INTERACTIVE AND ADAPTIVE NEURAL MACHINE TRANSLATION

by

Rebecca Knowles

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

September, 2019

Abstract

In this dissertation, we examine applications of neural machine translation to computer aided translation, with the goal of building tools for human translators. We present a neural approach to interactive translation prediction (a form of "autocomplete" for human translators) and demonstrate its effectiveness through both simulation studies, where it outperforms a phrase-based statistical machine translation approach, and a user study. We find that about half of the translators in the study are faster using neural interactive translation prediction than they are when post-editing output of the same underlying machine translation system, and most translators express positive reactions to the tool. We perform an analysis of some challenges that neural machine translation systems face, particularly with respect to novel words and consistency. We experiment with methods of improving translation quality at a fine-grained level to address those challenges. Finally, we bring these two areas – interactive and adaptive neural machine translation – together in a simulation that shows that their combination has a positive impact on novel word translation and other metrics.

ABSTRACT

Primary Reader and Advisor: Philipp Koehn

Secondary Readers: Kevin Duh and Matt Post

Acknowledgments

The Ph.D. is a long process, but in many ways it feels like the culmination of a much longer journey. In these acknowledgments, I'd like to thank the people who were part of this most recent adventure, as well as the people who helped to set me on this path and guided me along the way. It isn't concise, yet it still feels incomplete.

I worked with two excellent advisors over the course of my Ph.D.: Philipp Koehn and Mark Dredze. Philipp has given me freedom to pursue my research interests, offered guidance when I needed it, and has done so with a helpful sense of calm and a good sense of humor. Mark has remained a mentor to me, never hesitating to make time for my questions and concerns even after I "officially" left his group.

Matt Post and Kevin Duh served as members of my committee and as research collaborators. I am grateful for my committee members' comments and suggestions, and the early feedback they, Erica Michael, and Chadia Abras provided as members of my Graduate Board Oral exam committee.

Over the past years, I have had the opportunity to publish with a number of excellent coauthors and to discuss research with peers and mentors in and outside of the

Center for Language and Speech Processing. I would like to specifically acknowledge Marina Sanchez Torron and Sachith Sri Ram Kothur, with whom I collaborated on work directly related to this dissertation. The Machine Translation Reading Group has been a great place to share, discuss, and improve research ideas. My two summers in Berlin, Germany (complete with the World Cup win in 2014) were formative months for me as a researcher and as a person. I'm so grateful to my mentors from those two internships – Alex Klementiev, Gyuri Szarvas, and Weiwei Cheng – as well as the colleagues and friends that I first met there. I am also thankful for the research-related (and otter-related) enthusiasm of my peers in the Hackerman second floor office. Despite sitting all the way up on the third floor, Rachel Rudinger has been a wonderful constant in my graduate student experience from our visit days to our job searches; I'm delighted that we are finishing up together.

I would especially like to thank Ruth Scally, Yamese Diggs, Carl Pupa, and most recently Jennifer Linton, who contribute so much to CLSP, both personally and professionally. Similarly, Zack Burwell has served such a role for the computer science department. Whenever I've asked questions, needed a special server for a user study, wanted to share (or consume) baked goods, or been looking for wisdom, they've been there with support, kindness, and a deep knowledge of the inner workings of the center, department, and grid.

I am indebted to the excellent math, language, linguistics, and computer science teachers who taught me from middle school through college. Betty Medalie first planted the seed in my mind about math, Larry Gilbert always had an open door for conversations about math (or M*A*S*H), Dave Lowther gave me an opportunity to share math I was learning with my Abuelos, Bill Tabrisky convinced me to step out beyond my comfort level with unfailing kindness and confidence in me, Tony Asdourian gave me an unforgettable assignment on my first day of calculus, Rina Foygel taught me group theory while I was still in upper school, and Arnaldo Cohen has kept up a keen interest in my goals and studies. Fathim Craven started me on my path of learning French, Raymonde Arseneau challenged me as a French teacher and supported me as an advisor, and Christine Broening taught me French, German, and how to make an excellent chocolate roll cake. My undergraduate mathematics thesis was advised by Lynne Butler, who always set high expectations for me and backed them up with good advice and guidance. Nathan Sanders advised my linguistics thesis and helped me to navigate the Swarthmore honors program. Hanna Wallach and Glen Coppersmith were instrumental in my decision to pursue a Ph.D. in computer science. Sorelle Friedler, Meg Mitchell, Shizhe Huang, Donna Jo Napoli, K. David Harrison, Rachel Sutton-Spence, Bruce Bumbarger, Samira Phillips, Mimi Cukier, Richard Linder, Chris Callison-Burch, Theresa Wilson, Ben Van Durme, Rob Manning, Josh Sabloff, John Dougherty, Dave Wonnacott, Steven Lindell, Suzanne Lindell, Rich Wicentowski, and so many others offered guidance and advice before and during my studies. Matt Post and Adam Lopez's machine translation class helped spark my interest in the research area, though it took some time to make the switch.

ACKNOWLEDGMENTS

It is often said that one of the best ways to learn a subject is to teach it, so I would like to thank those that I've had the pleasure of teaching and of learning from: NACLO students, Data Structures students and course assistants, students in my classes at NDP, students in my HEART courses, and everyone else who has let me teach them one thing or another. Their interest, questions, and insights reminded me why I chose to pursue my studies in this field. In learning to become a better teacher, I received excellent guidance from Peter Fröhlich, Joanne Selinski, the CS Education reading group, former teachers, the SIGCSE community, and many more.

I am grateful to the many friends in Baltimore and around the world who have made this journey more pleasant. Outside of my studies, several organizations provided me with friendly, creative, and athletic outlets. The members of the Graduate Association of Women in Computer Science and Electrical & Computer Engineering have shared indispensable support, laughter, and camaraderie. While in Berlin, I regularly attended the *Berliner Quäkergruppe*, who welcomed me and every newcomer passing through their doors. At home in Baltimore, I am thankful for the friendship and space for silence that I have had through Stony Run Monthly Meeting. The German program at Johns Hopkins welcomed me as an auditing graduate student into several years of classes – vielen Dank! The Edible Book Festival gave me an excuse to bake *and* the unforgettable opportunity to ask Philipp Koehn to sign – in edible ink – a cake-based replica of his own textbook. Potomac Curling Club has been a wonderful community and a haven from academic and other concerns since 2015. Baltimore Rowing Club

played a similar role in the summer of 2018. I remain very thankful for my time on the water (frozen or otherwise) and all the unexpected doors it has helped to open in my life.

I'm very grateful for the support and love of my immediate and extended family. Despite their having given me the gift of *two* languages from birth, I'm not sure I can find the words to properly thank my parents, Nancy Fink and Jonathan Knowles. My father made a regular habit of letting me know that I brought him joy; I like to think that these recent successes would have done just that. My mother has been a source of encouragement in all things: reading my papers and this dissertation (any remaining typos are all my own), sharing all the ups and downs that come with being sports fans, offering wisdom or just lending an ear when I needed it, and infinitely more. I'd also like to thank Cati, who made many of the hardest days that much more bearable and many of the good days all that much better.

There are more people to thank for their support and friendship than I have space to thank in this section; I can only hope I've made my appreciation clear to all of you.

ACKNOWLEDGMENTS

Much of the material in this dissertation is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1232825. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Dedication

A mi madre.

Contents

List of Tables

List of Figures

Part I

Introduction

Chapter 1

Introduction

In recent years, machine translation quality has improved by leaps and bounds. For many domains and language pairs, it is of sufficiently high quality to enable *assimilative* use by a reader, for example someone interested in reading a news article in a language that they do not know. We have also seen advances in machine translation for *communication*; with easy access to online translation systems (including speech-to-speech translation), machine translation tools are used by some for near-instant communication across language barriers. This dissertation focuses on a third goal of machine translation: *dissemination*, or machine translation for the purpose of publishing and sharing information in more than one language.

Even with progress in machine translation research, there are still numerous translation use cases which require the attention of human translators, and this is likely to remain the case for the foreseeable future – particularly in the case of translation for dissemination. Human translation is necessary in cases where translation quality must meet a high threshold, such as medical, government, or legal translations. However, the higher quality of human translation comes at a cost in terms of speed: a human translator may be able to translate on the order of 2000 words a day (Chan, 2002) while a machine translation system can translate more than twice that number of words in a single *second*. Research in computer aided translation seeks to assist human translators by incorporating the strengths of machine translation into tools that can be used in their work.

This dissertation focuses on two main topics in neural machine translation and computer aided translation: interactivity and adaptivity. The interactive portion covers research on neural interactive translation prediction. Interactive translation prediction operates like "auto-complete for translators" – similar to how auto-complete on a smartphone or tablet predicts the next word that the user might type, interactive translation prediction predicts the next words in a translation, based on the source sentence and the translation produced thus far. The translator can either accept the system's suggestions or type their own corrections, after which the system adjusts and returns updated suggestions. Today's neural machine translation systems provide an interesting opportunity for exploring computer aided translation and interactive machine translation in particular. Not only are they showing state-of-the-art performance in MT evaluations, but their decoding process is naturally analogous to human sentence production in that it proceeds from the beginning of the sentence to the end, producing one token at a time. We describe a neural approach to interactive translation prediction and run a user study with professional translators to validate its usefulness and collect translator impressions. Approximately half of the translators were able to translate more quickly using interactive translation prediction (as compared to post-editing), and most had positive impressions of the tool.

The adaptive portion of the dissertation focuses on improving the underlying machine translation systems by incorporating feedback from human translator corrections and from data resources like small bilingual lexicons. Simulating translator interactions, we show that these adaptive techniques can perform fine-grained adaptation, improving translation over the course of translating a single document. Bringing the two portions of the dissertation together, we show that a combination of interactive and adaptive machine translation has a positive impact on several metrics that relate to translator efficiency and to known pain points for human translators.

1.1 Contributions

The main contributions of this dissertation are:

- An approach to interactive translation prediction using a neural machine translation system.

- A user study and examination of translator performance with and perception of neural interactive translation prediction.

- An analysis of neural machine translation performance on rare and challenging words.
- A demonstration of the effectiveness of several approaches to fine-grained adaptation of neural machine translation systems, which could be applied in computer aided translation settings.

1.2 Structure of the Dissertation

The dissertation is structured as follows:

- Part II provides a survey of prior art in relevant areas of research.
    - Chapter 2 describes machine translation research, including a brief overview of machine translation history, phrase-based statistical machine translation, and neural machine translation. It also covers issues specific to the handling of vocabulary in neural machine translation as well as approaches to domain adaptation.
    - Chapter 3 focuses on the interaction between human translators and machines. It includes a discussion of the history of computer aided translation research, with a focus on tools for interactive translation.
    - Chapter 4 covers the data and models that are used most frequently across the dissertation. Data and models that are used only once (rather than

  across multiple chapters) are introduced and described at the point at which they are used.

- Part III consists of two chapters describing research on neural interactive translation prediction.

