performance

While the native Spanish model has a 13.5% error rate at 10% rejection when processing ASR hypotheses, the bootstrapped models have double the error rate due to the use of out-of-domain machine translation. On ASR hypotheses, the English→Spanish model yields an error rate of 30.7%, while the Spanish→English model yields 26.5% at 10% rejection.

To better understand how machine translation further corrupts the bootstrapped Spanish→English performance, we compare the errors it makes to the Native Spanish model. In Table 5, we group the NLU errors by whether they are present in the Native Spanish model, the bootstrapped model, or both. 14.2% of the test set are examples where the Native Spanish model makes a correct prediction, but the Spanish→English model yields errors. By comparing to post-edited ASR+MT data, only 3% of those errors are directly attributed to ASR errors. The 11% of Spanish→English model-specific errors are mostly attributed to intent underspecification. For example, 60% of the ACCOUNT errors are NULL misclassifications. For billing issues, two-thirds of the errors are semantically similar misclassifications, such as PAYMENT, LOWER MY BILL, and BILL DETAILS.

Native	Spa→Eng	ASR	ASR+PE
+	+	66.6%	69.6%
+	-	14.2%	11.2%
-	+	1.5%	2.0%
-	-	17.7%	17.2%

Table 5: Comparison of Native Spanish intent model to bootstrapped Spanish→English models on ASR outputs. +/- indicate that whether the corresponding model's prediction was correct.

8 Analyst performance on translated text

Finally, we measure IA labeling performance on translated utterances. In our conventional scenario, intent analysts listen to audio segments in their native language and provide an intent label. Instead, we replace the original audio with machine translated or translation post-edits of ASR hypotheses. Fig. 3 provides error-rejection curves for IAs, with error rates at 0%, 10% and 20% rejection reported in Table 4.

We first assess the labeling loss when humans annotate ASR outputs in the absence of audio. Although their error rate increases from 11.0% to 24.4% when annotating ASR transcripts, their performance on human transcripts is within 5% of listening directly to the audio at 0% rejection. As the rejection rate increases, the difference becomes negligible. As we introduce Spanish→English machine translation, we observe that the labeling error increases from 24.4% to 33.9% on ASR, which is incidentally worse than the Spanish→English intent classification model's accuracy (31.8%)! However, the IA labeling error rate drops to 25.9% on post-edited MT outputs – only a 1.5% increase in NLU errors caused by translation. These results suggest that with proper ASR and MT adaptation through in-domain data, we could obtain similar English-speaking IA performance on machine translation outputs as the Spanish-speaking IAs on

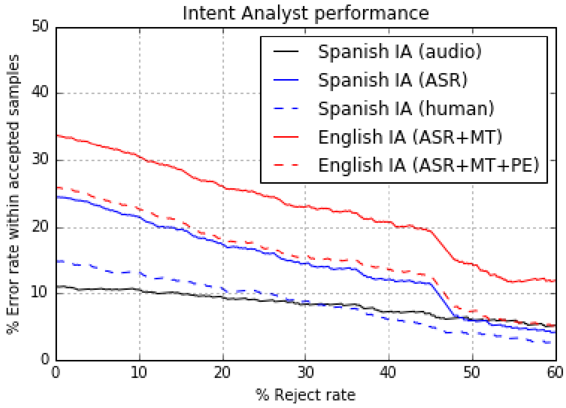


Figure 3: Error-rejection curves for the intent analyst (IA) labeling accuracy on Spanish audio, Spanish ASR and human transcripts, and their machine translations into English (MT). Rejection is computed from the English model's confidence scores on MT outputs. "+PE": performance on post-edited MT outputs.

their native language utterances.

9 Related Work

The use of MT to translate texts in other languages into English for sentiment analysis was proposed in Denecke (2008). Bautin et al. (2008) show that sometimes MT performs inadequate translations on essential parts of a text, affecting sentiment analysis performance. Our results confirm this phenomena due to a lack of in-domain MT training data. Schwenk and Douze (2017) explore learning multilingual sentence embeddings with neural MT, which can aid in multilingual search. Prior to that, multilingual approaches leveraged lexical resources such as MultiWordNet (Pianta et al., 2002) to bridge concepts from one language to another.

10 Conclusions

We have executed an experiment to measure machine translation's ability to rapidly bootstrap intent classification models for new languages. In our English→Spanish experiments, we observe that although the initial results appear to be substantially worse than a Native Spanish intent classification model, we show that MT can provide a degree of automation that supports human-assisted multilingual dialog systems that can be deployed to production on day one, reducing the need for human agent support over a fully manual solution. There is further promise that model improvements can be obtained by improving the ASR and machine translation models to include in-domain data. Finally, we observe it is better to use the online Spanish→English bootstrap in our production system rather than an offline English→Spanish intent model.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *5th International Conference on Learning Representations*, San Diego, USA. ICLR.

Bautin, Mikhail, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In Adar, Eytan, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT Evaluation Campaign. In *Proceedings of the International Workshop on Spoken Language Trnaslation (IWSLT)*, Lake Tahoe, USA, December.

Denecke, K. 2008. Using sentiwordnet for multilingual sentiment analysis. In *2008 IEEE 24th International Conference on Data Engineering Workshop*, pages 507–512, April.

Luong, Minh-Thang and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Pianta, Emanuele, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January.

Schwenk, Holger and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167. Association for Computational Linguistics.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

How to Move to Neural Machine Translation for Enterprise-Scale Programs—An Early Adoption Case Study

Tanja Schmidt
Welocalize, Inc.
Frederick, MD, United States

tanja.schmidt@welocalize.com

Lena Marg
Welocalize, Inc.
Frederick, MD, United States

lena.marg@welocalize.com

Abstract

While Neural Machine Translation (NMT) technology has been around for a few years now in research and development, it is still in its infancy when it comes to customization readiness and experience with implementation on an enterprise scale with Language Service Providers (LSPs). For large, multi-language LSPs, it is therefore not only important to stay up-to-date on latest research on the technology as such, the best use cases, as well as main advantages and disadvantages. Moreover, due to this infancy, the challenges encountered during an early adoption of the technology in an enterprise-scale translation program are of a very practical and concrete nature and range from the quality of the NMT output over availability of language pairs in (customizable) NMT systems to additional translation workflow investments and considerations with regard to involving the supply chain.
In an attempt to outline the above challenges and possible approaches to overcome them, this paper describes the migration of an established enterprise-scale machine translation program of 28 language pairs with post-editing from a Statistical Machine Translation (SMT) setup to NMT.

1 Introduction

The idea of using recurrent neural networks for machine translation was first presented by Kalchbrenner and Blunsom (2013), followed soon after by Cho et al. (2014) and Sutskever et al. (2014). In a mere three years from those papers, NMT systems were outperforming SMT systems for several translation tasks at the Association for Computational Linguists' Conference on Machine Translation (WMT). At the same time, large translation providers such as Systran, Google and Microsoft announced deployments of NMT systems for public consumption. The combination of these factors quickly made both buyers and providers of translation services aware of the new opportunities.

The rapid emergence of NMT has necessitated that LSPs focus on many new areas, including: qualitative evaluation of individual NMT systems, comparing translation quality and productivity of NMT and SMT systems, implementation and deployment of NMT systems, and building customized NMT systems for specific domains and/or clients.

2 Contextualization

Since the deployment of machine translation technology for commercial use, and especially the breakthrough of Statistical MT solutions, requests for MT as part of regular translation programs have constantly been on the rise. An explosion in the amount of content published as well as increasing pressure to publish content fast and simultaneously in different target markets and languages have caused clients to look into alternative, cheaper options and LSPs to adjust their translation workflows and processes. The continually improving quality of MT systems and new developments such as NMT add to this demand.

As a major global LSP, we count a range of big global companies among our end clients, for whom we typically provide ongoing, on-demand translation services into 20+ languages, covering various content types (= enterprise-scale translation program). It is our role to advise our clients on new developments in (MT) technology, opportunities for automation and workflow improvement as well as cost and time savings in their translation needs. In our case, we do not work with one specific MT provider, but recommend the MT solution we consider the best fit for a given end client, based on their specific needs and setup.

The arrival of NMT therefore requires us to reevaluate existing MT programs as well as the MT solutions offered by different providers we work with.

3 Planning for NMT for Enterprise-Scale Programs

Since its breakthrough, NMT quickly showed great promise to be able to deliver noticeably higher quality raw machine translations, especially for historically challenging and expensive translation pairs like English-Japanese. In our planning, we therefore started to evaluate a range of the then available, initially generic NMT systems for their qualitative performance on a subset of languages.

The evaluation of these *generic NMT* systems was performed with suitable test content from clients that gave us their permission to use their content for this purpose. We compared these generic systems with the *existing, customized SMT* solutions that were in place for the respective client programs, using automatic scoring for BLEU, GTM, Nist, Meteor, Precision, Recall, TER and Edit Distance (Levenshtein[1]), a post-editing test and human evaluations (see 3.2 Evaluation Methodology for details). While generic NMT frequently outperformed customized SMT on various metrics, the results were inconsistent across content types and languages. Lacking lexical coverage from the generic systems added to this picture, with some languages benefitting more from the increased fluency and grammatical accuracy of the NMT system (e.g. Japanese) while other languages seemed to struggle more with the terminological inaccuracies (e.g. German), at least from a human evaluation viewpoint. Selected results from this study were presented during the 2017 Machine Translation Summit in Nagoya, Japan, the 2017 School of Advanced Technologies for Translators in Trento, Italy, and with the Translation Automation User Society's (TAUS) MT user group (Marg et al., 2017a,b). While results were still mixed at this early stage, they showed that, for some languages, already the generic NMT systems were performing equally well when compared with the established, customized SMT systems. With MT providers starting to make customizable NMT solutions available and the promise in relation to an even better performance from these, we then progressed to direct comparisons on *custom SMT* to *custom NMT*, partly in the form of official client pilot projects.

In the following paragraphs, we outline the different phases in the pilot, evaluation and subsequent migration to a customized NMT solution for a translation program of 28 languages.

3.1 Pilot Scope

For the pilot, we selected a subset of four languages out of the total 28. The selection of the languages was driven by several factors: 1) client priorities (translation volumes and cost) needed to be reflected, 2) we wanted to look at languages from different language families, 3) we had to stay within a fixed budget. Based on these parameters, German, French, Russian and Japanese were selected. We then went ahead with engine training in a commercially available, customizable NMT system. To ensure that results were comparable, the new NMT systems were trained with data identical to the data used for the existing SMT systems.

3.2 Evaluation Methodology

The setup of machine translation pilots is largely driven by client needs, the available budget, as well as the planned final program purpose and setup. Depending on this purpose and setup, one or more of the following options are usually selected to analyze the suitability and quality of a given machine translation engine:

- Automatic scoring: comparatively easy, quick and cost-effective analysis, thanks to our proprietary scoring tool; also the most common method for a quick comparison of different system builds and measuring quality on larger samples
- Human evaluation: a) for Utility to determine understandability for informational purposes only, b) for Adequacy/Fluency to get data on suitability for post-editing, c) in the form of an engine ranking of several engines, d) with error annotation to get a better picture on nature of errors per engine.
- Productivity testing: to get a picture of real post-editing performance, by measuring the time spent editing individual

[1] http://www.levenshtein.net/

sentences or averaged over larger documents, typically expressed as throughput in words per hour.

The long-term objective for the program in question was clearly defined: migrate an existing SMT post-editing program to NMT, in order to provide higher quality raw MT to post-editors, and eventually increase productivity and reduce cost. It was therefore important to include real productivity data in the pilot, more so than human evaluations and error annotations (at this stage).

For this particular pilot, we used the TAUS DQF Quality Dashboard[2], the related SDL Trados Studio plugin[3] and a proprietary analytics tool to capture throughput and productivity. Productivity was measured both on the customized SMT solution currently in place, and a customized NMT system, built with identical data.

Both translation and post-editing productivity, among other factors, largely depend on individual speed of the translator/post-editor. It is therefore recommended to use several resources for productivity tests and then average the results. For our pilot, we opted for two resources per language.

The decision to use the TAUS DQF Quality Dashboard and the related SDL Trados Studio plugin was driven by the following factors:

- Readiness due to existing company account with the Quality Dashboard
- Ease of use: SDL Trados Studio plugin enables fast and easy setup of test projects in the Quality Dashboard and Trados Studio.
- Known user interface: Testers can work in a familiar environment (Trados Studio), therefore their performance will not be affected by a new, unknown tool.

In addition to the productivity data, we also ran automatic scores on the completed translations for both custom SMT and custom NMT. As per our internal research over the past years, Edit Distance based on the Levenshtein algorithm seems to be one of the most useful automatic scores for comparing the quality of the raw MT for post-editing. It has turned out to be the most reliable metric in our evaluations as well as easily understandable for both translators and clients when shown in the form of a side-by-side comparison of edits (Marg et al., 2017a; Marg, 2016).

3.3 Pilot Take-Aways

Results from the pilot showed a clear productivity increase from customized NMT compared to the existing, customized SMT for German and Japanese, and lower, but still valid increases for French and Russian.

In contrast to the reliability of the Levenshtein Edit Distance in our evaluations over the past years, in the case of this pilot, Edit Distance results contradicted the increase in productivity for all languages but German. With Edit Distance being 3-6 percentage points higher from the customized NMT system for Japanese, French and Russian, this can be seen as a moderate difference, but still needs further research and investigation.

3.4 Next Steps

Based on the results of both the internal testing for various languages and content types (*generic NMT*, see 3 Planning for NMT for Enterprise-Scale Programs) and the client pilot for the selected languages (*customized NMT*), as well as general industry results, the client felt confident enough to go ahead and plan for a live rollout across 28 languages.

4 Migration

4.1 Assessment Criteria

When we selected the NMT provider for our client pilot, we made the decision based on the availability of customizable systems at that time, results from previous internal tests with this system, a good cooperation with the provider, the general customization options/ease of use, etc. After the completion of our pilot, other providers announced that they would release customizable NMT solutions later in 2018. To make sure to provide our client with the best option both technology- and cost-wise, we reevaluated the selection of the system to be used based on the following criteria:

- Customizable NMT readiness: later (other providers) vs. now (pilot provider)
- Connector to the existing Translation Management System (TMS): in place (other provider) vs. to be built (pilot provider)

[2] https://www.taus.net/quality-dashboard-lp
[3] https://www.taus.net/evaluate/dqf-plugin-for-sdl-trados-studio

- Customization options: What options for customization are exposed to the user? Is it possible, for example, to force client-specific terminology?
- Cost: Which of the available solutions would be more cost-effective overall?

For enterprise-scale translation programs, an automated workflow is essential. With several hundred to thousands of words processed per day and target language, manual file handling and injection of the machine translation output would simply not be manageable for project managers, both on client and on LSP side. This is where a TMS comes into play to:

- automate the injection of matches from the Translation Memory (TM), a database of previous translations, and
- automate the injection of machine translation, via an API connection to the MT system.

The development of such APIs or connectors between individual systems can be very costly and time-consuming. Therefore, using an MT system that already has a connector for the relevant TMS can decrease costs and time of deployment significantly. This would typically be the preferred option, provided this MT system is at least on par with systems that do not yet have such a connector (on par in relation to other decisive factors such as output quality and other costs). An existing API connection from our client's current TMS to their current SMT system was therefore the main reason to change the selection of the NMT system from the pilot provider to the client's existing SMT provider who would deploy customizable NMT later in 2018.

4.2 Rollout Plan

With the newly selected system, our NMT rollout plan had to factor in the following aspects:

- Languages available in generic NMT now + customizable as of release date
- Languages not available with NMT so far
- Current Edit Distance from existing SMT systems vs. Edit Distance from generic NMT now + anticipated Edit Distance with customizable version (all Levenshtein)

4.3 Challenges

Challenges during an early adoption enterprise-scale migration like the one described in this paper can be grouped into two categories:

- Availability of languages in the new system due to early adoption
- General migration challenges in relation to the involved technologies and processes

Due to the urgency of the planned migration, language availability and the resulting language migration sequence were the most pressing topics.

Out of the 28 languages to migrate for the program in question, 23 were available with generic NMT in the selected system—and were planned to be available as a customizable version later in 2018. 5 were not available with NMT at all and had to stay in the current customized SMT until this would change.

To potentially bridge the gap until customized NMT would become available, we decided to reevaluate the results from our internal tests with generic NMT. We scheduled an extended autoscoring comparison of the current customized SMT engines and generic NMT from the selected system for all 23 languages available with NMT thus far. We then came up with a definition of language groups based on their results from this comparison to determine which languages could potentially be moved to generic NMT prior to customization.

When it comes to general migration challenges, we first had to clarify whether the existing TMS would allow us to select different NMT systems (generic for some, custom for other languages). Additionally, as the MT provided by us is not only being used for post-editing by our own supply chain, but also that of other LSPs, changes in setup have to be communicated and managed with those LSPs to ensure continued stability for our end client. Finally, we would have to plan for additional post-editor trainings to help our supply chain with the change from SMT to NMT. Similar to publications by Burchardt et al. (2017) and Castilho et al. (2017), our evaluations had highlighted differences in the types of errors found in NMT and SMT output which would have an impact on the post-editing approach. While more analyses are required, it is important that the differences in error typology are communicated to all translation providers, to enable them to develop efficient methods and to address all errors to the required final translation quality.

4.4 Research Proposal and Conclusion

During our session at the 21st Annual Conference of the European Machine Translation Association (EAMT 2018), we would like to present initial findings from this early adoption migration to NMT on an enterprise scale. We would like to demonstrate the solutions we implemented for the challenges outlined above, share details on the language migration sequence established based on our test results, and outline what additional challenges we might have come across during the migration.

5 Acknowledgement

We would like to thank our colleagues Elaine O’Curran, Alex Yanishevsky, Naoko Miyazaki and David Landan for their contribution.

References

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio. 2016-05-19. *Neural Machine Translation by Jointly Learning to Align and Translate.* Accepted as oral presentation at the 2015 International Conference on Learning Representations (ICLR 2015). arXiv:1409.0473v7 [cs.CL]. Accessed 26 March 2018

Burchardt, Aljoscha, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter and Philip Williams. 2017. *A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines.* In The Prague Bulletin of Mathematical Linguistics (PBML), number 108, pages 159-170. https://ufal.mff.cuni.cz/pbml/108/art-burchardt-macketanz-dehdari-heigold-peter-williams.pdf Accessed 29 March 2018

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley and Andy Way. 2017. Is Neural Machine Translation the New State of the Art? In The Prague Bulletin of Mathematical Linguistics (PBML), number 108, pages 109-120. https://ufal.mff.cuni.cz/pbml/108/art-castilho-moorkens-gaspari-tinsley-calixto-way.pdf Accessed 29 March 2018

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau and Yoshua Bengio. 2014-10-07. *On the Properties of Neural Machine Translation: Encoder–Decoder Approaches.* In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8). arXiv:1409.1259v2 [cs.CL]. Accessed 26 March 2018

Kalchbrenner, Nal and Philip Blunsom. 2013. *Recurrent Continuous Translation Models.* In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1700–1709. Association for Computational Linguistics. http://www.aclweb.org/anthology/D13-1176 Accessed 26 March 2018

Marg, Lena. 2016. *The Trials and Tribulations of Predicting Machine Translation Post-Editing Productivity.* Presented at the 2016 Language Resources Evaluation Conference (LREC). http://www.lrec-conf.org/proceedings/lrec2016/pdf/810_Paper.pdf Accessed 26 March 2018

Marg, Lena, Naoko Miyazaki, Elaine O’Curran and Tanja Schmidt. 2017. *Comparative Evaluation of NMT with Established SMT Programs.* In Proceedings of MT Summit XVI, Vol. 2: Users and Translators Track, pages 166-178. http://aamt.info/app-def/S-102/mtsummit/2017/conference-proceedings/ Accessed 26 March 2018

Marg, Lena, Naoko Miyazaki, Elaine O’Curran and Tanja Schmidt. 2017. *Generic NMT vs. Established SMT—An Assessment in Relation to Post-Editing.* In 2017 School of Advanced Technologies for Translators (SATT) Teaching Material (available upon request from satt-2017@fbk.eu).

Sutskever, Ilya, Oriol Vinyals and Quoc V. Le. 2014-12-14. *Sequence to Sequence Learning with Neural Networks.* In Proceedings of Advances in Neural Information Processing Systems 27 (NIPS 2014). arXiv:1409.3215v3 [cs.CL]. Accessed 26 March 2018

https://www.taus.net/think-tank/news/press-release/dqf-and-mqm-harmonized-to-create-an-industry-wide-quality-standard Accessed 26 March 2018

https://www.taus.net/evaluate/dqf-plugin-for-sdl-trados-studio Accessed 26 March 2018

http://www.levenshtein.net/ Accessed 26 March 2018

https://www.taus.net/quality-dashboard-lp Accessed 26 March 2018

A Comparison of Statistical and Neural MT in a Multi-Product and Multilingual Software Company - User Study

Nander Speerstra
Machine Translation Researcher, Infor
Baron van Nagellstraat 89
Barneveld, Netherlands
Nander.Speerstra@infor.com

Abstract

Over the last 4 years, Infor has been implementing machine translation (MT) in its translation process. In this paper, the results of both statistical and neural MT projects are provide to give an insight in the advantages and disadvantages of MT use in a large company. We also offer a look into the future of MT within our company and to strengthen the implementation of MT in our translation process.

1 Introduction

In the last few years, we have seen a change of direction regarding machine translation approaches. In different domains, more research is being focussed on neural machine translation (NMT) in comparison to phrase-based statistical machine translation: in both the research environment (Bojar et al., 2016) and commercial companies like Google (Wu et al., 2016) and Microsoft (Awadalla et al., 2018) NMT is increasingly important.

In the context of commercial translations, the continuous improvement of (N)MT has not passed unnoticed. More and more language service providers (LSPs) are implementing machine translation into their translation workflows and in addition, translation teams in large companies are investing in machine translation as part of their translation processes.

As a large global software development company, Infor[1] translates its products into many languages. This paper summarizes the results of the investigations into the potential benefits of machine translation for a company with many products, many target languages and very different translation circumstances per product. This study consists of two main parts: SMT and NMT. First, we give a description of our experiments, after which the results of the experiments are described. Lastly, the results and impact on our company are discussed.

We had 2 main goals for this user study: to find out the current importance of (S)MT in our company and the potential benefits of moving to NMT in the future. These goals are discussed in Section 4.

2 Experiments

2.1 Background

Infor is an enterprise software company that currently markets more than 125 different products, translating any number of these into 49 separate languages. The translation process involves both internal translators (up to 15 languages) and LSPs. A visual representation of the MT workflow is presented in Figure 1. Once the documentation is finished by technical writers, the translatable files are pre-processed: sentences that have been translated in previous versions of the product are re-used to prevent re-translation of already translated content. Subsequently, machine translation and an automatic post editing script is run to fix some of SMT's errors. From here, the post-editing and translation are done by vendors or internal linguists, who also perform a quality check.

For many products, both the user interface and the documentation are translated into different languages. The documentation is written as online help and generally consists of relatively short sen-

[1] http://www.infor.com/

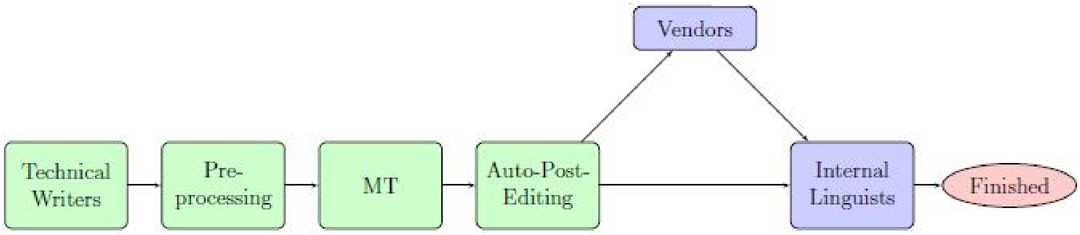


Figure 1: Translation workflow within Infor for MT projects

tences (1-15 words) with formatting and other tags. An example of this documentation material (English to Dutch) is given in Figure 2. User interface sentences often contain only one or very few words, which makes translation more difficult: often, different translations fit due to the shortness of the sentences while only one translation is terminologically correct.

The frequency of a product translation cycle varies: depending on the product, translation of edited existing and additional new materials may occur once, twice or twelve times per year. In addition, the number of times a product has been translated before (i.e. the amount of available training data) differs significantly: some products do not have a previous translation, others have been translated for over 20 years to certain languages.

As an example, the size of machine translation projects for three official Infor products (Infor LN, Infor BI and Infor d/EPM) is given in Table 1. The number of machine translated words differs per translation project, as does the update frequency.

Table 1: Number of machine translated words of 3 recent MT projects

Product	# words	# languages	Update cycle
LN	77,726	5	semi-annually
BI	109,922	7	annually
d/EPM	306,331	8	semi-annually

Most of Infor's documentation is written in US English and MT tests have only been performed on projects with English as the source language.

2.2 Statistical machine translation

Since 2014, MT projects have been executed at Infor using a moses-based statistical machine translation system from Morphologic Localisation: Globalese[2]. A handful of documentation translation projects were chosen as test projects for integrating MT in the translation process. These MT projects shared the following characteristics:

- They contained enough machine translatable segments to be worthwile
- There was sufficient training data (at least 50,000 sentences)
- Only some target languages were chosen of which most were close to the source language (English)

Two products were recurring to be machine translated for each occurring product update: LN and BI. The results of these product translations over the past two years (2016 and 2017) are discussed in Section 3.1.

For each of the products, one SMT system was used per language pair; i.e., if a product was translated to 6 languages, 6 SMT systems were trained and used for translation. This reflects the use of MT within Infor: we currently use one SMT system per product per language pair, as we do not generate parallel translations.

During the first tests, we noticed that MT makes a specific set of mistakes - often different mistakes per language. Therefore an automatic post-editing (APE) script was created that fixed basic errors introduced by the system, especially concerning tags. Example: 'Click on the <name> button' was machine translated to Dutch as

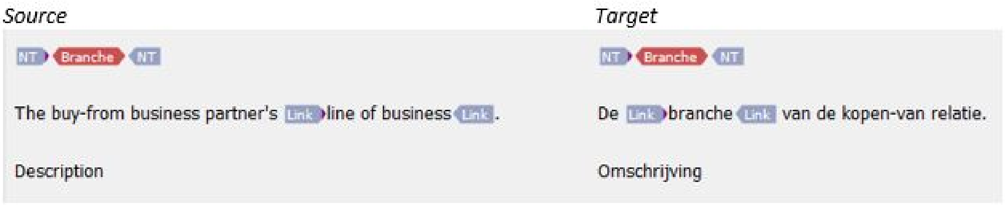
> Druk op de <name> knop

while the following translation would have been correct:

> Druk op de knop <name>.

APE fixes were only created for languages close to the source language (English), because the fixes required language-specific knowledge.

[2] http://www.globalese-mt.com/



Source	*Target*
NT Branche NT	NT Branche NT
The buy-from business partner's Link line of business Link.	De Link branche Link van de kopen-van relatie.
Description	Omschrijving

Figure 2: Example of Infor documentation, product LN (English to Dutch)

While we added MT to the translation workflow for the above mentioned MT projects, we also ran tests on other products in order to find out if we could use MT for projects with:

- User Interface translations
- A low number of training segments
- Languages that are not closely related to English

These last tests were evaluated based on the expert opinions of our internal linguists and are not based on statistics. The reason for this is that the currently used evaluation metrics like BLEU and NIST correlate poorly with human judgment (Wang and Merlo, 2016), and our linguists have to work with the MT output: their opinions outweigh the statistical outcomes when a decision is made about using MT in translation projects.

Each of the SMT projects was set up with a quality threshold[3] and only segments with a quality estimation score of over 85% were retained, because sentences with lower scores were found to be sufficiently lacking in quality as to render them unusable. We selected this threshold after an evaluation of a first set of projects.

The results of these tests are shown in Section 3.1.

2.3 Neural machine translation

In the last few years, NMT has been the main interest in the machine translation industry. Globalese has recently released Globalese 3[4], a neural machine translation system which has subsequently been tested extensively at Infor. NMT is supposed to have several advantages over SMT. First, we explored the advantages of NMT. Then, we focused on tests using Globalese 3.