  - Chapter 5 introduces neural interactive translation prediction, an editing mode for human translators interacting with machine translation output. It provides a proof of concept through simulations, which show that interactive translation prediction with neural models outperforms the same using phrase-based statistical machine translation systems, even when the underlying machine translation quality of the two models is comparable.
  - Chapter 6 takes the approach of the previous chapter and implements GPU-based neural interactive translation prediction in a computer aided translation tool for use in a user study. The results of a small user study show promise for the usefulness of the technique, and an analysis of translator surveys indicate overall positive sentiment toward neural interactive translation prediction.

- Part IV considers the types of errors that translators may encounter when using a computer aided translation tool and examines the ways that such machine translation challenges can be addressed to improve translation performance in the computer aided translation setting.

  - Chapter 7 analyzes machine translation behavior with a focus on word-level performance. This includes an examination of the challenge that rare and novel words pose to machine translation systems, an examination of word copying in neural machine translation, and brief case studies of consistency in human and machine translation at the document level. All of this work serves to provide evidence of the need for computer aided translation solutions that address these challenges.
  - Chapter 8 proposes two methods for addressing the aforementioned issues of rare words and consistency. The first approach, dictionary training, focuses only on improving the translation on novel words (which may appear multiple times in a new document). The second approach, single-sentence adaptation, can provide improvements in terms of consistency as well as novel word translation. Finally, the two approaches are combined. We evaluate these by simulating the realistic scenario of a human translator being tasked with translating whole documents sentence by sentence.

- Part V, which consists of Chapter 9, brings together the two main contributions of the dissertation: interactive and adaptive neural machine translation for computer aided translation. In simulation, this chapter shows the potential for improved performance when fine-grained adaptation is combined with neural interactive translation prediction.

1.3 Publications

Significant portions of this dissertation are based on the work published during my[1] time as a graduate student. Their contributions to individual chapters can be described approximately as follows. Individual chapters provide more information about the papers used and the contributions of the authors.

- Chapter 5 draws primarily from: Rebecca Knowles and Philipp Koehn (2016). "Neural Interactive Translation Prediction". In: *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*. Austin, Texas, USA. URL: `https://amtaweb.org/wp-content/uploads/2016/10/AMTA2016_Research_Proceedings_v7.pdf#page=113`
- Chapter 6 draws primarily from: Rebecca Knowles, Marina Sanchez-Torron, and Philipp Koehn (2019). "A user study of neural interactive translation prediction". In: *Machine Translation*. URL: `https://doi.org/10.1007/s10590-019-09235-8`
- Chapter 7 draws from the following papers:
  - Philipp Koehn and Rebecca Knowles (2017). "Six Challenges for Neural Machine Translation". In: *Proceedings of the First Workshop on Neural*

[1] Outside of this introduction, the main body of the remainder of this dissertation uses the first person plural ("we") rather than the singular ("I"), a choice made for the sake of consistency, tradition, and so as not to disregard the contributions made by colleagues and coauthors in the case of joint work. Those collaborations are described in more detail in the relevant sections.

  Machine Translation. Vancouver, Canada: Association for Computational Linguistics, pp. 28–39. URL: https://www.aclweb.org/anthology/W17-3204

  - Rebecca Knowles and Philipp Koehn (2018a). "Context and Copying in Neural Machine Translation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium: Association for Computational Linguistics, pp. 3034–3041. URL: https://www.aclweb.org/anthology/D18-1339

- Chapter 8 draws primarily from: Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn (2018). "Document-Level Adaptation for Neural Machine Translation". In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation.* Melbourne, Australia: Association for Computational Linguistics, pp. 64–73. URL: https://www.aclweb.org/anthology/W18-2708

Work from the following publication is also addressed briefly: Rebecca Knowles and Philipp Koehn (2018b). "Lightweight Word-Level Confidence Estimation for Neural Interactive Translation Prediction". In: *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing.* Boston, Massachusetts, USA: Association for Machine Translation in the Americas, pp. 35–40. URL: https://www.aclweb.org/anthology/W18-2102

1.4 Additional Publications

The following publications were also produced during my time as a graduate student, but do not form part of this dissertation.

- Machine Translation:
  - Huda Khayrallah, Rebecca Knowles, Kevin Duh, and Matt Post (2019). “An Interactive Teaching Tool for Introducing Novices to Machine Translation”. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE ’19. Minneapolis, Minnesota, USA: ACM, pp. 1276–1276. URL: `http://doi.acm.org/10.1145/3287324.3293840`
  - Rebecca Knowles, John Ortega, and Philipp Koehn (2018). “A Comparison of Machine Translation Paradigms for Use in Black-Box Fuzzy-Match Repair”. In: *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*. Boston, Massachusetts, USA: Association for Machine Translation in the Americas, pp. 249–255. URL: `https://www.aclweb.org/anthology/W18-2108`
  - Christo Kirov, John Sylak-Glassman, Rebecca Knowles, Ryan Cotterell, and Matt Post (2017). “A Rich Morphological Tagger for English: Exploring the Cross-Linguistic Tradeoff Between Morphology and Syntax”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valen-

cia, Spain: Association for Computational Linguistics, pp. 112–117. URL: `https://www.aclweb.org/anthology/E17-2018`

- Other topics:
  - Rebecca Knowles, Adithya Renduchintala, Philipp Koehn, and Jason Eisner (2016). "Analyzing Learner Understanding of Novel L2 Vocabulary". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pp. 126–135. URL: `https://www.aclweb.org/anthology/K16-1013`
  - Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner (2016b). "User Modeling in Language Learning with Macaronic Texts". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1859–1869. URL: `https://www.aclweb.org/anthology/P16-1175`
  - Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner (2016a). "Creating Interactive Macaronic Interfaces for Language Learning". In: *Proceedings of ACL-2016 System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, pp. 133–138. URL: `https://www.aclweb.org/anthology/P16-4023`

- Rebecca Knowles, Josh Carroll, and Mark Dredze (2016). “Demographer: Extremely Simple Name Demographics”. In: *Proceedings of the First Workshop on NLP and Computational Social Science.* Austin, Texas, USA: Association for Computational Linguistics, pp. 108–113. URL: `https://aclweb.org/anthology/W16-5614`
- Rebecca Knowles, Mark Dredze, Kathleen Evans, Elyse Lasser, Tom Richards, Jonathan Weiner, and Hadi Kharrazi (2014). “High Risk Pregnancy Prediction from Clinical Text”. In: *NeurIPS Workshop on Machine Learning for Clinical Data Analysis.* Montreal, Canada.
- Charley Beller, Rebecca Knowles, Craig Harman, Shane Bergsma, Margaret Mitchell, and Benjamin Van Durme (2014). “I’m a Belieber: Social Roles via Self-identification and Conceptual Attributes”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 181–186. URL: `https://www.aclweb.org/anthology/P14-2030`.

Part II

Background

Chapter 2

Machine Translation

2.1 Introduction

The field of machine translation has a long history. It can trace some of its earliest success to the cryptographic research focus during World War II, from which Warren Weaver drew inspiration in his famous quote: “When I look at an article in Russian, I say ‘This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode”’ (Weaver, 1949). Dorr, Jordan, and Benoit (1999) provides an extensive survey of the state of machine translation paradigms and architectures on the eve of the turn of the millennium, including rule-based machine translation, example-based machine translation, neural network based machine translation (albeit with extremely limited vocabularies), and statistical machine translation. Neural machine translation, the approach that most of this dissertation focuses on, is the latest in a sequence of machine translation paradigms to achieve and advance the state of the art.

In the remainder of this chapter, we first touch on the evaluation of machine translation. We describe phrase-based statistical machine translation (SMT), which had been the dominant paradigm until quite recently, and is the paradigm around which much of the recent work in computer aided translation has been performed. We then describe neural machine translation (NMT), the types of vocabularies that are typically used for those systems, and finish by discussing related work on domain and project adaptation for neural and phrase-based statistical machine translation, with particular attention to work relating to computer aided translation. Both phrase-based and neural machine translation are data-driven approaches: they require the use of *parallel corpora*, also known as *bitexts*. A parallel corpus is a set of translation segments (usually sentences) in one language with their corresponding translations into the other language of interest. The corpora that we use to build strong machine translation systems typically number in the millions of sentence pairs.

2.2 Machine Translation Evaluation

2.2.1 BLEU Score

Bilingual Evaluation Understudy (BLEU) scores (Papineni et al., 2002) are commonly used to evaluate the quality of machine translation. BLEU combines n-gram precision (the number of n-grams in the MT output that match the reference, divided by the total number of n-grams in the MT output) with a brevity penalty (1 if the MT output is longer than the reference, otherwise the length of the MT output divided by the length of the reference), as follows:

$$\text{BLEU}_N = (\text{brevity penalty}) \prod_{n=1}^{N} (\text{precision}_n)^{w_n} \tag{2.1}$$

More technically, computed over a corpus of machine translated candidate sentences with one reference each:[1]

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases} \tag{2.2}$$

$$\mathit{Count}_{\mathit{clip}}(x) = \min(\mathit{Count}(x), \mathit{Max\_reference\_count(x)}) \tag{2.3}$$

$$p_n = \frac{\sum_{s \in \{\mathit{candidate\ sentences}\}} \sum_{\mathit{n\text{-}gram} \in s} \mathit{Count}_{\mathit{clip}}(\mathit{n\text{-}gram})}{\sum_{s' \in \{\mathit{candidate\ sentences}\}} \sum_{\mathit{n\text{-}gram}' \in s'} \mathit{Count}(\mathit{n\text{-}gram}')} \tag{2.4}$$

$$\text{BLEU}_n = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.5}$$

Where w_n is a weight assigned to the particular size of n-gram (often set uniformly to $\frac{1}{N}$ or simply to 1), r is the reference length, and c is the candidate length.

[1] BLEU was designed to be run over multiple reference sentences, but in practice is most frequently evaluated with only one single reference.

While BLEU is widely used in MT evaluations, it has a number of weaknesses, including the fact that it does not account for synonyms.[2] It is possible to achieve improvements in translation quality (as judged by human annotators) without a corresponding improvement in BLEU (Callison-Burch, Osborne, and Koehn, 2006), so it is important to have other evaluation techniques available, particularly for tasks such as adapting to human translators, where a small improvement (for example in translation consistency of one relatively frequent word or phrase) could have a large impact on user satisfaction.

2.2.2 Other Metrics

The goal of all of these metrics is to approximate the gold standard: human evaluation of machine translation. We use additional metrics in our evaluation of computer aided translation tools (word prediction accuracy, accuracy of novel word translation, etc.), and those are introduced in the dissertation when they are first used. We use these task-specific metrics throughout this work to measure the impact of various approaches on specific outcomes of interest. We typically use BLEU as a benchmark to give an overall sense of quality or to compare two systems directly.

In addition to BLEU score there are a number of other available metrics. Commonly used ones include TER (translation error/edit rate; Snover et al., 2006), and WER

[2] Alternatives like METEOR (Banerjee and Lavie, 2005) have been proposed to resolve this; METEOR uses stemming and semantic resources to handle this, but is more computationally expensive as a result.

(word error rate), which can be calculated automatically from just the translation output and reference without the need for external or language-specific resources. METEOR (Banerjee and Lavie, 2005) and its successors can incorporate additional language specific features like stemming or semantic relatedness. This short list is by no means exhaustive, and merely scratches the surface of the range of research in this area. The WMT shared task on metrics provides regular evaluations of a wide range of metrics (Ma et al., 2019).