The differences between SMT and NMT systems have been researched in depth and Jean et al. (2014) discuss several advantages of NMT. First, NMT requires very little domain knowledge. Where SMT requires a language model, NMT does not assume any linguistic characteristics and simply reads the source and target sentences as is. Moreover, an NMT model is trained as a whole, whereas an SMT engine consists of several separately trained parts including but not limited to (one or more) phrase table(s) and a language model. NMT also uses less memory than SMT systems that need to process large tables containing sentence pairs. Lastly, research has shown that NMT is more fluent and more accurate regarding word order (Toral and Sánchez-Cartagena, 2017).

Some of the disadvantages are discussed by Wu et al. (2016). The models need more training time than SMT models, NMT has difficulties with rare words and sometimes it translates sentences syntactically incorrectly. Also, long sentences are more often translated poorly by NMT (Toral and Sánchez-Cartagena, 2017).

For our company, some of the disadvantages appear to be less relevant since Infor's documentation contains very domain-specific terminology and rare words are not used frequently. Also, sentences are often relatively short. However, problems like an increased training time do matter: with many products and many languages to translate to, more training time could require a larger investment in resources.

One of our main questions is regarding the number of viable target languages. For SMT, we found that only languages related to English (Romance and Germanic languages) result in workable machine translations. Will NMT enable us to translate into additional languages, as Microsoft claims its new NMT system does with Chinese (Awadalla et al., 2018)?

As of Globalese 3.1, it is possible to use *core*

[3] https://web.archive.org/web/20150209082134/http://www.globalese-mt.com/product/features/quality-estimation/

[4] http://www.globalese-mt.com/2017/09/05/globalese-3-0-released/

and *auxiliary* corpora as training data[5]. This core function makes sure that the core vocabulary is not overruled by the larger auxiliary corpora and, at the end of the training phase, the engine is further tuned to the core corpus. We created a test for Dutch, German and Russian, where an older BI project was selected to be re-translated with newly set up NMT engines. For each language, the translatable segments were processed with the following three machine translation systems:

- SMT
- NMT
- NMT with core functionality

The engines (SMT, NMT and NMT with core functionality) were trained using the number of training segments shown in Table 2. For this test the aforementioned SMT quality threshold of 85% was removed because the NMT systems from Globalese did not have a quality estimation script with which to compare. The test files for all engines were pre-translated as usual and the remaining 7203 sentences (77,261 words) were machine translated. These sentences were evaluated by internal linguists (one linguist per language).

Table 2: SMT vs. NMT: Translation project training size for Dutch, German and Russian

Language	# training segments
Dutch	499,106
German	275,887
Russian	198,360

This test includes two of our main questions: do we need more data with NMT than with SMT (i.e. will Russian and German be evaluated with worse results for NMT than for SMT) and can we translate to more languages without quality loss (i.e. are the evaluations for Russian similar to those for Dutch and German)? The three sets of translated files were given to internal linguists for evaluation without information on the engines that were used to produce them.

3 Results

Normally, machine translation results are expressed using evaluation scores like METEOR, BLEU and/or hTER. However, as these metrics generally do not correlate with linguists' findings (Sun, 2010), we chose to only report the number of machine translated segments (that were used in the translation projects) and the qualitative analyses of our linguists. Both the linguist reviews and the number of machine translated sentences gave us an indication of the usefulness of MT in translation projects.

3.1 Statistical machine translation

In the period 2016-2017, roughly 900,000 words have been machine translated using SMT for the products Infor LN and Infor BI. In Table 3, the number of translated words is shown for the last 2 years. The decreased number of machine translated words for BI in 2017 is caused by changes in the MT setup as a result of an evaluation of the 2016 results. These changes are discussed in Section 4.1.

Table 3: Number of SMT words for 2 products, in the period 2016-2017

Product	2016	2017	Total
LN	285,857	292,095	577,952
BI	259,530	56,174	315,704
Total	545,387	348,269	893,656

SMT was found to be useful in the translation projects of 10 products with a total of 2,026,760 machine translated words. In the largest MT project (BI 2016), translations were run from English to 12 different languages: Brazilian Portuguese, Danish, Dutch, French, German, Italian, Japanese, Norwegian (Bokmål), Russian, Simplified Chinese, Spanish and Swedish.

For three tests, the quality of the translations was insufficient for use in actual translation: tests of user interface translations, projects with a low amount of training segments and target languages that are not closely related to English. The user interface translations contained sentences that were too short and ambiguous for MT, which often led to incorrect translations. Projects with a low number of training data often resulted in very few workable translations due to the quality estimation threshold of 85%. Unrelated target languages resulted in poor translations and were not selected for new translation projects.

We did not have statistical metrics for the MT projects, but the discount on MT words is an indication of the importance of MT. For the project

[5] http://www.globalese-mt.com/2017/10/31/augmented-in-domain-engines/

BI 2016, we were given an average discount of 67% on machine translated sentences on an average word price of 15 ct/w. To that extent, the BI 2016 project led to a cost saving of €25,953.

3.2 Neural machine translation

Besides the motivations for using NMT over SMT in the literature,we performed a qualitative analysis on 3 sets of translations of the product Infor BI: translations using SMT, NMT and NMT with the core functionality. Internal linguists, one per language, were asked to rank the quality of the translation sets and give examples of correct and incorrect translations. Each of them returned the following ranking: (1) NMT with core functionality, (2) NMT and (3) SMT. The quality of (1) and (2) was comparable but with a slight preference for (1), (3) was said to have less workable translations compared to (1) and (2). This was expected for Dutch and German as we had enough training data for those languages, but also our Russian team evaluated NMT as more useful than SMT. The linguist for Dutch mentioned the quality of NMT with core functionality as follows: 'I think this version of the project is very good and MT is a great time saver here, not only because post editing doesn't seem so strenuous.'

For all languages, the results can be summarized as follows. SMT had many different issues, from incorrect word/tag order, incorrect capitalization, incorrect word order to illogical translations. Although most issues are minor, they were too numerous to make the translations directly usable and required heavy post-editing.

NMT and NMT with core functionality also had difficulties with word/tag order and word order in general. And, in contrast with SMT, NMT made strange (albeit fluent) semantic errors, where the translation was incomprehensible. An example of such an NMT error is shown in Figure 3, together with examples of errors concerning text in tags and word omissions. But compared to SMT, NMT was said to contain more workable translations and would take less post-editing time. Short sentences especially were much more often correct.

Consequences of this test will be discussed in Section 4.2.

4 Discussion

In this section, the results of the SMT and NMT experiments are discussed.

4.1 Statistical machine translation

As described in Section 3.1, about 2 million sentences have been machine translated with our SMT engines in the period 2014-2017. There are several points of interest that need a more elaborate discussion: the output quality, the number of languages found workable for SMT and the project initiation time.

4.1.1 Output quality

Overall, the output quality was good enough to use MT in translation projects. As this was a goal of machine translation (decreasing costs by post-editing instead of translating from scratch), SMT has been successfully used in translation projects. Because of the 85% threshold in official projects, about 40-50% of the translatable segments were actually machine translated. Increasing the quality of the output (and thus increasing the number of machine translated segments) is one of the key research areas within our company, as this affects the costs of translation projects directly.

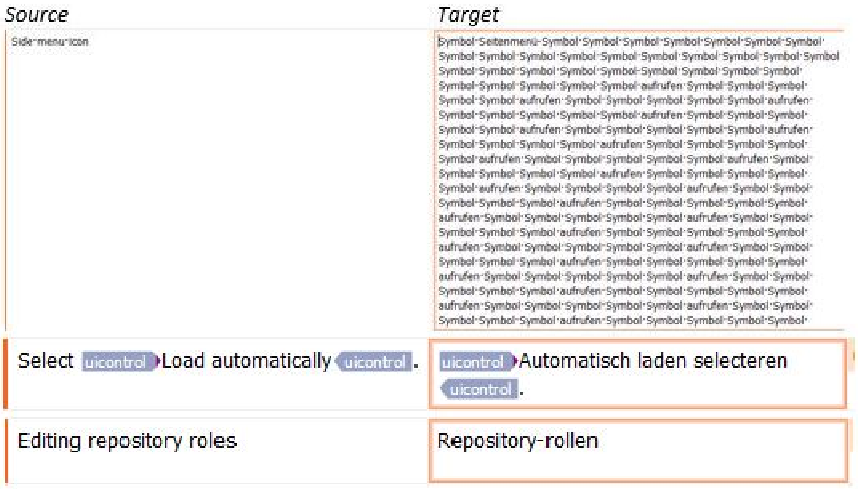
4.1.2 Number of languages

During our experiments, we found that target languages close to the source language were translated with a higher quality than target languages outside of the Romance and Germanic families. Since our projects have English as the source language, Germanic and Romance languages were most suitable for machine translation. Early tests on Chinese (zh-CN) and Japanese showed that, to our standards, those languages resulted in a quality unsuitable for use in actual projects.

Another issue with SMT was the necessity of an automatic post-editing script. This script fixed some known issues for specific languages, but this could only be set up by language experts. As our team does not have expertise in languages outside the Germanic and Romance families, only these languages had APE scripts.

4.1.3 Project initiation time

Because SMT requires several individual components to be trained, re-training the engines for a translation project was sometimes rather time-consuming. Especially when the number of languages in a project was high, it took several hours to manually prepare the engines. Although some actions were scripted, uploading new training segments and creating engines was at the time of the



Source	Target
Side·menu·icon	Symbol·Seitenmenü·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol· Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol·Symbol· Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol· Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen· Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol· Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen· Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol· Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol· Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol· Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol· Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol· aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol· Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol· aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol· Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol· aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol· Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol· aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol· Symbol·Symbol·Symbol·aufrufen·Symbol·Symbol·Symbol·Symbol·Symbol·
Select uicontrol Load automatically uicontrol .	uicontrol Automatisch laden selecteren uicontrol .
Editing repository roles	Repository-rollen

Figure 3: Examples of NMT errors in our test project: hallucination translations, words incorrectly placed in tags and omission of words

test project not yet available. We updated the engines after every project to make sure that the engines are trained on as much data as possible.

4.2 Neural machine translation

The tests using our neural engines have given a useful insight in the advantages and disadvantages of neural machine translation. In this section, the advantages and disadvantages are discussed. In both sections, a link is made to our SMT results.

4.2.1 Advantages

First of all, our test on Dutch, German and Russian showed that for all 3 languages (1) the NMT quality resulted in workable translations and (2) NMT is preferred over SMT. Where (1) was expected for the closely related language pairs English-Dutch and English-German, we weren't certain for Russian: in our SMT projects, the Russian language appeared to be too different from English to obtain workable translations. But the experiment showed that NMT resulted in useful output for Russian as well. Secondly (2), NMT was preferred over SMT for every language pair.

Another advantage of NMT is the time gain when preparing the engines. Because only 1 model was needed per language (compared to the multiple components in an SMT engine), the preparation time was significantly lower: for the NMT models in the test, preparation took only 10 minutes instead of the 30 minutes that it took to set up the SMT engines.

4.2.2 Disadvantages and solutions

Our NMT tests also revealed some of the downsides of NMT: training the engines took much longer than with SMT (2-3 times longer), more data was needed and the output was sometimes less reliable.

The training time appeared to be problematic at first, because we re-trained the engines before each project. This would require more resources with NMT with the same (or more) languages per product. Given the product LN (6 engines, one for each package, and 6 languages) and a training time of 2 days, this would result in a total semi-annual training time of 72 days (3 months on every 6 months).

However, we have not investigated whether it is necessary to update the engines after each translation project. If we would only update once a year, the effect on our resources is reduced.

Furthermore, we needed more data. Although not presented here, we have translation projects with less than 50,000 segments as training data. SMT was capable of generating qualitative output (for closely related language pairs), NMT was not. However, due to the core functionality function, we have been able to merge data from several projects into one large engine without causing terminology issues. This is a major improvement, as we can potentially machine translate each product for which we have enough training data in that specific language. This would also decrease the necessity of re-training engines after each project, because the new set of translations would have less impact in the large engines.

Lastly, the output was less reliable. Although

NMT is more fluent (Skadina and Pinnis, 2017), the output is less accurate and can sometimes miss the point completely. But this has been found to be an advantage by some of our linguists: because translations are more fluent than with SMT, it is easier to see that the translations should be removed and re-translated from scratch. This saves time when post-editing MT sentences.

5 Conclusion and outlook

In this paper, we have discussed the outcomes of statistical (SMT) and neural (NMT) machine translation experiments that we have conducted at Infor. With a total of over 2 million machine translated words, SMT has become a significant factor in product translations. SMT has been used for 10 products with up to 12 languages. Tests showed that SMT produced workable translations on language pairs that are closely related, and we needed handwritten auto-post-editing scripts to improve the output quality. A first test with NMT has shown that NMT performs better on all languages tested (Dutch, German and Russian) than SMT.

The purpose of the experiments was to determine the significance of MT in our workflow and whether NMT is the next step to take. Based on the number of machine translated words in the last few years, we now have a good understanding of the type of projects in which MT is of use, and it has already impacted the costs of translation projects in which MT was used. We have also seen that NMT scores higher than SMT according to our linguists, which is a clear indication that NMT is the next step in improving our MT process. With a potential of many more products to translate and many more languages to translate to, we will start experimenting with NMT in the same way that we did with SMT.

Acknowledgements: I would like to thank Gábor Bessenyei and Gergely Horváth from Globalese for giving us the opportunity to test their new neural machine translation system. I would also like to thank Thijs Trompenaars and my colleague John Musters for their important and useful reviews on this paper.

References

Awadalla, Hassan, Hany and Aue, Anthony and Chen, Chang and Chowdhary, Vishal and Clark, Jonathan and Federmann, Christian and Huang, Xuedong and Junczys-Dowmunt, Marcin and Lewis, Will and Li, Mu and Liu, Shujie and Liu, Tie-Yan and Luo, Renqian and Menezes, Arul and Qin, Tao and Seide, Frank and Tan, Xu and Tian, Fei and Wu, Lijun and Wu, Shuangzhi and Xia, Yingce and Zhang, Dongdong and Zhang, Zhirui and Zhou, Ming. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *In proceedings*.

Bojar, Ondrej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Jimeno Yepes, Antonio, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation (WMT16). *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*.