2.3 Phrase-Based Statistical Machine Translation

The dominant paradigm in machine translation until recently,[3] phrase-based statistical machine translation (SMT) systems typically consist of two main components: a phrase table and a language model. The phrase table consists of source language phrases (sequences of one or more tokens) and corresponding target language phrases, associated with translation probabilities. These can be automatically extracted from large parallel corpora using unsupervised or semi-supervised alignment techniques. The language model (most recently likely to be a neural language model) provides scores for sentences on the basis of target language fluency. A language model can

[3] And still outperforming neural machine translation on some low-resource tasks (Koehn and Knowles, 2017).

be trained on target language monolingual corpora (which are typically much larger than the parallel corpora available). Combined, the translation and language model probabilities provide scores for possible translations in a search lattice, incorporating both adequacy and fluency in translation.

For a much more in-depth exploration of phrase-based statistical machine translation, see Koehn (2010). We use Moses (Koehn et al., 2007) for all of our phrase-based statistical machine translation models.

2.4 Neural Machine Translation

In recent years, neural machine translation (NMT) models have proven themselves to be state of the art across a number of language pairs (Bojar et al., 2016; Bojar et al., 2017; Bojar et al., 2018). As these models have gained increasing traction, there has been a veritable boom in model architectures, ranging from encoder-decoder models (Kalchbrenner and Blunsom, 2013; Sutskever, Vinyals, and Le, 2014) to encoder-decoder models with attention (Bahdanau, Cho, and Bengio, 2015) to self-attention models (Vaswani et al., 2017) and beyond. In contrast to phrase-based statistical machine translation systems, these systems tend to share the property that they are trained jointly, end-to-end, rather than having a number of separate components like the phrase table and language model. In this dissertation, we primarily focus on recurrent neural network (RNN) sequence-to-sequence encoder-decoder machine translation models with attention, similar to those described in (Bahdanau, Cho, and Bengio, 2015).

Such a model consists of three main components:

- an **encoder** stage where the input sentence is processed by two recurrent neural networks, one running left-to-right, the other right-to-left, resulting in hidden states for each token that encode it along with its left and right context,
- a **decoder** stage where the output sentence is produced sequentially, one token at a time, by conditioning on previous output tokens via a hidden state (roughly corresponding to a language model in traditional statistical machine translation) and on the input encoding (roughly corresponding to a translation model), and
- an **attention mechanism** that conditions the prediction of each output token on a distribution over input tokens (roughly corresponding to a soft alignment function).

We walk through the model below, focusing on a high-level understanding of its characteristics rather than a fine-grained consideration of its implementation. For more details, see Bahdanau, Cho, and Bengio (2015), whose notation this section follows, or the description of the Nematus toolkit (Sennrich et al., 2017).[4]

At each time step t, the standard decoder computes the conditional probability of

[4] Nematus was used for many of our experiments.

generating a token y_t given the input sentence $\vec{x}$. This is defined to be:

$$p(y_t|\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}, \vec{x}) = g(\hat{y}_{t-1}, c_t, s_t) \tag{2.6}$$

where g is a non-linearity, $\hat{y}_{t-1}$ is the token produced by the previous decoding step, c_t is a context vector, and s_t is the hidden state for time t.

During encoding, so-called annotations h_t were produced for each token x_t in the input sentence $\vec{x} = (x_1, \cdots, x_T)$. These h_t were produced by concatenating the forward and backward hidden states produced for each token by the forward and backward RNNs, respectively. We can think of these as continuous representations of each input token in context; by virtue of the concatenation of the forward and backward RNNs, they contain information about the token x_t at position t in the input sentence as well as about its full left and right context.

The context vector c_t in Equation 2.6 is a weighted average of the annotations. First, weights $\alpha_{tj} = \exp(e_{tj}) / \sum_{k=1}^{T} \exp(e_{tk})$ are computed, where $e_{tj} = a(s_{t-1}, h_j)$ can be thought of as a soft alignment model (parameterized as a neural network and jointly trained with the rest of the system). The weight α_{tj} can be interpreted roughly as the probability that y_t is aligned to x_j, resulting in soft alignments used by the system's attention mechanism to weight the focus of the context vector. The context vector is then computed as $c_t = \sum_{j=1}^{T} \alpha_{tj} h_j$.

As indicated above, decoding in this attention-based neural machine translation approach proceeds token by token. At each step of the decoding process, a probability distribution over possible next tokens is computed. This is conditioned on the previous token, the context vector, and the hidden state. The highest scoring token is selected and used in the conditioning context for the next step. Alternatively, similar to beam search in traditional statistical machine translation decoding, the top n next tokens may be considered and competing hypotheses with different output tokens maintained. Each of the hypotheses (consisting of a token sequence and a hidden state, and ranked by the combined token translation probabilities) is extended at the next decoding step.

There are various choices for the exact design of the recurrent neural networks used in the encoder and decoder. There are multiple options for the cells used, such as long short term memory (LSTM) cells or gated recurrent units (GRU). There are also alternative architectures such as deep RNNs, convolutional neural network (CNN) approaches, attention-only approaches, and more.

The system is trained to minimize cross-entropy on the training corpus, optionally with early-stopping based on development set scores.

2.5 Vocabulary and Byte Pair Encoding

Vocabulary size has been a major concern in neural machine translation. Due to computational constraints, early work operated with small fixed-size vocabularies of under 100k word types (Sutskever, Vinyals, and Le, 2014; Bahdanau, Cho, and Bengio, 2015),[5] typically replacing rare and unknown words with a special UNK token. These small vocabulary models could either be postprocessed to handle rare and unknown vocabulary items, or could use alternative architectures to copy words or perform character-level translation (Arthur, Neubig, and Nakamura, 2016; Gulcehre et al., 2016; Luong and Manning, 2016; Nguyen and Chiang, 2018). Without adequate handling of rare and unknown words, it is impossible for neural translation systems to perform well on held-out data or to compete against open-vocabulary and near-open-vocabulary phrase-based statistical machine translation systems.

As a result, rather than encoding and decoding full words, many neural machine translation systems – including those described in this dissertation – operate on subword vocabularies. These subword vocabularies allow words in the corpus to be expressed either as individual vocabulary items or as sequences of subwords; in particular, frequent words are often given their own vocabulary entry, while infrequent words are split into subwords. This allows systems to be built with smaller vocabulary sizes (for computational efficiency) while still maintaining full or near-full coverage of corpus vocabulary. Here we will focus on byte pair encoding (BPE), a compression algorithm (Gage, 1994) which Sennrich, Haddow, and Birch (2016c) applied to text preprocessing for machine translation. In order to build a vocabulary of subword symbols, an iterative algorithm is employed. First, all characters observed in the

[5] Much earlier systems used even smaller toy vocabularies consisting of only tens of types, as noted in Dorr, Jordan, and Benoit (1999).

training set are added as symbols to the (initially empty) vocabulary. From there, the most frequent pair (as observed in the training data) of symbols in the vocabulary are merged into a new symbol, which is subsequently added to the vocabulary. Symbol pairs that cross word boundaries are not included in the frequency counts, ensuring that the symbols in the vocabulary are indeed subwords.[6] This process continues until a predetermined number of merges is reached.

The vocabulary will then contain a fixed number of subword symbols. New text, e.g., at test time, can be segmented according to the same process: after splitting it into individual characters, the merges learned through the vocabulary-creation algorithm can be applied (in order of greatest frequency) to deterministically convert text into sequences of items from the (subword) vocabulary.

Due to the iterative merging of the most frequent symbol pairs, the completed vocabulary will encode common whole words as their own unique symbols, while infrequent words will need to be represented by sequences of subword symbols. This has several benefits and drawbacks. Having frequent words treated as individual vocabulary symbols allows the system to learn appropriate embeddings for them, while certain types of less frequent words may benefit from their separation into subwords. In terms of source vocabulary, this provides the possibility of successfully translating or copying source words that have never been seen in training data. Also, for certain languages, this may allow for parameter sharing between tokens that share the same

[6] One could certainly also consider including word boundaries, which would lead to a vocabulary in which very frequent collocations (phrases) are included alongside full words, subwords, and characters.

stem or share affixes. While BPE is not a morphological segmentation algorithm, we do observe that for languages with a number of morphologically meaningful affixes (e.g., verb endings, affixes that indicate number, gender, or case, etc.), BPE often learns segmentations that look potentially morphological at a surface level. Segmentation will be indicated either using two "at" symbols, as standard in some BPE toolkits, or using a vertical bar ("@@"or "|"). These could include segmenting plurals ("cats" to "cat@@ s"), verbs ("laughing" to "laugh@@ ing"), or compound nouns ("fireflies" to "fire@@ flies"), among others. However, it can also learn linguistically uninformative segmentations, like "fling" to "fl@@ ing" (which might erroneously suggest that "fl" is the root of a verb).

While BPE is often described as solving the open vocabulary problem, there are several rare instances where it may fail, resulting in unknown words or subwords. The first is the case in which a new character is introduced (for example, a character in a different script, a currency symbol, an accented character, or other special character). Since that character was not in the initial vocabulary, it must be treated as an unknown symbol. Sennrich, Haddow, and Birch (2016c) also note that unknown symbols could occur if a string occurs in test data that had been merged at all instances in the training data, but point out that this could be resolved by undoing the last merges until a point is reached where all of the symbols are known. Both of these types of failures are rare, but may occur more frequently if the training data and test data are not well matched.[7]

In practice, there are a number of factors to consider in the use of BPE. This includes the number of merges to perform, which has an impact on the overall size of the vocabulary as well as the distribution of full words vs. subwords in the vocabulary. There is also the question of whether the BPE algorithm's merges should be performed over each language's data separately, or whether a joint merging model should be learned. Whether the vocabulary (as learned through the BPE algorithm) is used unchanged as the system's vocabulary or whether a new vocabulary (a subset of the BPE vocabulary) is extracted from the training data can also have an effect on the ability to encode or generate words that were unobserved at train time. Chapter 7 provides more detailed analysis on the effect of byte pair encoding on the translation of rare and novel words.

We use the `subword-nmt` implementation of BPE.[8] There exist other approaches to generating subword vocabularies as well.[9]

2.6 Adaptation

Domain adaptation has long been an area of interest for researchers in the machine translation community. Since machine translation models require large amounts of

[7] Anecdotally, this appears to be the case when processing patent abstract data using BPE trained on more general domain in English. The patent data includes Greek letters and other special characters, which do not occur in the initial vocabulary.

[8] `https://github.com/rsennrich/subword-nmt`

[9] E.g., `https://github.com/google/sentencepiece`

data, it is often the case that general domain systems are built using all available data. When these systems are applied to specialized domains that are not well-matched to the training data, they may exhibit a drop in overall performance; this drop is especially severe for neural machine translation systems (Koehn and Knowles, 2017). Here we focus on recent work on neural machine translation model adaptation, noting specific phrase-based statistical approaches (on which there exists a large body of work) when they are most relevant to the computer aided translation setting.

Recent work (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015) has proposed to do domain adaptation for NMT systems by training a general system then fine-tuning by continuing to train using only in-domain data (typically a smaller dataset). Wang et al. (2017) present a similar approach where they weight each source-target sentence pair during training based on scores from in-domain and out-of-domain language models. Kobus, Crego, and Senellart (2017) use special tokens to indicate domain. Chu, Dabre, and Kurohashi (2017) compare the approaches. These approaches typically use larger amounts of in-domain data to do adaptation, far greater than the amounts that might be available in a CAT setting.

Cettolo et al. (2014) proposed adapting statistical phrase-based machine translation systems to particular projects consisting of multiple documents. Blain, Schwenk, and Senellart (2012) and Blain et al. (2015) also perform adaptation for computer aided translation. Peris and Casacuberta (2019) and Peris, Cebrián, and Casacuberta (2017) propose adapting neural machine translation systems in computer aided translation settings. Ter-Sarkisov et al. (2015) perform project adaptation on a neural language model for use in computer aided translation with a phrase-based statistical machine translation system. None of these explore very small amounts of data at the sub-document level (though Peris and Casacuberta (2019) do perform adaptation to single sentences using a similar approach to our work in Kothur, Knowles, and Koehn (2018) and to the phrase-based statistical approach in Denkowski, Dyer, and Lavie (2014)).

Two recent papers have tried a domain adaptation approach using very small data sizes, ranging from 1 sentence to 128 sentences (Farajian et al., 2017; Li, Zhang, and Zong, 2018). They adapt models for new sentences by training on sentence pairs from a training corpus (or translation memory) that are similar to the new sentence, which means they cannot adapt to novel vocabulary. Karimova, Simianer, and Riezler (2018) have recently shown in a user study with translation students that an online adaptation approach (similar to our single-sentence adaptation approach in Kothur, Knowles, and Koehn (2018)) can decrease post-editing effort. Daems and Macken (2019) compare phrase-based statistical adaptive interactive machine translation and neural adaptive interactive systems in a user study, finding the tools to be similar in quality and finding that translators had generally positive reactions.

Chapter 3

Human Translation and Computer Aided Translation

3.1 Introduction

Computer aided translation[1] (CAT) refers to a process whereby a human translator translates with the aid of one or more translation-related technologies. Like machine translation, computer aided translation has a long history, predating the invention of the personal computer and computer aided translation tools as we know them today. The ALPAC report (ALPAC, 1966), generally remembered as almost entirely negative due to the blow it dealt to machine translation research (and subsequent research funding), had tentatively positive things to say about the early stages of computer

[1] Sometimes styled as computer-aided translation, computer-assisted translation, or computer assisted translation.

aided translation (then called machine aided translation). The report describes the usefulness of a computer-based glossary of technical terms for translation, then employed by the Federal Armed Forces Translation Agency in Mannheim, Germany, which increased translator speed and accuracy. It also describes post-editing and what might be categorized as an early form of translation memory used by the European Coal and Steel Community in Luxembourg. Results for this were more mixed, but they did note the potential ability for these tools to be continually improved and adapted with the addition of more translated data.

In one of the seminal pieces of literature on the topic, Kay (1980) set forth a proposal for a *translator's amanuensis*, a sort of computer aided translation tool, or, more accurately, he proposed a path towards building such a tool. Kay advocated for the iterative implementation of pieces of a tool that would be "gradually, almost imperceptibly, allowed to take over certain functions in the overall translation process." This would begin with tasks that the computer was best suited to do, then slowly take on more responsibility as its capabilities increased. At its core, though, it would be in the hands of the human translator, who would have the final say in which functions they would like to use: the tool "will always be under the tight control of a human translator," with the goal of increasing productivity but not replacing the translator. Cadwell, O'Brien, and Teixeira (2018) echo this in their examination of professional translator perceptions of machine translation, noting the importance of translators' agency and sense of control; a group of translators who were able to easily enable or disable machine translation assistance and had been given input in its development had more positive reactions to the introduction of it as a tool of their trade than a group who did not. In a survey of professional translators, Moorkens and O'Brien (2017) found that 63% had a preference for a user interface that could be customized to their particular needs and desires.

Computer aided translation today covers a wide range of types of tools, which we summarize briefly before focusing on the two most closely related to machine translation and the remainder of this dissertation: post-editing and interactive translation prediction.

3.2 CAT Technologies

In *Computer-Aided Translation Technology: A Practical Introduction*, Bowker (2002) provides a clear overview of CAT technologies. Aimed at an audience primarily of translation students, this book divides CAT tools into four main categories. We follow an approximation of those categories in this very brief introduction, albeit in a different order, and direct the reader to that book and other relevant resources for additional detail.[2]

[2] When no other resource is cited, Bowker (2002) provides information on the topic.

3.2.1 Categories of CAT Tools

Formats

While electronic text-to-text machine translation research has largely focused its attention on translating plain text sentences (or perhaps even documents) from one language to another, starting from preprocessed plain text files, data formatting is of major practical concern to human translators. They may be working from hard copies, in which case optical character recognition technologies may be useful in providing a starting point to convert the data into a format which will enable the use of other CAT tools (another alternative is to convert the text into a machine-readable format using automatic speech recognition). Alternatively, the text may already be provided in a particular electronic format, such as a PDF, spreadsheet, word processing document, or so on. In many cases, the translator is expected to return their finished translation in the same format in which the source language data was originally provided. This could include translations of text inside tables or figures, formatting requirements, and other such challenges. Thus, for human translators, the compatibility of a given CAT tool with the data formats, other CAT tools, and other resources they regularly use are of great importance. In this dissertation, like much other work in this space from the natural language processing community, we do not focus on data formatting issues, instead assuming that the text to be translated is already sentence-segmented and provided in text format.

Translation Memory

A translation memory (TM) is a collection of source language segments with corresponding target language translation, collected during or after the translation process (or after quality assurance is completed on translations). These segments often consist of sentences. Other times they are sentence fragments like list items, headers, or titles, and sometimes they are supersentential segments like paragraphs. From the perspective of a machine translation researcher, a translation memory is simply a parallel corpus or bitext.

The collection of translation memories serves an efficiency-related purpose for human translators and translation clients. Since many domains include data with repeated segments, it is often more efficient to reuse existing translations, rather than starting from scratch each time. In parliamentary text, one might often encounter standardized and repeated language like "Madam President, on a point of order." or "Resumption of the session" (both from European Parliament parallel text). In general, it would be appropriate to translate these consistently, so the translation memory serves a dual purpose: it saves time and it encourages consistency. Rather than translating these sentences from scratch, the CAT tool can interface with the translation memory, so that the target side text is presented to the translator each time that a source sentence contained in the translation memory is to be translated.

Even when no exact match to the source side of the translation memory can be found, the translation memory may still prove useful. Consider an example of a near-match, with only a one word difference between the source sentence to be translated and a sentence in the translation memory. Suppose we wish to translate "Mr. President, on a point of order." and while our translation memory does not contain that exact sentence, it does contain the sentence "Madam President, on a point of order.", so the system returns as a suggested translation "Frau Präsidentin, zur Geschäftsordnung." (perhaps also indicating the location of the mismatched word, as in Esplà, Sánchez-Martínez, and Forcada (2011)). It may be faster for the translator to produce the correct translation "Herr Präsident, zur Geschäftsordnung." by replacing "Frau" with "Herr" and removing the feminine "in" ending from "Präsidentin" than by translating the full sentence from scratch.[3]

This process of modifying the translation returned from the translation memory is called "post-editing" and is described in more detail in Section 3.2.2. The selection of these near-matches is often done using a so-called "fuzzy match" algorithm; individual CAT tools typically maintain proprietary fuzzy match algorithms, which are then thresholded before use (a high threshold allows only nearly identical segments to be used, while a low threshold would more often provide results, though they might not increase translator productivity as much). One example of such a fuzzy match algorithm would be to use the word-level edit distance between the sentence to be translated and sentences in the translation memory. When no satisfactory fuzzy match is returned from the translation memory, it may be appropriate to return machine

[3] In Knowles, Ortega, and Koehn (2018) we examine approaches to performing these types of corrections automatically, using machine translation.

translated output instead.

Translators may have different levels of trust in and attitudes towards machine translation (sometimes perceived as being intended to replace human translators) and translation memories, since the latter are fully human-generated, possibly even by the translator themselves (Heyn, 1996). There has been a range of work on this and on combining translation memories and machine translation from the machine translation community; this line of work also has close ties to the body of work on example-based machine translation (Dandapat et al., 2011; Federico, Cattelan, and Trombetti, 2012; Koehn and Senellart, 2010; Marcu, 2001).

Terminology

Certain translation domains (e.g., law, medicine, engineering, etc.) require knowledge of highly technical or domain-specific terminology in order to produce appropriate translations. Terminology management systems can assist translators in this area, by providing searchable records of terms with relevant information (translations, contexts, definitions, the source of the terminology management system entry, regional information,[4] etc.). Translators can use these systems themselves to look up translations, can have them automatically pre-translate words from the source data, or can add to the terminology systems themselves for future use. As mentioned above, consistency

[4] Information about regional usage may be particularly important for colloquial or metaphorical language. For example, Church and Gale (1991) note that the Canadian Hansards contain frequent hockey metaphors including the term "rondelle" ("puck"), whereas a European French resource only includes senses of "rondelle" related to round shapes.

is often required of human translators, and particular translation clients may have specific needs and desires about how certain words should be translated, which can be entered into a terminology management system. Terminology systems can also be augmented automatically, using large scale parallel corpora and automatic alignments (Barrière and Isabelle, 2011).

Corpus Analysis

In addition to using term banks, dictionaries, and terminology management systems, translators may benefit from additional information about potential translations of terms. This could include term frequency counts in monolingual or bilingual corpora or information about collocations in which those terms frequently occur. When choosing between multiple translation options, both collocations and wider context may be useful. In order to see translation options in a wider context, translators can access a bilingual concordancer. A bilingual concordancer takes a source term and its translation and returns sentence pairs from a parallel corpus or translation memory where the source side contains the source term and the target side contains the translation. By examining these sentence pairs, a translator can gain an understanding of senses of a particular translation (for example, the difference between “bank” as the side of a river and “bank” as a financial institution or even “bank” as a collection, as in “terminology bank”). Church and Gale (1991) describes early work in building sentence-level concordances for word sense disambiguation and Isabelle et al. (1993) show examples of the bilingual concordancer TransSearch. Subsequent studies examined how translators used the TransSearch tool, finding that translators are especially likely to use it for determining the translation of polysemous words (Simard and Macklovitch, 2005; Macklovitch, Lapalme, and Gotti, 2008). Wu et al. (2003) and Callison-Burch, Bannard, and Schroeder (2004) describe work on other bilingual concordancers.

Other Features

Many of the standard features of a word processor may also be incorporated into a CAT tool: spelling and grammar checkers, search-and-replace, thesaurus, and edit tracking, among others. Isabelle et al. (1993) describe some such work. This can also include handling of markup (Tezcan and Vandeghinste, 2011), or other approaches to automatically handling formatting. Rodriguez Vazquez, O'Brien, and Fitzpatrick (2017) evaluates CAT tools from an accessibility perspective (particularly focusing on blind users) and finds significant room for improvement in modern CAT tools.