Jean, Sébastien and Kyunghyun Cho and Roland Memisevic and Yoshua Bengio. 2014. On Using Very Large Target Vocabulary for Neural Machine Translation. CoRR, Vol. abs/1412.2007.

Skadina, Inguna and Marcis Pinnis. 2017. NMT or SMT: Case Study of a Narrow-domain English-Latvian Post-editing Project. *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*.

Sun, Yanli. 2010. Mining the Correlation between Human and Automatic Evaluation at Sentence Level. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Toral, Antonio and Víctor M. Sánchez-Cartagena. 2017. A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions. CoRR, Vol. abs/1701.02901.

Wang, Haozhou and Paola Merlo. 2016. Modifications of Machine Translation Evaluation Metrics by Using Word Embeddings. *Proceedings of the Sixth Workshop on Hybrid Approaches to Translation*, Osaka, Japan, December 2016, 33–41. *CoRR*.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*.

Translators’ papers

Does Machine Translation Really Produce Translations?

Félix do Carmo
ADAPT Centre / CTTS
Dublin City University
felix.docarmo@adaptcentre.ie

Abstract

I will try to answer the question of whether Machine Translation (MT) can be considered a full translation process. I argue that, instead, it should be seen as part of a process performed by translators, in which MT plays a fundamental support role. The roles of translators and MT in the translation process is presented in an analysis that get its elements from Translation Studies and Translation Process Research.

1 Introduction

This presentation is based on my research, which covers both Translation Studies and Machine Translation, and on my practice of more than 20 years as a translator, a translation company owner and a translator trainer. The main results of my research are my PhD thesis and KAITER, my project as an EDGE Fellow at the ADAPT Centre.

2 Translation and post-editing

Post-editing (PE) is a term used by the industry for a process in which translators work over a version of a source text (ST) in the target language, created by an MT system. One should discuss to what extent is this version an actual translation.

2.1 Definition of translation

Translation is a text transformation process by which a text's communication effectiveness is improved by replicating the text in a language code different from the original one. For this improvement of effectiveness to be reached, the process must be as efficient as possible.

This definition of translation has one clear implication for MT: if it does not improve the effectiveness of the ST, and if it is not more efficient than human translation, it does not fulfil the requirements of a translation process. So, every MT product that requires subsequent work by a translator is an evidence that MT is not a complete translation process.

Some of the practices in MT research cannot hold against this view. Monolingual PE, for example, cannot guarantee the effectiveness of communication of the ST, since it has no access to the ST.

2.2 Post-editing is not just revision or editing

When they are post-editing, translators work on segments that may require that only a few words need editing, but also on segments that imply total rewriting. Besides, they work with results from not only MT but also Translation Memories. So, they need to read several text extracts in both languages, while the reliability of the suggestions they receive is not established. To guarantee the quality that is requested, translators also need to resort to reliable external sources of reference. In this complex work environment, translators must be very efficient readers, and they need to make good and fast decisions. This means that post-editors must be specialised translators, which again shows that MT is not a full translation process.

3 Conclusion

A clear understanding of the different dimensions of translation and PE, as specialised processes, shows that MT, more than an autonomous translation process, achieves its best potential as a support for human translation and editing tasks.

Acknowledgements

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under the EDGE COFUND Marie Skłodowska-Curie Grant Agreement no. 713567. This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number 13/RC/2077.

References

do Carmo, Félix. 2017. *Post-Editing: A Theoretical and Practical Challenge for Translation Studies and Machine Learning*. Universidade do Porto. https://repositorio-aberto.up.pt/handle/10216/107518.

Pre-professional pre-conceptions

Laura Bruno
Antonio Miloro
Paula Estrella
Facultad de Lenguas, UNC
Córdoba, Argentina
laurabruno@fl.unc.edu.ar
tr.miloro@gmail.com
pestrella@famaf.unc.edu.ar

Mariona Sabaté Carrove
Facultad de Letras
Universitat de Lleida
Pl. de Víctor Siurana, 1
E-25003 Lleida, Spain
msabate@dal.udl.cat

Abstract

While MT+PE has become an industry standard, our translation schools are not able to accompany these changes by updating their academic programs. We polled 100 pre-professionals to confirm that in our local context they are reluctant to accept post-editing jobs mainly because they have inherited pre-conceptions or negative opinions about MT during their studies.

1 Pre-professionals and MT+PE

Following global trends, the translation industry in Latin America has dramatically changed over the years and Argentina is emerging as an important translation provider. However, our local context is quite different from that of the USA, Canada or Europe, as we are a monolingual country and placed at the end of the supply chain, with more intermediaries between MT producers and Post-Editors. While the emergence of new MT+PE technologies has had an impact on businesses and companies, our translation schools have fallen short of this challenge and failed to update their academic programs. Therefore, our soon-to-be professionals are not acquainted with the MT+PE process and they inherit old prejudices that sometimes do not even hold anymore. We felt the need to ask those future translators their opinion about the technologies they will come across once they enter the labour market. So, we polled 100 advanced undergraduates students in their 4th and 5th year (i.e. *pre-professionals*) and invited them to answer 8 questions about their experience and opinion on MTPE. For space reasons the questions and detailed results are not included in this abstract. Our sample consists of 90 non-bilingual informants from the School of Languages, Universidad Nacional de Córdoba, *UNC* and 10 bilingual (Spanish-Catalan) informants from the School of Arts, Dept. of English and Linguistics, University of Lleida (*UdL*). We decided to include students from these two institutions to compare the results obtained as both groups of informants have a different academic and cultural profile and background, but none of them reported any experience in PE. When compiling the results we found that opinions about MT were mostly negative in 40% (UL) and 50% (UNC) of the cases while opinions about PE were mostly positive 43% (UL) and 40% (UNC) of the cases. When looking at the participants' comments, we found that most negative comments revolved around the payment model, MT producing awkward output or errors, and not having the chance to practice PE during their studies. On the other hand, positive comments were about recent improvements of MT quality, MT+PE speeding-up their tasks and helping them develop other skills. These results are consistent with our expectations and confirm that new professionals in our local context are reluctant to accept post-editing jobs mainly because they already have prejudices or negative opinions about MT. To tackle this issue, we are currently working on extra-curricular training on MT+PE to help them take more informed decisions and to give them the opportunity to gain expertise on this technology. Obtaining and analyzing feedback from future translation professionals is an essential step and will also help shape an alternative course methodology to be implemented at an early stage rather than introducing MT + PE in the final academic years.

Determining translators' perception, productivity and post-editing effort when using SMT and NMT systems.

Ariana López Pereira
Universitat Autònoma de Barcelona
arianalopezp@gmail.com

Abstract

Thanks to the great progress seen in the machine translation (MT) field in recent years, the use and perception of MT by translators need to be revisited. The main objective of this paper is to determine the perception, productivity and the post-editing effort (in terms of time and number of editings) of six translators when using Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) systems. This presentation is focused on how translators perceive these two systems in order to know which one they prefer and what type of errors and problems present each system, as well as how translators solve these issues. These tests will be performed with the Dynamic Quality Framework (DQF) tools (quick comparison and productivity tasks) using Google Neural Machine Translation and Microsoft Translator (SMT) APIs in two different English into Spanish texts, an instruction manual and a marketing webpage. Results showed that translators considerably prefer NMT over SMT. Moreover, NMT is more adequate and fluent than SMT.

1 Introduction

Machine Translation (MT) is nowadays one of the most useful resources for translators and the translation industry. Post-editing has become a usual practice within companies (Torres Hostench et al., 2016). With the great progress seen in NMT (Castilho et al., 2017), there are still some problems to overcome when using it, especially regarding terminology issues. Despite these innovations, SMT systems are still very popular. Hence, it is important to discover the differences between the two systems in order to use them properly.

2 Aim of this proposal

The aim of this paper is to determine the translators' perception when using SMT and NMT, as well as to observe the differences when using SMT and NMT based on the topic of the source text. The research questions addressed will be:

- Do translators prefer SMT or NMT?
- Which issues present the use of SMT and which ones NMT? Does the SMT present more accurate results? Is the NMT more fluid?
- Are these issues different based on the topic of the text (marketing and user documentation source texts)?
- How do the translators post-edit these issues?

Results showed that the translators preferred NMT, which was more fluent and adequate than SMT. NMT was both more adequate and fluid, both for the instruction manual and the marketing texts. SMT presented best results in the marketing test, compared to the user documentation test.

References

Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J., & Way, A. (2017). Is Neural Machine Translation the New State of the Art? The Prague Bulletin of Mathematical Linguistics, 108(1). https://doi.org/10.1515/pralin-2017-0013

Torres Hostench, O.; Cid-Leal, P. and Presas, M. (coord.) (2016). El uso de traducción automática y posedición en las empresas de servicios lingüísticos españolas: informe de investigación ProjecTA 2015. Bellaterra.

Machine translation post-editing in the professional translation market in Spain: a case study on the experience and opinion of professional translators

Lorena Pérez Macías
University Pablo de Olavide, Faculty of Humanities, Seville, Spain
lpermac@upo.es

Abstract

The objective of this paper is to analyse some aspects related to the practice of post-editing services in the current translation market in Spain. To this aim, some quantitative data collected through an online survey and concerning the experience and opinion of professional translators regarding post-editing will be shown.

1 Introduction

The continuous progress in new technologies has helped to the development of new translation techniques such as machine translation post-editing in recent years. However, the emergence of machine translation has generated an intense debate between those who think that technology will eventually replace human translators and those who see machine translation as an ally that allows the professional translator to perform translations with no risk of being completely replaced (Alonso and Calvo, 2015). Taking into account some previous studies on post-editing in the language services sector in Spain, such as Rico y García (2016) and the report of the ProjecTA group (2016), a practical and interpretative data collection study was carried out in 2017 (Pérez, 2017). The aim of this study was to explore the reality surrounding perceptions of machine translation post-editing in the professional translation market in Spain on issues such as quality, productivity or ethics, among others.

2 Methodology

In this study, a methodological triangulation has been carried out. First of all, with an exhaustive review of the most up-to-date literature in this field, in an attempt to detect points of interest and trends that could be explored. This was followed by a preliminary focus group to approach the topic (qualitative method of initial exploration), in which a first contact was made with a sample of the study's target audience and which served to categorise the main points that would form part of the research. A survey (quantitative instrument) was then developed using these data and served as the main tool for collecting information. In order to optimize the design of the survey, it was necessary to pass a series of control tests (robustness tests) before its launch to avoid problems of understanding on the part of the respondents or the inclusion of certain biases.[1] The target population of the study consisted of the total number of active translators in Spain who are familiar with post-editing, whether or not they have experience in this field and the sampling method was snowball sampling. The final sample was composed of 104 subjects from the target population.

3 Conclusions

This research has made it possible to evaluate post-editing as a field that raises major professional and ethical dilemmas, since it involves many factors, including productivity, quality, and the context of subordination in translation, among others.

References

Alonso, Elisa and Calvo, Elisa. 2015. Developing a blueprint for a technology-mediated approach to translation studies. *Meta: Translators' Journal* 60(1): 135-157.

Pérez, Lorena. 2017. *Análisis de las percepciones en torno a la práctica de la posedición en el sector profesional de la traducción en España.* PhD thesis. Universidad Pablo de Olavide.

PROJECTA. 2016. *Informe de investigación ProjecTA 2015. El uso de traducción automática y posedición en las empresas de servicios lingüísticos españolas*. https://ddd.uab.cat/pub/estudis/2016/148361/usotraaut_2016.pdf (consulted 20.2.2018).

Rico, Celia and García, Álvaro. 2016. *Análisis del sector de la traducción en España (2014-2015).* Universidad Europea de Madrid, Villaviciosa de Odón.

[1] The final version of the survey is available at the following link: https://www.upo.es/limesurvey/index.php?r=survey/index&sid=272728&lang=es

Perception vs. Acceptability of TM and SMT Output: What do translators prefer?

Pilar Sánchez-Gijón
Grup Tradumàtica
Dept. de Traducció i d'Interpretació i d'Estudis de l'Àsia Oriental
Universitat Autònoma de Barcelona
pilar.sanchez.gijon@uab.cat

Joss Moorkens
ADAPT Centre
School of Computing
Dublin City University, Ireland
joss.moorkens@dcu.ie

Andy Way
ADAPT Centre
School of Computing
Dublin City University, Ireland
andy.way@computing.dcu.ie

Abstract

This paper reports the results of two studies carried out with two different group of professional translators to find out how professionals perceive and accept SMT in comparison with TM. The first group translated and post-edited segments from English into German, and the second group from English into Spanish. Both studies had equivalent settings in order to guarantee the comparability of the results. It will also help to shed light upon the real benefit of SMT from which translators may take advantage.

1 Introduction

Machine Translation (MT) remains unpopular among translators. Even though MT seems to be rejected because of its lack of quality, translators may be reluctant to use MT for many other reasons (Ferreras, 2017). This paper tries to approach translators' perception of SMT raw output in comparison with TM.

For that purpose, two different studies were carried out. The first one involved seven professional translators (Moorkens and Way, 2016). They were asked to rate 60 English-German translated segments. 30 of them were segments from a domain-appropriate TM, but without the quality threshold being set. The other 30 segments were translated through an SMT system. This study was replicated (Rico, Sánchez-Gijón and Torres-Hostench) with professional translators from English to Spanish.

2 Aim of this proposal

This paper aims to determine whether translators' reluctance to use MT correlates their preferences choosing translation suggestions. The research questions that will be addressed in this paper are:

- Do translators edit any MT translation proposal if it is available?
- Do translators prefer TM high fuzzy matches (up 85%) than MT proposals when there is no information about their origin (i.e., proposals are presented without any metadata)?
- Are there any difference in their preferences between EN-DE and EN-ES translators?
- Is the methodology of these studies suitable to measure MT degree of acceptance in comparison to TM while translating or post-editing?

Results will show that, in fact, MT acceptance increases when translation proposals are presented without metadata.

References

Ferreras, Olga. 2017. *La satisfacción de los usuarios con la traducción automática.* Masters' Dissertation, Màster Traduática: Traducció i Tecnologies, UAB, July 2017.

Moorkens, J., and Way, A 2016. *Comparing Translators Acceptability of TM and SMT Outputs.* Baltic J. Modern Computing, 4(2):141-151.

Rico, C., Sánchez-Gijón, P., and Torres-Hostench, O. (2018). The Challenge of Machine Translation Post-editing: An Academic Perspective. *Trends in E-Tools and Resources for Translators and Interpreters*, Brill: 203-218.

Learning to use machine translation on the Translation Commons Learn portal

Jeannette Stewart
Translation Commons
Las Vegas, NV 89113, USA
jeannette@translationcommons.org

Mikel L. Forcada
Universitat d'Alacant
E-03690 St. Vicent del Raspeig, Spain
mlf@ua.es

Abstract[1]

We describe the Learn portal of Translation Commons (TC), a self-managed community of volunteer translators community aimed at sharing tools, resources and initiatives for the translation community as a whole. Members are encouraged to upload and share their free resources on the platform and to create free courses and tutorials. Specifically there are no educational material on machine translation yet and we invite experts to contribute.

1 Translation Commons

A self-managed volunteer community, Translation Commons[2] (TC) is a nonprofit established to share tools, resources and initiatives that unite the language community and encourage cross-functional collaboration. TC fosters collaboration, responds to the needs of the people using endangered and minority languages and is targeted to the needs of language service professionals and students by bridging the gap between academia and industry.

2 The Learn portal

Learn is the TC portal for all community learning activities. The portal includes a Learning Center with learning materials and courses, a Translation Hub compiling valuable free resources for translators, and a section hosting a revival of the eCoLo (electronic Content Localisation) translation training initiative, spearheaded by an EU-funded consortium of European universities. Learn contains almost no educational material on machine translation yet; we invite experts to contribute.

2.1 The Learning Center

In this section members find, join or create courses, workshops, seminars, one-to-one training, as well as articles, resources and tutorials. The eCoLo Training kits are being updated and will be located under courses here. We encourage members to share or create tutorials or any educational material.

2.2 The Translation Hub

The Translation Hub (TH) is a compilation of valuable online and offline resources for translators, such as terminology databases, glossaries, translation tools, public and private organisations linked to the language industry, and much more. The aim is to catalogue all free resources, including opensource and free trials for cloud-based and desktop-based software. TC members can upload links of free online resources through their dashboard and these will be added to any of the categories in the TH.

2.3 eCoLo

The eCoLo platform, which has recently been restarted at TC, provides useful training materials for both students and teachers in order to help improve skills in different areas of computer-assisted translation: translation memory, software localization, machine translation, project management, and terminology. You will find multilingual material, training kits, training scenarios and full courses on various translation and localization techniques.

[2] http://translationcommons.org

Use of NMT in Ubiqus Group

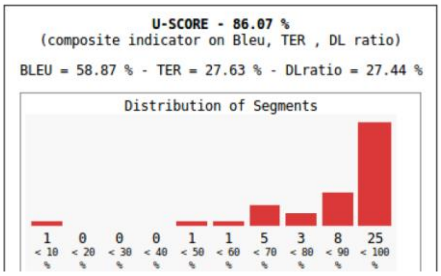
Paloma Valenciano
Translator / Posteditor
General Manager
Traducciones Políglota
paloma.valenciano@poliglota.com

Abstract

After more than 30 years' experience as a translator and as a reviser, I have recently started to post-edit. During these 10 months discovering a new approach to my profession, the experience has been highly positive.

Ubiqus, the French group to which we belong, has developed 20 engines based on OpenNMT. OpenNMT derives from an academic project initiated in 2016 by Harvard NLP; Systran joined the project and an open source toolkit was released in January 2017. The community grew when individuals as well as localization professionals contributed. Ubiqus adopted this toolkit at the very beginning of 2017 and contributed to its development as well as with some extensions, developing a layer to integrate OpenNMT in our workflow environments, including SDL Studio and with our internal ERP, which enables to provide a highly efficient end-to-end system.

I have been using the EN-ES and FR-ES engines mainly for legal texts. I very soon felt comfortable with the task, I started measuring my productivity by timing my output. I was surprised by the improvement since the very beginning, and as the NMT engine was further trained and I got more used to the post-editing task I achieved even better results, improving productivity by almost 30%.
Ubiqus has also developed a scheme for the systematic scoring of all translation jobs, U-Score, a composite indicator of the overall performance of the machine. The U-Score is obtained by aggregating the information of BLEU, TER and DL-ratio and averaging them. It then performs a transformation allowing to spread the scale a bit. The scores have been clearly improving in the last months with a constant training of the engines.



U-SCORE - 86.07 %
(composite indicator on Bleu, TER , DL ratio)

BLEU = 58.87 % - TER = 27.63 % - DLratio = 27.44 %

Figure 1: U-Score

Over the last 30 days, 24.2 million source words have been postedited within the Ubiqus Group using the 20 engines, which are constantly retrained.

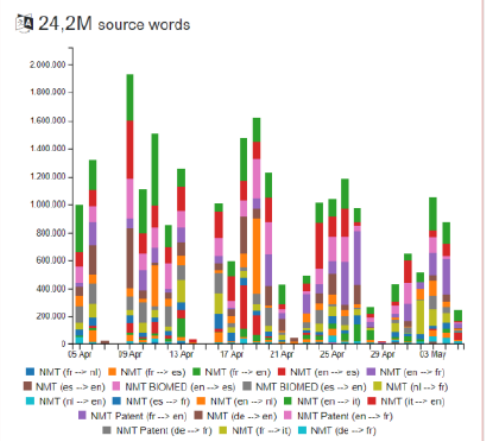


Figure 2: No. of words postedited, April 2018

An In-house Translator's Experience with Machine Translation

Anna Zaretskaya
TransPerfect
Passeig de Gràcia, 11, Esc. B, 5
Barcelona, Spain, 08007
azaretskaya@translations.com

Marcel Biller
TransPerfect
Passeig de Gràcia, 11, Esc. B, 5
Barcelona, Spain, 08007
mbiller@translations.com

With the expansion of MT usage at TransPerfect, we have developed an implementation strategy that involves continuous work with linguists on a wide variety of MT-related tasks. Today, MT undeniably plays a big role in translators' lives. As an internal linguist at TransPerfect, I have experienced it in my everyday work. A big part of it is now related to MT and these tasks include not only MT post-editing, but also MT evaluation and improvement.

I remember when MT was first introduced as a new task for the internal linguists: the transition was smoother for some of us than for others. As to my personal experience, at first I was rather sceptical. This is because I used to think that MT post-editing (MTPE) was rather similar to proofreading, but worse: instead of correcting human mistakes, I would need to correct the mistakes of a machine. However, after having gained some experience my view has changed. Now I see MTPE more like a regular translation task, where in addition to TM matches and other useful resources, I have at my disposal suggestions from the MT. I am free to delete them and retranslate the segment from scratch if I think they are not useful. While in proofreading, I just correct someone else's translation, in MTPE I am the author of the final translation product and I am fully free to create it the way I choose.

The most difficult part of MTPE, in my opinion, is to decide when it is better to use a segment partially or in full and when to re-translate it from scratch. At first it takes time, but it is a matter of practice: right now it takes me only a couple of seconds to decide whether I should or should not correct a particular MT segment.

I specifically enjoy being able to spot and "fix" the MT errors that I spent the most time correcting. All the linguists who work on post-editing jobs for TransPerfect report back to our MT developers feedback and inform them of the frequent and systematic MT errors they would like to be fixed. Their feedback is then implemented in the MT system. In this way, the post-editing time is continuously decreasing. This feedback is the most efficient way to improve the systems. Providing useful feedback is not so easy at first, one has to understand how the system works and what kind of feedback can be implemented. In addition, one has to have an analytical mindset, be able to identify patterns and systematic errors and generalize. This is a skill that can be acquired and improved with practice.

For me, this is the most fascinating aspect of working with MT. I like seeing how the system produces a better output each time and takes into account the feedback I have provided. I like being a part of the developments in MT and other Artificial Intelligence applications for language, as I believe it has great potential to make our way of working more interesting.

Our profession is constantly evolving thanks to the emergence of new technologies. One of them is neural MT and we can already observe how it influences the way we perform post-editing. These systems are different in the way they function and the type of errors they make. While providing improved fluency, they are prone to committing errors that are not very common for phrase-based systems, such as word omissions. That is why it is important for linguists to be aware what systems they are using, keep track of the latest developments and have the necessary expertise.

Efficient work and constant collaboration with linguists is essential for both MT development and testing, i.e. for successful MT implementation. Our internal linguists are MT experts and all of them have gone through extensive training on MT technologies and post-editing. Training and preparation of linguists is as important as taking into consideration their suggestions for improvement of the MT workflow and the MT quality.

Project/product descriptions

OctaveMT: Putting Three Birds into One Cage

Juan A. Alonso
Lucy Software Ibérica (ULG Group)
Copèrnic 42-44 1r
08021 Barcelona, Spain
juan.alonso@ulgroup.com

Albert Llorens
Lucy Software Ibérica (ULG Group)
Copèrnic 42-44 1r
08021 Barcelona, Spain
albert.llorens@ulgroup.com

Abstract

This product presentation describes the integration of the three MT technologies currently used – rule-based (RBMT), Statistical (SMT) and Neural (NMT) – into one scalable single platform, OctaveMT. MT clients can access all three types of MT engines, whether on a user specified basis or depending on several translation parameters (language-direction, domain, etc.)

1 Introduction

Historically, Lucy Software and Services (a company of the United Language Group) has been focusing its development efforts on its RBMT system. However, during the last few years, we started to develop and use SMT technology and during the last months we have also been working on the NMT area. Our mid-term goal is to have an operational RBMT–NMT hybrid engine.

The aim of this presentation is to introduce and describe the integration of all three MT technologies into one single product platform, OctaveMT.

2 System Architecture

The system architecture is depicted in Figure 1. The platform keystone is the LT Task Scheduler component, a portable and scalable task distribution system offering high performance for many kinds of services. It accepts translation requests from one or more MT Clients through a RESTful API. These translation requests are stored in the Task Pool component of the Task Scheduler.

The translation tasks are then handled by one or more MT engines. Each engine has an eServant component that monitors its activity; when it is idle, it fetches one request from the Task Pool.

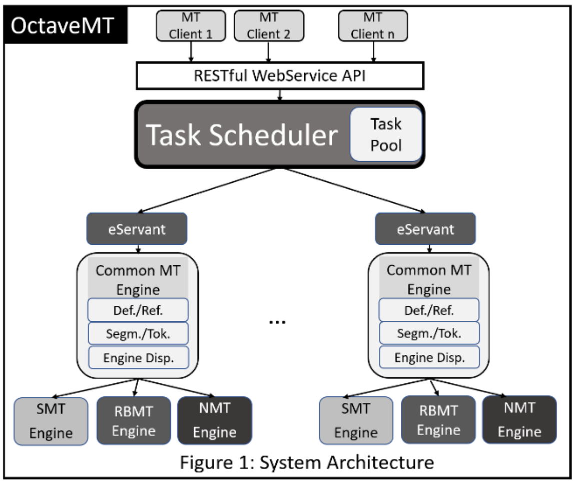


Figure 1: System Architecture

This task is then fed through the deformatter, the segmenter, the tokenizer and, finally, the engine dispatcher. The dispatcher sends the segmented text to the back-end engine type specified in the translation task (RBMT, SMT or NMT). After that, the translated text is sent back to the reformatter, and finally delivered to the originator MT Client through the Task Scheduler.

3 Advantages of this Approach

By re-using common sub-components for the three types of translation engines, tasks such as document format handling and conversion, which typically are a problem for raw SMT & NMT engines, can be properly handled. Additionally, this approach allows to use standard load-balancing techniques to build distributed high-performance MT infrastructures.

TransPerfect's Private Neural Machine Translation Portal

Diego Bartolomé, José Masa
TransPerfect
Passeig de Gràcia 11, Esc. B 5è 2a
08007 Barcelona, Spain
{dbartolome,jmasa}@translations.com

Abstract

We will present our solution to replace the usage of publicly available machine translation (MT) services in companies where privacy and confidentiality are key. Our MT portal can translate across a variety of languages using neural machine translation, and supports an extensive number of file types. Corporations are using it to enable multilingual communication everywhere.

1 Introduction

Machine translation (MT) is widespread today[1]. Companies are using it extensively both for productivity increase and thus turnaround time and cost reduction, and also for gisting or understandability in many situation such as e-discovery. At TransPerfect, we have developed a neural machine translation platform that can be installed on premises or on our own cloud to guarantee data confidentiality and control, link client-specific neural MT engines to it, and enable supervised and unsupervised learning[2].

2 Access to the platform

The access is through a URL (to be presented at the conference), and can be customized for each client. Our main features are:

- **Single Sign On**: no need for specific usernames or passwords, users at our clients can access with their company e-mail and password.
- **IP address range restriction**: only users accessing through a pre-defined range of IP addresses are allowed into the system. This is essential for security in our top clients like banks or pharma companies.
- **Real-time translation of plain text and documents**: users can translate plain text and also more than 40 file types, including scanned PDFs and Office documents.
- **Neural MT engines**: neural MT engines are available in more than 25 languages, with a supervised and unsupervised learning option. Supervised means that the engines learn from linguists' feedback, and unsupervised refers to self-learning capabilities. A functionality to suggest a better translation is available, as well as automated language detection.
- **Reporting**: powerful reporting is available to enable real-time tracking of number of processed words, quality of the engines, and other business KPIs.
- **Data storage**: we delete data after 24 hours, and some clients have even more restrictive policies to delete translated plain text immediately and documents after they are downloaded.

3 Additional features

Besides the above, we are currently integrating additional features that have been commonly requested such as customization of glossaries and do not translate lists, seamless integration with our human post-editing services, and addition of speech-to-text and text-to-speech as input and output modes, respectively.

References

[1] TransPerfect, *The Year of Artificial Intelligence in Translation*, http://www.transperfect.com/blog/the-year-of-AI-translation.

[2] TransPerfect, *MT: Robot Intelligence Technology with a Human Touch*, TransPerfect blog, http://www.transperfect.com/blog/machine-

Terminology validation for MT output

Giorgio Bernardinello
STAR Group
Wiesholz 35, 8262 Ramsen
Switzerland
giorgio.bernardinello@star-group.net

Abstract

WebTerm Connector is a plugin for STAR MT Translate which combines machine translation with validated terminology information. The aim is to provide "understandable" information in the target language using corporate language and terminology.