CAT Tools and Workbenches

There have been a number of CAT tools and workbenches produced by both industry and academia, which implement some or all of the computer aided translation features described in this chapter. Hutchins (1998) provides a history of early work on CAT tools, which he calls translator's workstations or workbenches. Among the most commonly used industry tools are SDL Trados (`https://www.sdltrados.com/`), Wordfast (`https://www.wordfast.com/`), and MemoQ (`https://www.memoq.com/`). Open source tools include MateCat (Federico et al., 2014), CASMACAT (Alabau et al., 2014), and OmegaT (`https://omegat.org/`). MateCat and CASMACAT have shared academic origins. Lilt (`https://lilt.com/`) was specifically built with interactive and adaptive machine translation technologies in mind. In our user study (Chapter 6) we use CASMACAT, which allows us to collect keystroke, mouseclick, and timing information during the translation process.

3.2.2 Post-Editing

Post-editing (PE) is one of the simplest ways that human translators can interact with machine translation output. In post-editing, a human translator receives a source sentence and a machine translated version in the target language, which they correct and modify until it is a satisfactory translation of the source. Translators may also post-edit fuzzy matches from a translation memory, but the remainder of this section focuses primarily on post-editing machine translation output. The post-editing approach stands in contrast to translating a sentence unaided or with the aid of computer aided translation tools like bilingual concordancers or terminology banks; it involves a very different cognitive process on the part of the translator. This has been studied extensively in the literature, with focus on both translator productivity and effort.

Green, Heer, and Manning (2013) experiment with unaided human translation and find that post-editing is faster and produces higher quality translations. Plitt and Masselot (2010) had also observed similar results in their realistic production environment experiments, comparing post-editing to "traditional" translation (though they do not specify whether additional computer aided translation features were available to the translators). Comparing post-editing against computer aided translation with the assistance of translation memories, terminology databases, and other assistance through a CAT tool workbench, Läubli et al. (2013) observed translation productivity gains of between 15 and 20% when post-editing. Langlois, Simard, and Macklovitch (2016) present results of a study where translation students benefitted (in terms of efficiency) from post-editing machine translation output (based on systems trained for the specific domain of interest). Federico, Cattelan, and Trombetti (2012) augment a translation memory with machine translation output, and also find that this improves translator productivity. More recently, Sanchez-Torron and Koehn (2016) perform a user study to examine the relationship between phrase-based statistical machine translation system BLEU scores and translator productivity, and find that a 1 BLEU point increase results in a corresponding 3-4% translator speed increase in post-editing. Denkowski (2015) shows work on statistical machine translation adaptation and post-editing.

Koponen (2012) considers cognitive and technical effort in post-editing, finding that reordering during post-editing is more cognitively demanding than changing the form of otherwise correct words. Koponen (2016) provides a survey of additional work on effort.

In measuring translator productivity, a recurring theme is inter-translator variation. Federico, Cattelan, and Trombetti (2012) observed overall gains when using post-editing, but noted that the largest relative productivity gains were for slower translators. Plitt and Masselot (2010) reported similar trends. Koehn and Germann (2014) note that differences between translators are greater than the difference between the machine translation systems being compared. All of this suggests that computer aided translation tools are not "one-size-fits-all" and may instead need to be chosen to best fit a given translator's skills and needs.

In addition to human post-editing, there exists a body of research on automating this task (automatic post-editing; Chatterjee et al., 2018).

3.2.3 Interactive Translation Prediction

Interactive translation prediction has been known by many names: interactive machine translation (IMT, though that term usually covers a broader set of techniques), text prediction (Foster, Langlais, and Lapalme, 2002), interactive-predictive machine translation (Domingo, Peris, and Casacuberta, 2017), target-text mediated interactive translation prediction (Foster, Isabelle, and Plamondon, 1997), and others. In interactive translation prediction (an editing mode for translators interacting with machine translation), the human translator guides the translation process. The machine translation system provides suggestions (much like an "auto-complete" function), which the translator can accept if they approve. If the translator prefers a different translation, they can type the word of their choice, and the system will adapt and provide new suggestions that are appropriate given the translator's additions.

Early work on interactive translation prediction can be found in the TransType and TransType2 projects (Langlais, Foster, and Lapalme, 2000; Foster, Langlais, and Lapalme, 2002; Bender et al., 2005; Barrachina et al., 2009). Using statistical machine translation, interactive translation prediction can be performed by re-decoding constrained by the prefix (Green et al., 2014; Wuebker et al., 2016) or by searching for the prefix in the original search graph (Och, Zens, and Ney, 2003; Barrachina et al., 2009). Sanchis-Trilles et al. (2014) showed that interactive translation prediction could be as fast as post-editing in a user study with professional translators. The following theses deal extensively with issues of computer aided translation: Foster (2002) and Green (2014). Neural approaches were introduced in Knowles and Koehn (2016) and Wuebker et al. (2016) (concurrent work), and Domingo, Peris, and Casacuberta (2017) also proposed a neural approach.

This dissertation focuses on text-based translation, but there also exists work on various approaches to interaction modalities: Sanchis-Trilles et al. (2008) (mouse actions), Alabau, Sanchis, and Casacuberta (2011) (hand-writing), Barrachina et al. (2009) (speech), and Grissom II et al. (2014) (real-time simultaneous translation).

Chapter 4

Corpora and Models

This dissertation makes use of a number of publicly available corpora, especially parallel corpora, as well as some trained machine translation systems. Where those datasets and models are used in multiple chapters, we describe them in detail here. Within the chapters where they are used, we discuss issues specific to the particular use case, but refer the reader back to this chapter for implementation and parameter information. For models and datasets that are used in only one chapter, we describe those in detail in the chapter in question.

4.1 Corpora

4.1.1 WMT News

The WMT news translation shared task focuses on building translation systems suited for translating news data from large quantities of parallel corpora. Constrained submissions to this task use a shared set of training data resources, including Europarl (Koehn, 2005), Common Crawl (Smith et al., 2013), News Crawl, and various monolingual corpora. These resources can be found at the official shared task website for each year, for example: `https://www.statmt.org/wmt16/translation-task.html`. Over the years, the task has covered a range of language pairs. Combined, these corpora contain millions of lines of parallel text and hundreds of millions of tokens for many of the language pairs.

In addition to using training data from WMT (and models trained on WMT training data), we use the test sets from the WMT 2016 and 2017 news evaluation (Bojar et al., 2016; Bojar et al., 2017). These test sets were collected from online news sources. For a given language pair, half of the test data was originally in English, while half was originally in the other language in the pair. This data was then translated by professional translators.

4.1.2 European Medicines Agency (EMEA)

We use the European Medicines Agency (EMEA) parallel corpus,[1] consisting of sentence-aligned documents focusing on medical products, downloaded from OPUS (Tiedemann, 2012). The full corpus consists of documents in 23 European languages, though we only make use of the English–German data. The text data was extracted from PDF documents (including converting tables into text), and the data is subsequently sentence-aligned. Additional details concerning the corpus and its collection can be found in Tiedemann (2009).

The corpus contains high levels of domain-specific terminology and repetition. Each document describes a new medication, meaning that new documents often contain novel vocabulary. Other novel vocabulary items include highly-specific medical terminology; these tend to appear fewer times within the document. Certain documents contain primarily tables of dosage information.

4.2 Models

4.2.1 Phrase-Based German–English Model

In two instances, we use a phrase-based statistical machine translation model, trained using Moses (Koehn et al., 2007). The system we use is Johns Hopkins University's German–English submission to the WMT shared news translation task

[1] http://opus.lingfil.uu.se/EMEA.php

(Ding et al., 2016). It uses all available parallel and monolingual training data released for the WMT evaluations. As described in that work, the systems were trained with the following settings:

> "a maximum sentence length of 80, grow-diag-final-and symmetrization of GIZA++ alignments, an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield et al., 2013) used at runtime, hierarchical lexicalized reordering (Galley and Manning, 2008), a lexically-driven 5-gram operation sequence model (OSM) (Durrani et al., 2013) with 4 count-based supportive features, sparse domain indicator, phrase length, and count bin features (Blunsom and Osborne, 2008; Chiang, Knight, and Wang, 2009), a distortion limit of 6, maximum phrase-length of 5, 100-best translation options, compact phrase table (Junczys-Dowmunt, 2012), minimum Bayes risk decoding (Kumar and Byrne, 2004), cube pruning (Huang and Chiang, 2007), with a stack-size of 1000 during tuning and 5000 during test, the no-reordering-over-punctuation heuristic (Koehn and Haddow, 2009), a domain-weighted neural language model."

The model uses POS tags, morphological tags and Och clusters (Och, 1999) of size 50, 200, 600 for additional interpolated language models, operation sequence models, lexicalized reordering models, sets of sparse features, syntactic prereordering, and compound splitting. The feature function weights are optimized with k-best MIRA (Cherry and Foster, 2012) on the concatenation of the 2008-2014 test sets. With the exception of compound splitting, this model uses whole words (rather than the byte pair encodings used by the neural models). On the WMT 2016 news test data, it has a BLEU score of 34.5.

4.2.2 Edinburgh's 2016 Neural Models

The neural machine translation models described in this section use Nematus,[2] a fork of the DL4MT toolkit[3] by Cho (2015). The models use an RNN encoder-decoder with attention architecture with one layer encoders and decoders; a full description (and comparison to the Bahdanau, Cho, and Bengio (2015) model) can be found in Sennrich et al. (2017).

For German–English, we use the publicly available model[4] from the University of Edinburgh's 2016 WMT shared news translation task submission (Sennrich, Haddow, and Birch, 2016a). It was trained on all the available parallel data (a 115 million word parallel corpus including Europarl, News Commentary, and CommonCrawl) and a similar amount of synthetic parallel data that was generated by translating part of the monolingual news data (about 75 billion words of additional English monolingual data, including LDC Gigaword, monolingual news, and monolingual CommonCrawl) into German (Sennrich, Haddow, and Birch, 2016b). It uses byte pair encoding (Sennrich, Haddow, and Birch, 2016c) for a vocabulary of 90,000 words. When decoding with a beam size of 1, the German–English model has a BLEU score of 34.5.

For English–German, Czech–English, and English–Czech, we also use the publicly available models from the University of Edinburgh's 2016 WMT submission. They were trained on all available WMT parallel training data, along with back-translated

[2] `https://github.com/rsennrich/nematus/`

[3] `https://github.com/nyu-dl/dl4mt-tutorial/`

[4] `https://github.com/rsennrich/wmt16-scripts/`

data.[5] All of the models released are the best single models for their translation pair and direction.