1 WebTerm

WebTerm is STAR's web-based terminology management system that can be seamlessly integrated into STAR MT Translate in order to search for terms and display terminology information.

2 STAR MT Translate

STAR MT Translate is STAR's web-based application for MT systems that can also be integrated into Microsoft Office products. STAR MT engines are based on customer-specific translations and terminology. As additional feature, the web-based application can access the company's Transit translation memory.

3 WebTerm Connector

The Web Term Connector plugin allows users to search for translations of terms in the company's dictionaries within the STAR MT Translate environment. Furthermore, it is possible to display additional terminology information for the source and the target language terms, if this is available in the dictionaries.

3.1 Searching for terms

On the assumption that the company dictionaries contain validated terminology, the translation is searched for in the dictionaries first. If no translation is found, especially for longer strings and sentences, the TM is then checked for perfect matches. If none are found, the MT engine will provide the translation.

3.2 Information about terms

If more information on parts of the source or the target sentence is needed, the user can simply highlight the text to search for in the company-specific dictionaries. If the word is found, its translation and the available terminology information are displayed at the bottom of the same web page. The customer can define what information from the dictionaries should be shown.

The major advantage of this solution is that users can quickly access company terminology without having to switch between browser tabs.

3.3 Translation of terminology information

Sometimes the dictionaries contain definitions or context descriptions in just in one language, e.g. the target language. But the user might need them in the source language in order to understand them, which is why MT technology is also integrated into the terminology area. This makes it possible to obtain an automatic translation of text parts of the dictionary simply by clicking the "Translate" button there.

3.4 Improved term searching functionality

All strings that were highlighted for terminology searching but for which no match was found in the dictionaries are listed in a log file. The customer can then use this information to improve not only their WebTerm dictionaries but also their MT engines.

The ModernMT Project

Nicola Bertoldi **Davide Caroselli** **Marcello Federico**

MMT Srl - Trento, Italy
www.modernmt.eu

Abstract

This short presentation introduces ModernMT: an open-source project [1] that integrates real-time adaptive neural machine translation into a single easy-to-use product.

1 Neural Machine Translation

Neural machine translation (MT) technology has been widely adopted by the translation industry, to produce ready-to-use drafts or high quality translations via post-editing. Deployed engines are either generic, such as Google Translate, or customized, like the MT@EC engine, which is particularly suited to the EU policy documents. Generic MT works well on average, but it can be bad at handling domain specific terminology or style. On the other hand, custom MT can definitely be more accurate but does not scale. Actually, for the translation industry the granularity of a domain can be very fine – i.e. specific customers might use terminology in their own original way – thus, ending up with thousands of domains. Customization becomes quickly unfeasible, for both computational and infrastructure costs. This is where ModernMT comes in!

2 Adaptive NeuralMT

ModernMT is a new open-source MT software that consolidates the current state-of-the-art MT technology into a single and easy-to-use product. ModernMT adapts to the context in real-time and is capable of learning from and evolving through interaction with users, with the final aim of increasing MT-output utility for the translator in a real professional environment. This is achieved by augmenting a generic neural MT system with an internal dynamic memory, storing all available user translation memories (TMs). This process is completely transparent when users work with a CAT tool, like MateCat (Federico et al., 2014). The ModernMT memory is kept in sync with the user TMs. When ModernMT receives a translation query, it quickly analyses its context, recalls from its memory the most related translation examples, and instantly adapts its neural network to the query.

3 Performance

Our adaptation approach (Farajian et al., 2017) has proven to deliver the translation quality of customized machine translation at the maintenance costs of a generic system. Thanks to careful optimization, the whole adaptation process, which is run for every sentence, just takes a fraction of second. Further, experiments have also shown that our context-based and memory-based adaptation method impacts positively on the translation of terminology.

4 Software as a Service

ModernMT is also provided as a SaaS solution for enterprises. Benefits include cutting-edge innovations, baseline models trained on billions of words of premium data, support for nine language pairs (and more to come). Finally, professional translators can instead benefit from all the advantages of ModernMT in MateCat[2], a free CAT tool that is perfectly integrated with our service.

References

Federico M. et al. 2014. *The MATECAT Tool*, Proc. COLING, Dublin, Ireland.

Farajian M. A. et al. 2017. *Multi-domain Neural Machine Translation through Unsupervised Adaptation*, Proc. WMT, Copenhagen, Denmark.

[1] https://github.com/ModernMT/MMT.

[2] www.matecat.com

Developing a New Swiss Research Centre for Barrier-Free Communication

Pierrette Bouillon, Silvia Rodríguez Vázquez, Irene Strasly
Department of Translation Technology, FTI, University of Geneva, Switzerland
{pierrette.bouillon|silvia.rodriguez|irene.strasly}unige.ch

Abstract

The project 'Proposal and Implementation of a Swiss Research Centre for Barrier-free Communication' (BFC) is a four-year project (2017–2020) funded by the Rectors' Conference of Swiss Higher Education Institutions (*swissuniversities*).[1] Its purpose is to ensure that individuals with a visual or hearing disability, people with a temporary cognitive impairment and speakers without sufficient knowledge of local languages can communicate and enjoy barrier-free access to information in all spheres of life, with a special focus on higher education.

1 The Project

In Switzerland, the principles of equality, non-discrimination and social inclusion are advocated at a federal and cantonal level. However, educational qualification rates continue to be extremely low among hearing and visually impaired individuals in particular, with numbers below 1% in the case of tertiary education. The work conducted within the framework of the Barrier-free Communication (BFC) project, a joint effort between the Zurich University of Applied Sciences and the University of Geneva, aims at developing new guidance and technological resources for teaching and administrative staff of higher institutions with disabilities. A total of ten research areas revolving around language resources and technology have been defined. These include, among others, audio description, life subtitling, easy-to-read and plain language, multilingual web accessibility and speech to sign language translation. In the context of this last research area, a use case has been developed in collaboration with the Geneva University Hospitals (HUG).

2 BabelDr: A speech to sign translation system for anamnesis

Today, hospitals have to increasingly deal with patients who have no language in common with the staff. BabelDr (babeldr.unige.ch) was elaborated to specifically address this issue. The system can be characterized as a flexible speech-enabled fixed-phrase translator (Bouillon et al., 2017). The set of sentences are limited, but the user can speak freely, which improves usability. As in a translation memory, the system will map the doctor's question to the closest match, using different matching techniques. The key features are: (i) security (data are stored locally), (ii) reliability (translations have been done by humans or interpreters with an on-line platform), and (iii) flexibility (source content and translations can be easily added, in different formats, written or oral/aural). The actual sign language version was developed for LSF-CH (Swiss-French Sign Language) in collaboration with a deaf nurse and a professional hearing sign language interpreter in professional conditions. BabelDr will be tested in real settings at HUG in the summer. The existing version for spoken languages contains 7 domains which cover the most frequent health issues, with around 3500 sentences per domain.

References

Bouillon, Pierrette *et al.* 2017. BabelDr vs Google Translate: a user study at Geneva University Hospitals (HUG). *20th Annual Conference of the European Association for Machine Translation (EAMT)*. Prague, Czech Republic 47–52.

[1] https://www.swissuniversities.ch/en/organisation/

Massively multilingual accessible audioguides via cell phones

Itziar Cortes
and Igor Leturia
Elhuyar Fundazioa
i.leturia@elhuyar.eus

Iñaki Alegria, Aitzol Astigarraga
and Kepa Sarasola
Faculty of Informatics
University of the Basque Country
(UPV/EHU)
i.alegria@ehu.eus

Manex Garaio
PuntuEUS Fundazioa
manex@domeinuak.eus

Abstract

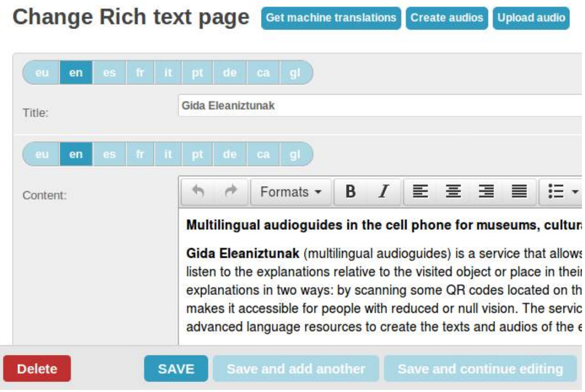
Bidaide[1] is a web service that allows the visitors of a museum, route or building to read or listen to explanations relative to the visited place on their own mobile and in their own language. The visitor can access the explanations in various ways: by scanning some QR codes located in the place, by GPS positioning (in outdoor routes), or by automatic Bluetooth proximity activation. This makes it accessible for people with reduced or null vision. On the other hand, this platform also offers to the manager of the visited site the most advanced language resources to create the texts and audios of the explanations in many languages.

In museums, train stations, airports, etc., travellers have to read many messages that usually are at most offered in four languages. But if we consider that practically all travellers carry their own mobile phone, why not offer them this content in 20 languages? The texts corresponding to all those languages do not all have to be physically present on the poster (this way the poster would be confusing and difficult to assimilate).

QRpedia[2] is a similar mobile web-based system which uses QR codes to deliver contents to users, in their preferred language, but just Wikipedia articles.

The *Bidaide* web platform allows the manager of a museum, route or building to easily create text and audio contents in many languages by means of machine translation and speech synthesis. Once the text in a language is created, you just have to press the buttons "Get machine translations" and "Create audios" to obtain the translations and audios. There is also the option of post-editing or manually translating the texts, and the option of recording the audios. You can also use automatic technologies for some languages and do it manually for others.

The application was created by Elhuyar and Donostia 2016 European Capital of Culture foundations, and the University of the Basque Country. It is based on *Ohar eleanitzak* (Garaio, 2014), an open-source application used in the Albaola Museum and in the events organised by Donostia 2016 (Agerri et al., 2017). Elhuyar Foundation improved this multi-lingual solution making it accessible for people with reduced or null vision. It is installed in various museums, touristic routes and public buildings in the Basque Country[3].



Title: Gida Eleaniztunak

Content:

Multilingual audioguides in the cell phone for museums, cultur

Gida Eleaniztunak (multilingual audioguides) is a service that allows
listen to the explanations relative to the visited object or place in thei
explanations in two ways: by scanning some QR codes located on th
makes it accessible for people with reduced or null vision. The servic
advanced language resources to create the texts and audios of the e

Figure 1: Creation and management of contents

References

Agerri, Rodrigo, Aitzol Astigarraga, Iñaki Alegria, Itziar Cortes, Arantza Diaz de Ilarraza, Igor Leturia, and Kepa Sarasola. 2017. L*anguage Technology projects in the frame of the European Capital of Culture 2016*. META-Forum 2017.

Garaio, Manex. 2014. *QR kodeak eta eduki-kudeaketa eleanitza*. Univ. of the Basque Country UPV/EHU.

[1] http://bidaide.elhuyar.eus

[2] https://www.learntechlib.org/p/182074/

[3] http://gidaeleaniztunak.elhuyar.eus/adibide-arrakastatsuak/

ELRI
European Language Resource Infrastructure

Thierry Etchegoyhen,[1] Borja Anza Porras,[1] Andoni Azpeitia,[1] Eva Martínez Garcia,[1]
Paulo Vale,[2] José Luis Fonseca,[2] Teresa Lynn,[3] Jane Dunne,[3]
Federico Gaspari,[3] Andy Way,[3] Victoria Arranz,[4] Khalid Choukri,[4]
Vladimir Popescu,[4] Pedro Neiva,[5] Rui Neto,[5] Maite Melero,[6]
David Perez,[6] António Branco,[7] Ruben Branco,[7] Luís Gomes[7]

[1] Vicomtech, Spain - {tetchegoyhen, banza, aazpeitia}@vicomtech.org
[2] AMA, Portugal - {paulo.vale, jose.fonseca}@ama.pt
[3] DCU, Ireland - {teresa.lynn, jane.dunne, federico.gaspari, andy.way}@adaptcentre.ie
[4] ELDA, France - {arranz, choukri, vladimir}@elda.org
[5] Linkare, Portugal - {pneiva, rneto}@linkare.com
[6] SESIAD, Spain - maite.melero@upf.edu, dperezf@minetad.es
[7] University of Lisboa, Portugal - {antonio.branco, ruben.branco, luis.gomes}@di.fc.ul.pt

Abstract

We describe the European Language Resources Infrastructure project, whose main aim is the provision of an infrastructure to help collect, prepare and share language resources that can in turn improve translation services in Europe.

1 Description

The European Language Resources Infrastructure (ELRI) project is an initiative co-funded by the European Union under the Connecting Europe Facility programme, under Grant Agreement INEA/CEF/ICT/A2016/1330962. ELRI has a duration of 24 months and started in October 2017.

2 Objectives

The main objective of ELRI is the provision of an infrastructure to help collect, prepare and share language resources that can in turn improve translation services. In particular, resources shared with the DGT will contribute to improve the EU automated translation services that are freely available to all public institutions.

The initiative notably addresses current issues related to sharing resources directly at the European level or beyond, by providing National Relay Stations where resources remain under member states' laws and regulations until further clearance is negotiated and granted.

ELRI targets resources that are relevant to Digital Service Infrastructures and currently involves public institutions and public translation centres in France, Ireland, Portugal and Spain, with a future extension to additional member states as a key objective beyond the current action.

3 Benefits

ELRI provides the following main benefits:

- The provision of separate data sharing layers at the national, European, and community levels, ensures compliance with the relevant sharing restrictions at every step.
- Raw language resources are converted automatically into a format useful for translation experts and machine translation systems.
- ELRI provides broad compliance verification covering intellectual property rights, the Public Sector Information directive and DSI-specific needs.
- Language resources can be shared as deemed appropriate by stakeholders.
- Registered users of the national relay stations can benefit from the automatically created translation memories.
- The European Union's eTranslation services will benefit from the collected and prepared language resources that have been authorised for sharing at the European level.

4 Acknowledgements

Co-financed by the European Union
Connecting Europe Facility

The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.