4.2.3 Neural English–Spanish Model

For English–Spanish, we trained our own neural machine translation model, again using a Nematus (Sennrich et al., 2017) RNN encoder-decoder architecture with attention. We use these training parameters: vocabulary of size 50,000, word embedding layer size of 500, hidden layer size of 1000, batch size of 80, Adadelta optimizer (Zeiler, 2012), maximum sentence length of 50, and default learning rate of 0.0001. All other parameters are set to Nematus defaults. The system is trained on Europarl v7 (Koehn, 2005) and News Commentary v10 data,[6] which comprised the WMT 2013 training data for English–Spanish. This training set contains 3.95 million sentence pairs, over 102 million source tokens, and over 106 million target tokens. We preprocess the data using the standard preprocessing scripts: tokenization, truecasing, and byte pair encoding (Sennrich, Haddow, and Birch, 2016c). We used the WMT 2012 News Test data for validation. The system has a BLEU score of 29.79 (beam 12, less than 1 BLEU below the best score from WMT 2013) or 28.40 (beam 1) on the WMT 2013 test set.

[5] Backtranslations were also released publicly by the authors at `http://data.statmt.org/rsennrich/wmt16_backtranslations/`.

[6] `http://www.casmacat.eu/corpus/news-commentary.html`

Part III

Neural Interactive Translation Prediction

This section of the dissertation focuses on neural interactive translation prediction, one particular type of computer aided translation tool in which a translator produces a translation by using an "auto-complete" style interactive interface. We begin in Chapter 5 by describing the neural interactive prediction algorithm, comparing it against a phrase-based statistical machine translation approach, and analyzing why the neural interactive approach outperforms the phrase-based statistical approach. We show results for the neural approach on several language pairs and translation directions, all in simulation. In Chapter 6, we move from simulation to a user study, collecting information about human translator performance and satisfaction with an implementation of neural interactive translation prediction. We find that most of the translators in our study reacted positively to the tool, and more than half of them were faster (as compared to post-editing) when using it.

Chapter 5

Neural Interactive Translation Prediction

5.1 Introduction

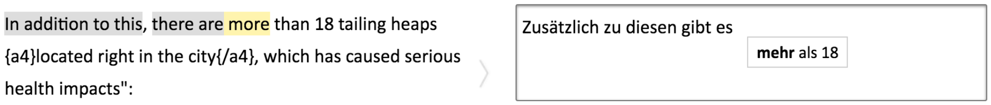
This section covers work published in the following publications: Knowles and Koehn (2016), Knowles and Koehn (2018b), and Knowles, Sanchez-Torron, and Koehn (2019).

Interactive translation prediction (also called interactive machine translation or target-text mediated interactive machine translation) is an editing mode for translators who interact with machine translation output. In this mode, the machine translation system makes suggestions for how to complete the translation ("auto-complete"), and the translator either accepts suggested words or writes in their own translation. When the suggestion is rejected, the machine translation system recomputes its prediction for how to complete the sentence from the given prefix and presents the corrected version to the translator. Implementations of interactive translation can be found in the CASMACAT[1] (see Figure 5.1) and Lilt[2] computer aided translation tools. This approach stands in contrast to the common practice of post-editing machine translation output. In post-editing, the translator receives complete machine translation output, which they then revise until it is both adequate and fluent. In interactive translation prediction, the translator drives the translation and can actively choose to guide the direction of translation or to accept suggestions from the machine translation system. There is evidence that this interaction mode is preferred by translators over post-editing (Koehn, 2009).

The goal of interactive translation prediction is to generate suggestions that the translator will accept. In prior work, phrase-based machine translation systems have been used for interactive translation prediction, and suggestions were made either by re-decoding constrained by the prefix (Green et al., 2014) or by searching for the prefix in the original search graph (Och, Zens, and Ney, 2003; Barrachina et al., 2009). As our baseline in this work, we use a statistical machine translation system for interactive translation prediction which follows Koehn (2009) and Koehn, Tsoukala, and Saint-Amand (2014). The system attempts to match the partial translator input (called the *prefix*) to the search graph, using approximate string matching techniques (minimal string edit distance) when an exact match cannot be found. Recently,

[1] http://www.casmacat.eu/

[2] https://lilt.com/



In addition to this, there are more than 18 tailing heaps {a4}located right in the city{/a4}, which has caused serious health impacts":

Zusätzlich zu diesen gibt es

mehr als 18

Figure 5.1: Interactive translation prediction in CASMACAT: The system suggests to continue the translation with the words *mehr als 18*, which the user can accept by pressing the TAB key.

neural translation models have been proposed and in some cases have shown superior performance over phrase-based models (Jean et al., 2015; Sennrich, Haddow, and Birch, 2016a). We propose to use such models for interactive translation prediction. Parallel to this work, Wuebker et al. (2016) also explore a similar approach to using neural MT for interactive translation prediction.

The decoding mechanism for neural models provides a natural way of doing interactive translation prediction. We show that neural translation models can provide better translation prediction quality and improved recovery from rejected suggestions. We also develop efficient methods that enable neural models to meet the speed requirements of live interactive translation prediction systems and demonstrate that they can successfully be used by real translators through a user study.

5.2 Neural Interactive Translation Prediction

The decoding process for the types of neural translation models described in Section 2.4 operates by generating one token at a time, from the beginning of the sentence to the end, each conditioned on the previously generated tokens. This leads naturally to a simple implementation of interactive translation prediction in the neural setting. Instead of using the model's predictions in the conditioning context for the next step, the tokens in the prefix provided by the translator can be used. Hence, the next token prediction is conditioned on the choice of the translator, rather than the prediction of the model.

During decoding for translation as described in Section 2.4, the model's predictions $\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}$ are fed back into the model to produce the next predicted token, with the probability of generating y_t defined as:

$$p(y_t|\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}, \vec{x}) = g(\hat{y}_{t-1}, c_t, s_t) \tag{5.1}$$

In order to do interactive prediction, we instead feed the true prefix $\{y_1^*, \cdots, y_{t-1}^*\}$ produced by the translator back into the model. This is actually quite similar to training, where a human-produced (reference) translation is used. Thus we redefine the conditional probability of generating a token y_t to be:

$$p(y_t|\{y_1^*, \cdots, y_{t-1}^*\}, \vec{x}) = g(y_{t-1}^*, c_t, s_t) \tag{5.2}$$

During standard decoding for translation (without an interactive component), beam search is often employed in order to produce higher quality output. In our simulations, we consider two variations on neural interactive translation prediction, with and without beam search:

- The **no beam search** (greedy) method produces the single best hypothesis for each new token, given the prefix provided by the translator, which is fed into the model during decoding (as described above).
- The **beam search** method force-decodes the prefix provided or validated by the translator and then performs beam search to select the most probable full translation of the sentence. In contrast to standard beam search for translation, where many or all of the hypotheses may differ from one another, all hypotheses in neural interactive translation with beam search will share the same prefix (provided by the translator), but may differ from one another at any point after that. We show results for beam size 12, but note that a beam size of 2 provides most of the improvement (a similar observation was made by Sutskever, Vinyals, and Le (2014) with respect to standard MT evaluation).

While beam search is known to produce better translation quality than models without beam search, it is more computationally expensive. Additionally, the standard application of beam search requires translating to the end of the sentence, whereas the approach without beam search could be used to generate just the subsequent n tokens if needed. We demonstrate that full beam search performs well on the interactive translation prediction task, but note that it is too slow for use in a live system.

Despite its similarities to training, passing the translator prefix into the system (when the translator diverges from the predicted sequence) may produce subsequent errors in the translation, for instance by causing the attention mechanism to be misfocused. We show that the system is often able to recover from these errors, but that it occasionally results in incoherent sequences of suggestions. However, we show that the sequences of errors produced by the neural systems tend to be shorter than those produced by the traditional search graph based systems.

5.3 Related Work

The following sections provide more technical detail on two approaches for statistical machine translation systems: search graph decoding (Section 5.3.1) and prefix-constrained decoding (Section 5.3.2). We also discuss additional related work on neural interactive translation prediction.

5.3.1 Search Graph Decoding

Och, Zens, and Ney (2003) propose an approach to interactive translation prediction that makes use of the kind of search graph (which they call a word hypothesis graph,[3] and describe as a subset of the search graph) produced by a word- or phrase-based statistical machine translation system. Such a directed acyclic graph $G = (V, E)$ has vertices $v \in V$ that represent partial translations, with each edge $(v, v') \in E$ associated with a target language unit (a word or phrase) and its score (as a combination of language model and translation model probability). Forward and backward probabilites can be computed to each vertex v; these can be used with the edge-associated probabilities to determine the most probable suggested completions of the sentence for any given prefix v in the graph. However, a human translator may prefer a prefix that is not found in the graph (for example, by adding an out of vocabulary token or by choosing a token that was simply not found in the hypothesis graph). They handle this by performing a "tolerant search" – first they find vertices with low edit distance to the desired prefix, then select from those the highest probability vertex and use its most probable completion as the suggestion to the translator. They built this prototype as part of the TransType2 project, and found in simulation that presenting full sentence completions had the potential to save translators more keystrokes than presenting single-word completions. In addition to providing the one-best suggested completion of a sentence, one could also view multiple suggestions. Barrachina et al.

[3] Ueffing, Och, and Ney (2002) provide additional information on this.

(2009) also discuss this approach, and mention reordering multiple hypotheses in order to present maximal diversity at the beginning of suggested sentence completions.

The approach that we use as the phrase-based statistical machine translation baseline follows this search-graph based approach, with refinements as described in Koehn, Tsoukala, and Saint-Amand (2014). That work uses the core algorithm described in Koehn (2009), computing minimal costs to reach vertices using dynamic programming. The main refinements are prioritizing matching the last word in a prefix and other improvements to approximate word matching (using edit distance to penalize small changes less than major ones, and stemmed matching to do the same).

In these approaches, heuristics are required to handle the case of "falling off the search graph," while the neural approach does not face the same challenge as it maintains a probability distribution over the full vocabulary at all times.

5.3.2 Prefix-Constrained Decoding

Green et al. (2014) describes prefix decoding for cube pruning (Chiang, 2007). Hypothesis translations are required to match the user prefix, and then search is conducted as usual. They note that the pop limit must be suspended until each beam contains at least one translation that could meet this matching constraint, if possible.

In Wuebker et al. (2016), the authors propose target beam search, in which beams are associated with target word counts rather than source word counts, and in which long-range reorderings are allowed. They also add synthetic phrase pairs to the phrase table by generating automatic alignments between the source sentence and the translated prefix. This addition addresses one issue faced by the search graph approaches (resolved in those approaches via search that is tolerant of mismatches), by making sure that the phrase table always includes all of the target prefix words with some alignment to the source.

5.3.3 Neural Approaches

The neural machine translation approach to interactive translation prediction is a very natural extension of standard neural machine translation decoding, and has been presented in several publications. Contemporaneous with Knowles and Koehn (2016), Wuebker et al. (2016) also proposed the approach to interactive neural machine translation described in Section 5.2. They compare it to a prefix-constrained system that is tuned to interactive translation specific objectives and find in simulation that the neural approach outperforms the phrase-based approach but is slower by two orders of magnitude. Peris, Domingo, and Casacuberta (2017) describes this same approach to interactive translation prediction, and also proposes a method for a different interactive technique, in which the human translator validates potentially discontiguous segments of a full translation (rather than validating a prefix), after which the system returns a new translation that keeps all of the validated segments intact. Lam, Kreutzer, and Riezler (2018) propose a reinforcement learning based form of neural interactive translation, wherein the system produces partial translations, requests human quality judgments (rather than corrections) when entropy suggests the system is uncertain, and updates the model after each such interaction; as with the rest of the approaches described here, they evaluate it in simulation. Two of our papers (Knowles and Koehn, 2016; Knowles, Sanchez-Torron, and Koehn, 2019) discuss the use of probability distribution masking to allow character-level interactions with a subword vocabulary in interactive translation prediction.[4] Peris and Casacuberta (2019) also describe this same masking technique.