The SUMMA Platform: Scalable Understanding of Multilingual Media

Ulrich Germann,[♦] Peggy van der Kreeft,[♠] Guntis Barzdins,[♣♥] Alexandra Birch[♦]

[♦] University of Edinburgh; [♠] Deutsche Welle; [♣] LETA; [♥] University of Latvia

for the SUMMA Consortium[*]

corresponding authors: `ugermann@ed.ac.uk`, `peggy.van-der-kreeft@dw.com`

Abstract

We present the latest version of the SUMMA platform, an open-source software platform for monitoring and interpreting multi-lingual media, from written news published on the internet to live media broadcasts via satellite or internet streaming.

1 Introduction

The SUMMA platform is a highly scalable open-source infrastructure for monitoring and interpreting news streams in multiple languages,[1] and a variety of media formats, from written text published on the internet to live TV broadcasts via satellite.

Three use cases drive the project.

External Media Monitoring

BBC Monitoring (BBCM) is a business unit within the BBC tasked with monitoring and digesting international news broadcasts and other media as an internal service to the BBC as well as a paid service to outside customers.

The SUMMA platform will allow BBCM's staff journalists to widen their monitoring coverage and focus on news interpretation and analysis by alleviating them from mundane monitoring tasks.

Internal Monitoring

Deutsche Welle (DW) is an international broadcaster covering world-wide news in 30 different languages. Regional news rooms produce and broadcast content independently. Monitoring DW's output with the SUMMA platform will enable DW as an organisation to better keep track of its own output and determine which stories have been covered where, and where there are gaps in the coverage.

Data Journalism

The SUMMA database will give journalists access to many thousands of news stories with additional metadata such as named entity tags provided by SUMMA's NLP processing modules, providing for large-scale analysis of the constantly evolving news landscape.

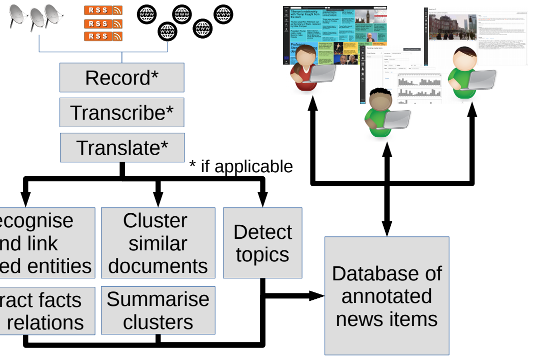

Record*
Transcribe*
Translate*
* if applicable
Recognise and link named entities
Cluster similar documents
Detect topics
Extract facts and relations
Summarise clusters
Database of annotated news items

Figure 1: The SUMMA Platform Architecture

2 Architecture

The design of the SUMMA platform is shown in Fig. 1. Incoming media streams are downloaded and/or recorded, depending on the source. Audio is automatically transcribed, and non-English material is machine-translated into English. The resulting text-based news items are then processed with downstream NLP modules: topic detection; named entity recognition and linking, and extraction of relations between named entities to build up a knowledge base of "facts" (i.e., factual claims made in news reporting); and document clustering and multi-document cluster summarization.

News items, mentions of named entities, etc., are stored in a central database that can be accessed by users via web-browser-based user interfaces, or by programs via programmatic interfaces (APIs).

Acknowledgements

This work was conducted within the scope of the Research and Innovation Action *SUMMA*, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688139.

[*] The SUMMA Consortium comprises the University of Edinburgh, LETA, Idiap Research Institute, Priberam, Qatar Computing Research Institute, University College London, and Sheffield University as research partners, and the BBC and Deutsche Welle as use case partners.

[1] Arabic, German, English, Farsi,* Latvian,* Portuguese,* Russian, Spanish, Ukranian* (* planned for late 2018)

Smart Pre- and Post-Processing for STAR MT Translate

Judith Klein
STAR Group
Wiesholz 35, 8262 Ramsen
Switzerland
Judith.Klein@star-group.net

Abstract

After many successful experiments it has become evident that smart pre- and post-processing can significantly improve the output of neural machine translation. Therefore, various generic and language-specific processes are applied to the training corpus, the user input and the MT output for STAR MT Translate.

1 STAR MT Translate

STAR MT Translate is STAR's web-based MT system which can also be integrated in Microsoft Office products using the STAR MT Office Connector. In this MT application all data is kept safe within the customer's environment.

The aim is to provide "useful" translations for customers in their company-specific domain, e.g. transportation. Typically, the users are not translators but various professionals, e.g. mechanics, who need to understand the information that is only available in a foreign language.

In professional translation projects mostly structured text is translated, possibly supported by MT. The user input to STAR MT Translate however is much more flexible and "unpredictable". It contains customer-specific terminology but it consists of a wide range of linguistic phenomena, including ungrammatical sequences.

To meet both kinds of text, firstly, the core of the engines are built from customer-specific in-domain translations; secondly, to deal with the variety of language usage, the engines are complemented by out-of-domain data, and they use neural technology.

2 Smart Pre- and Post-processing

Good training data is the essential requisite to obtain good MT. It must cover the language phenomena that are likely to occur in the user input sent to the MT system for translation.

Even if the corpus includes the required characteristics it usually also contains "noise" that considerably reduces the quality of the MT output.

Therefore, STAR has developed a systematic strategy to identify and delete this "noise".

Firstly, language-independent processing rules delete incomplete formatting or punctuation, irrelevant characters, fragmentary sentences etc. Nominalizations of various, inconsistent number formats (dates, decimal separators, etc.) as well as URL and email addresses are defined in another step. A specific set of rules ensures that the corpora contain a balanced amount of similar sentences (regarding the length of sentences, the number of tokens, etc.) and determine the prioritization of segments depending on their completeness.

Finally, language-specific processes are applied that identify irrelevant text, e.g. typos, foreign language, informal expressions, and – depending on the language – include a special handling of morphological variants.

The same generic processing steps and the adequate language-specific rules are applied to the input text, in order to send to the MT engine sentences that are made up in the same way as the ones it has learned.

And finally, post-processing uses these rules, too, in order to obtain high-quality MT output.

`mtrain`: A Convenience Tool for Machine Translation

Samuel Läubli* and **Mathias Müller*** and **Beat Horat** and **Martin Volk**
{laeubli,mmueller,horat,volk}@cl.uzh.ch
Institute of Computational Linguistics
University of Zurich

Abstract

We present `mtrain`, a convenience tool for machine translation. It wraps existing machine translation libraries and scripts to ease their use. `mtrain` is written purely in Python 3, well-documented, and freely available.[1]

Machine translation libraries usually focus on core model training, while data preparation and automatic evaluation are left to the user. This presents a barrier to experimental reproducibility, rapid prototyping, and entry to the field from neighbouring disciplines. In the spirit of the Experimental Management System for Moses (Koehn, 2010), our tool is meant to automate these tasks.

`mtrain` is designed to handle most aspects of a machine translation experiment: it manages preprocessing, model training, and automatic evaluation. Preprocessing involves automatically splitting a data set into training, validation, and test sets; tokenization; casing; byte-pair encoding; and normalization. On top of these standard preprocessing steps, `mtrain` can also deal with inline XML markup and intelligently transfer XML tags to translations (Müller, 2017).

Our tool provides training automation for statistical phrase-based models with Moses (Koehn et al., 2007) and neural RNN encoder-decoder models with Nematus (Sennrich et al., 2017). After training, `mtrain` offers automatic evaluation of translation quality. It outputs the well-known BLEU, TER, and METEOR metrics (Clark et al., 2011). Given a folder that contains trained models, the separate component `mtrans` can be used to translate from files or standard input.

All steps can be configured with config files or command line options, but default settings already lead to functional baseline systems, making it easier for inexperienced users to use the tool. Going forward, we consider wrapping additional machine translation libraries that are native Python 3, such as Sockeye (Hieber et al., 2017).

References

Clark, Jonathan H, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL*, pages 176–181.

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: a toolkit for neural machine translation. *arXiv preprint*, arXiv:0902.0885.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180.

Koehn, Philipp. 2010. An experimental management system. *The Prague Bulletin of Mathematical Linguistics*, 94:87–96.

Müller, Mathias. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of EACL*, pages 65–68.

[1] https://github.com/ZurichNLP/mtrain

*equal contribution

Empowering Translators with MTradumàtica: A Do-It-Yourself statistical machine translation platform

Adrià Martín-Mor
Universitat Autònoma de Barcelona
K-1020
Campus UAB
08193 Barcelona (Catalan Republic*)
adria.martin@uab.cat

Pilar Sánchez-Gijón
Universitat Autònoma de Barcelona
K-2002
Campus UAB
08193 Barcelona (Catalan Republic*)
pilar.sanchez.gijon@uab.cat

Abstract[1]

According to Torres Hostench et al. (2016), the use of machine translation (MT) in Catalan and Spanish translation companies is low. Based on these results, the Tradumàtica research group,[2] through the ProjecTA and ProjecTA-U projects,[3] set to bring MT and translators closer with a two-fold strategy. On the one hand, by developing MTradumàtica, a free Moses-based web platform with graphical user interface (GUI) for statistical machine translation (SMT) trainers. On the other hand, by including MT-related contents in translators' training. This paper will describe the latest developments in MTradumàtica.

1 Introduction

Currently, the development of MTradumàtica —developed as an experimental platform for SMT trainers (Martín-Mor, 2017)—, focuses on functionalities that might be needed by translators and translation companies: TMX processing, user management and integration with CAT tools.

1.1 TMX processing

TMX is a well-known bilingual format among translators, who store their translations in translation memories (TM). By allowing the processing of TMX files, users will be able to upload their own TMs to MTradumàtica in order to train SMT systems.

1.2 User management

In a near future, an authentication protocol will be implemented. This means that users will log in to MTradumàtica and will have exclusive access to their texts and engines. A permissions' system will also be implemented in order to grant or protect access to specific features.

1.3 Integration with CAT tools

CAT tools typically allow the integration with MT systems. OmegaT, since version 4.1.3u2, allows users to connect with a customised Moses engine.[4] In order to simplify the process of integrating MTradumàtica into CAT tools, a new feature protected with password will allow users to generate URLs for the desired engines. The users will only need to paste these URLs in order to get MT matches from the CAT tool.

2 Concluding remarks

This paper presented the latest developments in Mtradumàtica,[5] conceived primarily as an experimental Do-It-Yourself SMT platform for translation trainers, from which also professional translator might benefit.

References

Martín-Mor, Adrià (2017). MTradumàtica: Statistical machine translation customisation for translators. *Skase*, *11*(1), 25–40.

Torres Hostench, Olga *et al.* (2016). L'ús de traducció automàtica i postedició a les empreses de serveis lingüístics de l'Estat espanyol. Retrieved from https://ddd.uab.cat/record/166753.

[2] www.tradumatica.net.

[3] Funded by the Ministerio de Economía y Competitividad of the Spanish government (Ref: FFI2013-46041-R and FFI2016-78612-R). www.projecta.tradumatica.net.

[4] http://blogs.uab.cat/tradumatica/2018/01/.

[5] Source code at www.github.com/tradumatica; demo version at www.m.tradumatica.net.

* This article is signed, as citizens of the Catalan Republic proclaimed by the Parliament of Catalonia, in protest against the imprisonment of political activists and members of the Catalan government and in solidarity with all the citizens who suffered reprisals by the Spanish state following the Catalan self-determination referendum held on October the 1st, 2017.

Speech Translation Systems as a Solution for a Wireless Earpiece

Nicholas Ruiz, Andrew Ochoa, Jainam Shah, William Goethels, Sergio DelRio Diaz
Waverly Labs, Brooklyn, NY, USA
{nick,andrew,jainam,william,sergio}@waverlylabs.com

Abstract

The advances of deep learning approaches in automatic speech recognition (ASR) and machine translation (MT) have allowed for levels of accuracy that move speech translation closer to being a commercially viable alternative interpretation solution. In addition, recent improvements in micro-electronic mechanical systems, microphone arrays, speech processing software, and wireless technology have enabled speech recognition software to capture higher quality speech input from wireless earpiece products. With this in mind, we introduce and present a wearable speech translation tool called Pilot, which uses these systems to translate language spoken within the proximity of a user wearing the wireless earpiece.

1 What is Pilot?

The Pilot Translating Earpiece is a sophisticated earbud which uses dual microphones and custom noise cancelling algorithms to produce clear speech before it is passed through our mobile app and to our speech translation engine in the cloud. It relays speech translation very quickly with minimal latency. Pilot consists of two translation earbuds that pair with custom speech translation software for Android or iOS. Pilot allows consumers to share their secondary earbud with a conversation partner for face-to-face simultaneous speech translation and currently supports 15 languages[1].

2 How Does Pilot Work?

Pilot operates in two modes: *Converse* and *Listen*. As the primary use case, *Converse* mode allows multi-party conversations with transcriptions logged in the app. In a one-on-one conversation, users can share their secondary earbud with a partner and quickly pair it with the partner's phone. Currently in beta, *Listen* mode adapts the microphone firmware settings to pick up ambient sound and performs far-field ASR and MT. Pilot uses several speech translation paradigms, depending on the language pair, either by running ASR and MT sequentially, or as tightly coupled speech translation[2]. Translations are primarily run on the server, while the app is responsible for routing the audio to and from the earpiece.

Practical challenges *Bluetooth*: Conventionally, Android and iOS devices are limited to one microphone connection at a time. Although routing the partner's earbud recordings through the same phone is possible, it requires low-level kernel programming to implement. Our team will resolve this issue in a future release. *Microphone pick-up*: Occasionally a conversation partner's speech can be picked up by the user's earbud, and vice-versa. While digital signal processing can eliminate some of this effect, the position, distance, and power of the speech must be taken into account.

As the provider of one of the first translation wearables to market, we are eager to how learn translation technology affects situational dialogue without an interpreter present. While our first version pieces maturing technologies together, we are working on improving the user experience by minimizing user's dependence on their phone's screen.

[1] Arabic, Chinese (Mandarin, Cantonese), English, French, German, Hindi, Italian, Japanese, Korean, Portuguese, Russian, Spanish, Greek, Turkish, and Polish ASR and MT.

[2] Speech synthesis is currently not informed by ASR or MT.

Multi-modal Context Modelling for Machine Translation

Lucia Specia
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, S1 4DP
Sheffield, UK
l.specia@sheffield.ac.uk

Abstract

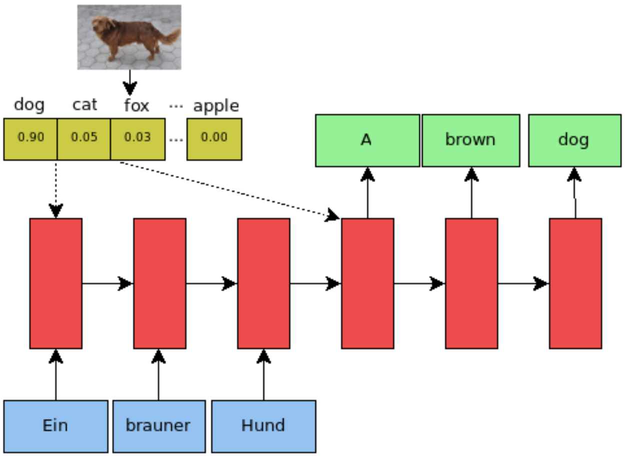
MultiMT is an European Research Council Starting Grant whose aim is to devise data, methods and algorithms to exploit multi-modal information (images, audio, metadata) for context modelling in machine translation and other cross-lingual tasks. The project draws upon different research fields including natural language processing, computer vision, speech processing and machine learning.

1 Description

Human translators have access to a number of contextual cues beyond the actual segment to translate when performing translation, for example, images associated with the text. Machine translation approaches, however, have historically disregarded any form of non-textual context and make little or no reference to wider surrounding textual content. This results in translations that miss relevant information or convey incorrect meaning. Such issues drastically affect reading comprehension and may render translations less useful. One example is the word ‘seal’ in the sentence ‘The man is holding a seal’. When translating to German, this sentence can become ‘Ein Mann hält ein Siegel’ or ‘Ein Mann hält einen Seehund’. Pictures such as the ones below could help in making this decision:

With an emphasis on images as additional modality and drawing parallels to work on image captioning, thus far the project has mainly targeted four main lines of research: data acquisition, representations, models and evaluation. For data acquisition, we have been following two approaches: (i) making multimodal data multilingual, where English image description datasets is extended to include translations of the descriptions in multiple languages, and (ii) making multilingual data multimodal, where parallel data is complemented by visual representations.

For representations, we have been exploiting high-level, abstract representations, such as the presence and frequency of objects in images, rather than relying on low-level, dense representations. We show that these representations are effective in both image captioning and machine translation. Our models are extensions of sequence-to-sequence neural models where different modalities can complement parallel text in different ways:



Project website: https://multimt.github.io/

Project PiPeNovel: Pilot on Post-editing Novels

Antonio Toral, Martijn Wieling
Center for Language and Cognition
Faculty of Arts
University of Groningen, The Netherlands
{a.toral.ruiz,m.b.wieling}@rug.nl

Sheila Castilho, Joss Moorkens, Andy Way
ADAPT Centre
School of Computing
Dublin City University, Ireland
firstname.secondname@adaptcentre.ie

Abstract

Given (i) the rise of a new paradigm to machine translation based on neural networks that results in more fluent and less literal output than previous models and (ii) the maturity of machine-assisted translation via post-editing in industry, project PiPeNovel studies the feasibility of the post-editing workflow for literary text conducting experiments with professional literary translators.

Machine translation (MT) has progressed enormously over the last years and it is widely used nowadays for gisting purposes. However, its use in professional translation is still largely confined to the post-editing of technical and legislative text. The aim of PiPeNovel is to carry out a pilot study to assess the feasibility of broadening the use of the post-editing workflow to literary text, in particular to novels. The translation direction covered in the project is English-to-Catalan. Now PiPeNovel is about to finish and we present the three main activities conducted in the project:

(1) MT. First, we built a literary-adapted neural MT (NMT) system and evaluated it against a system pertaining to the previous dominant paradigm in MT: statistical phrase-based MT (PBSMT) (Toral and Way, 2018). Both systems were trained on over 1,000 novels. We conducted a human evaluation on three novels by Orwell, Rowling and Salinger; between 17% and 34% of the translations, depending on the book, produced by NMT (versus 8% and 20% with PBSMT) were perceived by native speakers of the target language to be of equivalent quality to translations produced by a professional human translator.

(2) Post-editing effort. Subsequently, using these MT systems, we conducted a post-editing study with six professional literary translators on a fantasy novel (Toral et al., 2018). We analysed temporal effort and found that both MT approaches result in increases in translation productivity: PBMT by 18%, and NMT by 36%. Post-editing also led to reductions in the number of keystrokes (technical effort): by 9% with PBMT, and by 23% with NMT. Finally, regarding cognitive effort, post-editing resulted in fewer (29% and 42% less with PBMT and NMT respectively) but longer pauses (14% and 25%).

(3) Translators' perceptions. Finally, we analysed the perceptions of the translators that took part in the post-editing experiment (Moorkens et al., 2018), which were collected via questionnaires and a debrief session. While, as stated before, all participants were faster when post-editing NMT, they all still stated a preference for translation from scratch, as they felt less constrained and could be more creative. When comparing MT systems, participants found NMT output to be more fluent and adequate.

Acknowledgements

PiPeNovel is funded by the European Association for Machine Translation through its 2015 sponsorship of activities programme. The ADAPT Centre at Dublin City University is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106).

References

Moorkens, Joss, Antonio Toral, Sheila Castilho and Andy Way. 2018. Perceptions of Literary Post-editing using Statistical and Neural Machine Translation. *Translation Spaces* (under review).

Toral, Antonio and Andy Way. 2018. What Level of Quality can Neural Machine Translation Attain on Literary Text? In *Translation Quality Assessment*. Springer (in press).

Toral, Antonio, Martijn Wieling, and Andy Way. 2018. Post-editing Effort of a Novel with Statistical and Neural Machine Translation. *Frontiers in Digital Humanities* (in press).

Smart Computer-Aided Translation Environment (SCATE): Highlights

Vincent Vandeghinste
Tom Vanallemeersch
Bram Bulté
Liesbeth Augustinus
Frank Van Eynde
Joris Pelemans
Lyan Verwimp
Patrick Wambacq
Geert Heyman
Marie-Francine Moens
Iulianna van der Lek-Ciudin
Frieda Steurs
KU Leuven
first.lastname@kuleuven.be

Ayla Rigouts Terryn
Els Lefever
Arda Tezcan
Lieve Macken
Ghent University
first.lastname@ugent.be

Sven Coppers
Jens Brulmans
Jan Van den Bergh
Kris Luyten
Karin Coninx
UHasselt – tUL – EDM
first.lastname@uhasselt.be

Abstract

We present the highlights of the now finished 4-year SCATE project. It was completed in February 2018 and funded by the Flemish Government IWT-SBO, project No. 130041.[1]

We present key results of SCATE (Smart Computer-Aided Translation Environment). The project investigated algorithms, user interfaces and methods that can contribute to the development of more efficient tools for translation work.

Improved fuzzy matching: Levenshtein distance is not the best predictor for post-editing effort. Linguistic metrics and different metrics (such as TER) combined show better results.

Integration of Translation Memory (TM) and Machine Translation (MT): Combining TM matches, fuzzy match repair and SMT shows improvements over a baseline SMT.

Informed Quality Estimation: Accuracy and fluency error detection systems form the basis of the sentence-level Quality Estimation system, which results in better correlations with temporal post-editing effort compared to the Quest++ baseline. Detected errors can additionally be highlighted in the MT output.

Identifying bilingual terms in comparable texts: We found improvements when combining word embeddings with character-based models using a neural classifier trained on a seed lexicon. This includes short multi-word term phrases.

Post-Editing via Automated Speech Recognition (ASR): ASR for post-editing can benefit from additional information sources, such as the source language, the MT translation model and the activation of domain-specific terminology, for which we boosted ASR language model probabilities. The ASR language model is also enriched with character-level information, making it possible to model out-of-vocabulary words, which are very common in new domains.

Intelligible Translator Interfaces: We iteratively developed a functional prototype that integrates several of the aforementioned translation aids. In contrast with other approaches, our system applies the design concept of intelligibility to support translators' decision-making process when they interact with their translation environment. The evaluation showed that the prototype allows translators to better evaluate translation suggestions from MT, TM and term base but it had no major impact on their performance in terms of speed and quality. Furthermore, a small-scale lab experiment revealed no significant difference in efficiency between translating with the prototype and with a commercial tool, which shows less suggestions by default.

Integration: We created an interactive demo so that translators can experience and evaluate our research results: http://scate.edm.uhasselt.be/.

[1] http://www.ccl.kuleuven.be/scate

news.bridge – Automated Transcription and Translation for News

Peggy van der Kreeft
Deutsche Welle
Projects & Development
Bonn, Germany
peggy.van-der-kreeft@dw.com

Renars Liepins
LETA
Latvian News Agency
Riga, Latvia
renars.liepins@leta.lv

Abstract

news.bridge provides a platform for multilingual video processing, including automated transcription and translation, subtitling, voice-over, and summarization, with post-editing facility of videos in a broad range of languages. The platform is currently in beta testing at Deutsche Welle for republishing of videos in other languages.

1 The Project

news.bridge is an 18-month research project funded by Google Digital News Initiative (DNI). It is currently running (January 2018 – July 2019) and is a follow-up of a 6-month prototype DNI project, which proved the viability and potential of the concept. Therefore, news.bridge was started to turn this into a deployable and possibly commercially exploited platform. There has been major interest from the broadcasting and wider media world to participate in beta testing the platform, which shows high potential. news.bridge is a small consortium of four members. (1) German broadcaster Deutsche Welle is coordinator and user partner; (2) LETA, the Latvian News Agency, is platform developer; (3) Le Mans University is technology provider, in particular for transcription, translation and punctuation; and (4) Priberam, a Portuguese spinoff, focuses on summarization.

Visit our website (http://newsbridge.eu) for more details and follow us on Twitter (@newsbridge_htl).

2 The Platform

The news.bridge platform is a modular, dockerized system, developed by LETA, which currently runs remotely, but can also be installed locally. It ingests content from the Deutsche Welle repository, processes (transcribes, translates) it on demand, after selecting specific items. As it is meant for use in the production department, it has a post-editing user interface, allowing corrections to be made to the translation and transcription. To widen the number of languages covered, it includes internal, customized translation services from universities as well as off-the-shelf services, such as Google Translate, IBM Watson transcription and voice-over, via API. This combination leads to a tool that covers an extremely wide range of language combinations. Over 100 languages are included.

3 Multilingual News Production

Deutsche Welle, as coordinator and user partner, is currently enhancing and testing the tool for production. It has run a few user evaluation workshops and more are planned. It envisages different use cases. The tool is used, for instance, for transcription of interviews in any of the languages covered by the system. It is being tested for reprocessing of DW videos (with existing transcripts) in several languages and is being evaluated for gain in time and effort. The use of existing transcripts is a major factor, as the error rate of transcription (e.g. names, missing punctuation) is much higher than that for translation. Overall, initial evaluation results indicate that journalists welcome a tool that can help them produce videos cross-lingually. A high-quality transcript in one language is a solid basis for subtitling and voice-over into different languages. Finding the best tools for each language pair is part of the benchmarking effort.

Europarl Datasets with Demographic Speaker Information

Eva Vanmassenhove
ADAPT Centre
School of Computing and Engineering
Dublin City University
Dublin, Ireland
eva.vanmassenhove@adaptcentre.ie

Christian Hardmeier
Department of Linguistics and Philology
Uppsala universitet
Uppsala, Sweden
christian.hardmeier@lingfil.uu.se

1 Problem Statement

Research on speaker-adapted neural machine translation (NMT) is scarce. One of the main challenges for more personalized MT systems is finding large enough annotated parallel datasets with speaker information. Rabinovich et al. (2017) published an annotated parallel dataset for EN–FR and EN–DE, however, for many other language pairs no sufficiently large annotated datasets are available.

2 Datasets

To address the aforementioned problem, we publish a collection of parallel corpora licensed under the Creative Commons Attribution 4.0 International License for 20 language pairs available online: `https://github.com/evavnmssnhv/Europarl-Speaker-Information`. We tagged parallel sentences from Europarl (Koehn, 2005) with speaker information (name, gender, age, date of birth, euroID and date of the session) based on monolingual Europarl source files which contain speaker names on the paragraph level. We used meta-information of the members of the European Parliament (MEPs) released by Rabinovich et al. (2017) to retrieve the demographic annotations. An overview of the language pairs as well as the amount of annotated parallel sentences per language pair is given in Table 1.

3 Analysis

Additionally, we analyzed the EN–FR dataset with respect to the percentage of male versus female speakers in various age groups (see Figure 1).

Languages	# sents	Languages	# sents
EN–BG	306,380	**EN–IT**	1,297,635
EN–CS	491,848	**EN–LT**	481,570
EN–DA	1,421197	**EN–LV**	487,287
EN–DE	1,296,843	**EN–NL**	1,419,359
EN–EL	921,540	**EN–PL**	478,008
EN–ES	1,419,507	**EN–PT**	1,426,043
EN–ET	494,645	**EN–RO**	303,396
EN–FI	1,393,572	**EN–SK**	488,351
EN–FR	1,440,620	**EN–SL**	479,313
EN–HU	251,833	**EN–SV**	1,349,472

Table 1: Overview of annotated parallel sentences per language pair

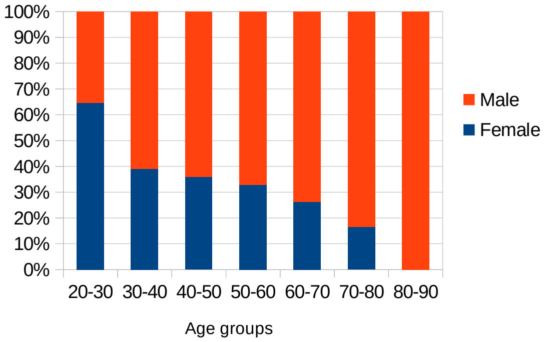


Figure 1: Percentage of female and male speakers per age group

References

Ella Rabinovich, Shachar Mirkin, Raj Nath Patel, Lucia Specia and Shuly Wintner, 2017. Personalized Machine Translation: Preserving Original Author Traits. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, Valencia, Spain, 1074–1084.

Philipp Koehn, 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the In MT Summit*, Phuket, Thailand, 79–86.