5.4 Simulation Experiment Setup

We begin with a simulation study, where a preexisting human translation is used in place of the translator's live input to an interactive translation system. We do this by treating the preexisting human translation as though it were being typed live, one token (or letter) at a time, by a translator interacting with a prediction system. Algorithm 1 shows the approach to simulating neural interactive translation prediction. The phrase-based statistical machine translation baseline follows the same outline, using the search graph to predict the next token. We perform the simulation exhaustively (predicting each token given all true token prefixes in the sentence) and compute metrics from this exhaustive output rather than sampling a subset of sentence prefixes for examination.

While we expect that in practical use, the human translator may sometimes match

[4] This is described more extensively in Section 5.7.1.

the machine translation's suggestions more closely (for example, by accepting synonyms which we score as "wrong" as they are not exact matches) or may diverge more than any one given reference sentence does, we can nevertheless compare methods based on their prediction accuracy on the human reference relative to one another.

Algorithm 1 Simulated Neural Interactive Translation Prediction

```
$\vec{x}, \vec{y^*}$ : source language sentence and reference (target language) translation
$x_t, y^*_t$ : token $t$ of $\vec{x}$ and $\vec{y^*}$, respectively
$T$ : length of $\vec{y^*}$ (number of tokens)

***Approach to predicting next token:***

function PREDICT($\vec{r}, \vec{x}, b$)
    if $b = 1$ then
        ▷ Perform greedy (no beam) decoding to generate the next token, given
          the reference prefix ($\vec{r}$, length $t-1$) and source sentence ($\vec{x}$).
        $\hat{y}_t \leftarrow \arg\max_{y_t} p(y_t|\vec{r}, \vec{x})$
    else
        ▷ Perform beam search (beam size $b$) to translate the sentence to completion, selecting the most probable full sentence completion: $\{\widetilde{y}_t, \cdots, \widetilde{y}_{T'}\}$,
          while keeping the reference prefix ($\vec{r}$) fixed. Then select the first token
          of the sentence completion $\widetilde{y}_t$ to be the prediction.
        $\hat{y}_t \leftarrow \widetilde{y}_t$
    end if
    return $\hat{y}_t$
end function

***Process of simulating interactive translation prediction:***

$\hat{y}_1 \leftarrow$ PREDICT($\{\}, \vec{x}, version$)
for $t \in \{2 \ldots T\}$ do
    $\hat{y}_t \leftarrow$ PREDICT($\{y^*_1, \cdots, y^*_{t-1}\}, \vec{x}$)    ▷ Predict next token given reference prefix.
end for

▷ Compute word and character prediction accuracy by comparing $y^*_t$ and $\hat{y}_t$ for
$t \in \{1, \ldots, T\}$.
```

5.4.1 Data

We perform a direct comparison between neural and statistical machine translation-based interactive translation prediction on the German–English datasets[5] made available for the news translation shared task at the 2016 Conference on Machine Translation (WMT), as described in Section 4.1.1. We use the official 2999 sentence test set (average sentence length 23 tokens) to measure the performance of our methods. This language pair and direction was chosen for this comparison on the basis of the existence of neural and phrase-based statistical machine translation systems of very comparable quality.

We also show neural interactive translation prediction results for German–English, English–German, Czech–English, and English–Czech on the 2017 WMT news test sets. For Czech language data, the test sets contain 3005 sentence pairs and for German language data, 3004. For English–Spanish, we show results on the 2013 WMT test set, consisting of 3000 sentence pairs. Those datasets are also discussed in Section 4.1.1.

5.4.2 Phrase-Based Model

As a baseline against which to compare neural interactive translation prediction, we begin with a phrase-based statistical model which can be used to produce search graph based predictions in an interactive translation prediction setting. We use the German–English system submitted by Johns Hopkins University to the 2016 WMT

[5] `http://www.statmt.org/wmt16/`

shared task (Ding et al., 2016), as described in Section 4.2.1, which has a BLEU score of 34.5 on the WMT 2016 news test set. This model does not use the byte pair encodings used by the neural models.

5.4.3 Neural Translation Models

For German–English, we use the Nematus model released by the University of Edinburgh (Sennrich, Haddow, and Birch, 2016a), which also scores 34.5 on the WMT 2016 news test set when decoding with a beam size of 1. The system uses byte pair encoding with a vocabulary of 90,000 words. For English–German, Czech–English, and English–Czech, we also use the publicly available models from the University of Edinburgh's 2016 WMT submission. All of these systems are described in more detail in Section 4.2.2.

We also train our own English–Spanish neural machine translation system, also using the Nematus toolkit. As described in more detail in Section 4.2.3, the system is trained on WMT data and evaluated on WMT 2013 news test data (the most recent year in which the language pair was included in the evaluation). The system has a BLEU score of 29.8 (beam 12, less than 1 BLEU below the best score from WMT 2013) or 28.4 (beam 1) on the WMT 2013 test set.

System	Configuration	BLEU
Neural	**no beam search**	**34.5**
	beam size 12	36.2
	+ ensemble	37.5
	+ r2l reranking	38.6
Phrase-based		**34.5**

Table 5.1: Quality measured by BLEU scores (case-sensitive) on the WMT 2016 news test set for both the phrase-based and neural German–English models.

5.4.4 System Quality

Without beam search, the German–English neural system used has the same BLEU score as the phrase-based system on the WMT 2016 test set (Table 5.1). While these identical BLEU scores do not guarantee or even imply that the systems make the same kinds of errors or that they would be judged by human annotators to be of identical quality, we can claim that the systems are of *comparable* quality. For this reason, we choose to compare neural and phrase-based interactive translation prediction on the German–English language pair, allowing us to make a stronger claim about the fact that neural interactive translation outperforms phrase-based than we would be able to make if the underlying neural translation system were clearly of higher quality than the phrase-based one.

Since we are concerned with translation speed, we consider a few simplifications of the neural translation model. We do not use ensemble decoding ("ensemble") or a reranking stage ("r2l reranking").[6] Each of these simplifications makes decoding several times faster at a cost to quality of 1-2 BLEU points. See Table 5.1 for a

[6] For more on these methods, see Sennrich, Haddow, and Birch (2016a).

Input: *Das Unternehmen sagte, dass es in diesem Monat mit Bewerbungsgesprächen beginnen wird und die Mitarbeiterzahl von Oktober bis Dezember steigt.*

	Correct	**Prediction**	**Prediction probability distribution**
✓	the	the	**the (99.2)**
✓	company	company	**company (90.9)**, firm (7.6)
✓	said	said	**said (98.9)**
✓	it	it	**it (42.6)**, this (14.0), that (13.1), job (2.0), the (1.7), ...
✓	will	will	**will (77.5)**, is (4.5), started (2.5), 's (2.0), starts (1.8), ...
✓	start	start	**start (49.6)**, begin (46.7)
	inter@@	job	job (16.1), application (6.1), en@@ (5.2), out (4.8), ...
✘	viewing	state	state (32.4), related (5.8), **viewing (3.4)**, min@@ (2.0), ...
✘	applicants	talks	talks (61.6), interviews (6.4), discussions (6.2), ...
✓	this	this	**this (88.1)**, so (1.9), later (1.8), that (1.1)
✓	month	month	**month (99.4)**
✘	,	and	and (90.8), **, (7.7)**
✘	with	and	and (42.6), increasing (24.5), rising (6.3), **with (5.1)**, ...
✓	staff	staff	**staff (22.8)**, the (19.5), employees (6.3), employee (5.0), ...
✘	levels	numbers	numbers (69.0), **levels (3.3)**, increasing (3.2), ...
✘	rising	increasing	increasing (40.1), **rising (35.3)**, climbing (4.4), rise (3.4), ...
✓	from	from	**from (97.4)**
✓	October	October	**October (81.3)**, Oc@@ (12.8), oc@@ (2.9), Oct (1.2)
✘	through	to	to (73.2), **through (15.6)**, until (8.7)
✓	December	December	**December (85.6)**, Dec (8.0), to (5.1)
✓	.	.	**. (97.5)**

Figure 5.2: Example with nearly average prediction accuracy. Note the good recovery from failure and that several of the correct choices rank highly in the probability distribution of predicted words (values in parentheses indicate percent of probability mass assigned to words; only words with probability $\geq$1% are shown). Tokens containing @@ are an artifact of byte pair encoding.

comparison of quality scores for the different settings. There is clear potential for improvement if computational concerns are removed. We discuss more issues of speed and practical implementation in Section 5.7.

5.5 Simulation Results

Figure 5.2 provides an example of the neural interactive translation prediction system's output for one sentence. The figure displays the correct word choices (taken from the reference translation), the model's prediction (using the prefix of the reference translation up to that point as conditioning context), and the most probable word choices according to the model's probability distribution. In this example, we see some instances where the token predicted by the system is a synonym of the reference token (e.g., *rising* and *increasing*), a replacement that a human translator (rather than our reference simulation) might actually accept in practice. We measure simulated performance of the interactive translation prediction systems according to exact matches with the reference, but we also examine the frequency of these and other types of errors. Note that all predictions are made with the correct history, either because it was correctly generated by the system or because the correct tokens were force-decoded.

The neural method copes well with failure, and typically resumes with plausible predictions. One exception is the prediction of *talks* after having seen *... will start interviewing*. This may be due to the attention mechanism being thrown off after a sequence of low-probability prefix words, the way that byte pair encoding segments the German compound noun *Bewerbungsgesprächen*, or other reasons. We provide additional analysis of all of these issues in Section 5.6.

System	Configuration	Word Prediction Accuracy
Neural	no beam search	61.6%
	beam size 12	63.6%
Phrase-based	-	43.3%

Table 5.2: Word prediction accuracy: Percentage of words predicted by the interactive translation prediction system that matched the human reference translation exactly for German–English on the WMT 2016 news test set.

5.5.1 Word Prediction Accuracy

Figure 5.2 also illustrates word prediction accuracy, the primary evaluation metric we use to measure the quality of the interactive prediction methods. It measures how many words are predicted correctly (see the first column in the figure). Note that we measure on the level of words, so we score the split form *inter@@ viewing* (an artifact of byte pair encoding) as a single word, rather than as two words. This allows us to directly compare the neural and phrase-based systems despite the fact that they use different segmentations (the former translates and predicts tokens at the subword level, while the latter predicts tokens at the whole word level). For both the neural and phrase-based interactive prediction systems, we generate a sequence of predictions such that the prediction $\hat{y}_t$ is based on the human validated prefix $\{y_1^*, \cdots, y_{t-1}^*\}$ for all t. We can then compute the word prediction accuracy as the percentage of the predictions $\hat{y}_t$ that exactly match their corresponding reference y_t^*.

Table 5.2 shows the prediction accuracy for the three methods. The neural systems clearly outperform the method based on the search graph of the phrase-based model with over 60% prediction accuracy for the neural systems and just 43.3% for the

Language Pair	**WPA**	**BLEU**
English→German	60.7%	24.2
German→English (2016)	61.6%	34.5
German→English	62.7%	29.6
English→Czech	56.1%	19.1
Czech→English	57.0%	24.5
English→Spanish (2013)	59.1%	28.4

Table 5.3: Word prediction accuracy (WPA) of neural interactive translation prediction with beam size 1 and BLEU score for standard neural machine translation decoding with beam size 1 on WMT 2017 test set (unless otherwise noted).

phrase-based. The difference between the beam search approach (63.6%) and the no beam search approach (61.6%) is much smaller than the difference between those approaches and the phrase-based approach, despite their corresponding BLEU scores (Table 5.1). We discuss potential reasons for this improvement in Section 5.6.1.

We also show word prediction accuracy and BLEU scores for several other language pairs and datasets in Table 5.3. For both German and Czech, we observe slightly higher word prediction accuracies when translating into English than when translating from English; in both cases this matches the intuition that it is more challenging to translate from a less morphologically complex language to a more morphologically complex one. In all cases, the word prediction accuracy is above 50%, even without beam search.

5.5.2 Letter Prediction Accuracy

In interactive translation prediction, the user can interact at either the whole-word level (accepting suggestions) or at the character level (providing corrections). This means that the interactive translation prediction system must be able to react to a single user keystroke. Returning to the example from Figure 5.2, we observe that the system assigns higher probability to the word *numbers* than to the word *levels*, while the latter is preferred by the human translator (or, at least, the human translator who produced this particular reference). This is marked as an error for the purpose of calculating word prediction accuracy. Now we can imagine this occurring in a real translation setting: if the user types the letter *l*, the system should quickly update its prediction to *levels*, the most likely token (from the probability distribution that generated the original hypothesis) that starts with this letter. In general, when the user types the initial letters of a word, the system should predict the most probable word with this letter prefix. In the beam search setting, the system first runs through the first word of each of the hypotheses in the beam (from most to least probable) to see if any match the translator's letter prefix, before falling back to the probability distribution over the full vocabulary. With a beam size of 12, the correct word appears in the beam (but not as the predicted word) 25.2% of the time. We discuss the particulars of implementation in Section 5.7.1.

To measure the accuracy of system predictions for word completion, we count the number of incorrectly-predicted characters. To give a more complex example, suppose

System	**Configuration**	**Letter Prediction Accuracy**
Neural	no beam search	86.8%
	beam size 12	87.4%
Phrase-based	-	72.8%

Table 5.4: Letter prediction accuracy: Percentage of letters predicted correctly for German–English on the WMT 2016 news test set.

that the human translator wants to use the word *increased* at the point at which the system first predicts *rising*. After seeing the letter *i*, the system updates its prediction to *increasing*. It predicts all letters correctly until it comes to the final *e*. When the user enters *increase*, the system updates its prediction to *increased*. We count this as two wrongly predicted letters: **i**_ncreas_**e**_d_. Table 5.4 shows the scores for both the neural and phrase-based methods. Again, the neural methods clearly outperform the phrase-based method.

Note that this measure is not as clearly tied to user actions as word prediction accuracy. In the user interface shown in Figure 5.1, correctly predicted words are accepted by pressing TAB, while incorrectly predicted words have to typed in completely (assuming no word completion). So, word prediction accuracy reflects the proportion of words that do not have to be typed in. The effort savings for word completion are less clear, since there are various ways the user could interact with the system. In our example, when the user sees the prediction *increasing* but wants to produce *increase*, there are several choices even within the user interface of a computer aided translation tool like CASMACAT. The user could accept the system's suggestion, and then delete the suffix *ing* and type in *ed*. Or, they could type in the entire prefix *increase* until

System	**Configuration**	1	2	3	4	5
Neural	no beam search	55.9%	61.8%	61.3%	62.2%	61.1%
	beam size 12	58.0%	62.9%	62.8%	64.0%	61.5%
Phrase-based	-	28.6%	45.5%	46.9%	47.4%	48.4%

Table 5.5: Ratio of words correct after first failure for German–English on the WMT 2016 news test set. The columns (numbered 1 through 5) indicate the position of the word relative to the first failure in the sentence, with 1 being the word immediately following the first failure and 5 being the word five tokens after the first failure.

the system makes the correct prediction, which in this example does not yield any savings at all: the user may accept the prediction with TAB or type in *d* on their own.

5.6 Analysis

Having observed that neural interactive translation prediction outperforms a phrase-based approach using the search graph when the underlying machine translation quality is similar, we consider some of the reasons why this is the case. We first examine how well each of the systems recovers from errors (Section 5.6.1), then consider the length of the sequences of erroneous predictions (Section 5.6.2), and conclude with additional analysis of the probability assigned to synonyms or other erroneously predicted words (Section 5.6.3).

5.6.1 Recovering from Failure

Let us imagine a human translator interacting with an interactive translation prediction system, working to translate the sentence from Figure 5.2: *Das Unternehmen* *sagte, dass es in diesem Monat mit Bewerbungsgesprächen beginnen wird und die Mitarbeiterzahl von Oktober bis Dezember steigt.* To start, the interactive translation prediction system performs well, and the translator accepts the first six words (*The company said it will start*) suggested by the system. However, the next word that the system suggests is *job*, but the translator would prefer the word *interviewing* instead. They would then produce the word *interviewing* themselves, the interactive translation prediction system would adjust, and they would continue interacting by either accepting suggestions or correcting the system with their own preferred words. We call this misprediction (*job* instead of *interviewing*) the first failure in the sentence.

To get a more detailed picture of the performance of the neural prediction method, we explore how it recovers from failure. That is, after the translator rejects a suggestion and provides their own (currently simulated by a reference), we look at what the system does the next time(s) that it predicts a word. First, how well does the method predict the words following its first failure? We look at a window of up to five words following the first failure in a sentence (note that if the first failure is near the end of the sentence, the window will be truncated to the end of the sentence). In the case of our example, this would involve looking at the 5 system suggestions and simulated translator interactions (acceptance or correction) following the erroneous suggestion of *job*. The next suggestion is incorrect (*applicants*), followed by two correct suggestions (*this* and *month*), then two incorrect suggestions (*,* and *with*); these would be the 5 words in the window following the first failure. See Table 5.5 for performance of the

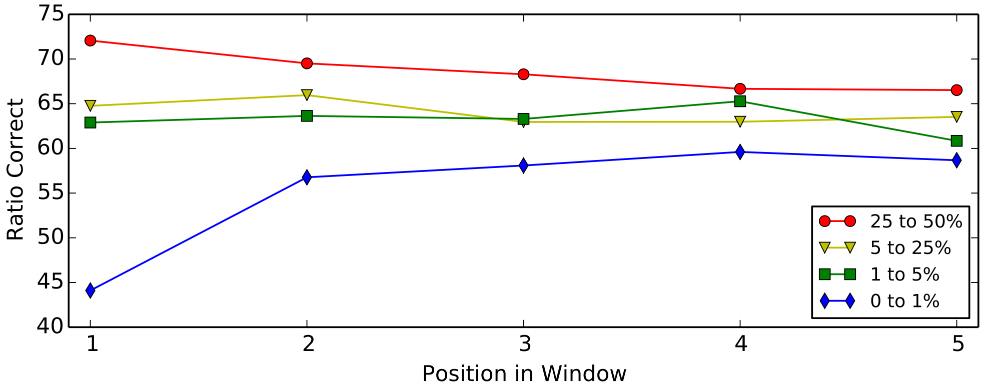


Figure 5.3: Neural (no beam) recovery from first failure at each position in the window of 5 words following the first failure, binned by probability assigned to correct solution (see legend) for the neural German–English system without beam search on the WMT 2016 news test set.

various interactive translation prediction systems.

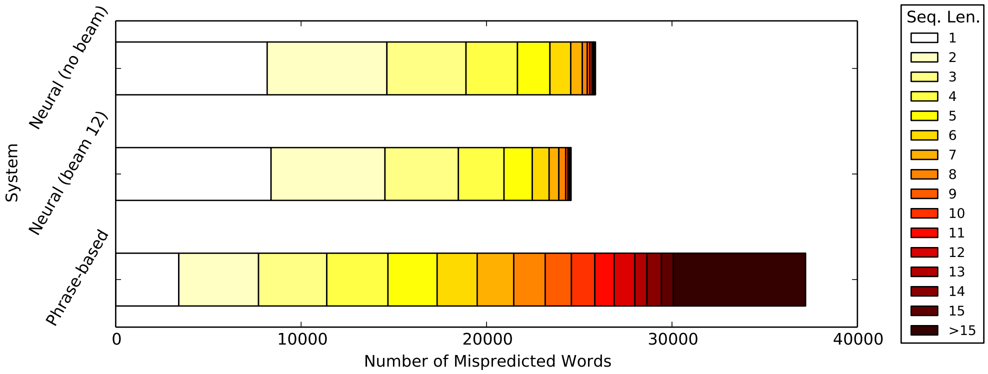
The neural system is successful in predicting the word following the first failure in the sentence (the first word in the window) 55.9% of the time. The second word in the window is predicted correctly 61.8% of the time, with similar accuracy for the remainder of the winder. So, failing on a word does impact the prediction of the word immediately following the failure, but it has less of an impact on words in the rest of the window following the failure. The phrase-based method only correctly predicts 28.6% of the first words immediately after failing on a word, a larger drop. This suggests that the phrase-based method has a harder time recovering initially, though for both types of system, the percentage predicted correctly does return to or exceed the overall word prediction accuracy within the five word window.

Interestingly, not all failures have an equal impact on the predictability of the subsequent words. Figure 5.3 shows the prediction accuracy for the neural method (without beam search) in more detail for the same five word window following the system's first error. We examine the way that the probability assigned to the correct word (which the model failed to predict) influences recovery from errors. When the model assigns extremely low probability (below 1%) to the correct answer, it performs very poorly on the next word, getting it correct only 44.1% of the time. On the other hand, when the model assigns relatively high probability to the correct word (25% to 50%), the probability of correctly guessing the next word rises to 72.1%. We can intuitively understand the probability assigned to the correct word as approximating how close the model was to being correct when it made the first error. When its prediction is far from correct, it has difficulty recovering, but when it is close to correct, it does not suffer a drop in performance in predicting the next words.

We observe examples of this phenomenon (and its ties to near-synonyms) in Figure 5.2. When the model assigns low probability to the correct answer (e.g., *interviewing*), there are sequences of incorrect predictions. In the case of *rising*, the model predicts *increasing*, a near-synonym, and assigns the highest probability to *increasing*, *rising*, and *climbing* (in descending order).

5.6.2 Length of Sequences of Mispredicted Words

Another revealing set of statistics is the length of sequences of word prediction failures. If the method fails on one word, and predicts the next word correctly,



System	**Config.**	1	2	3	4	5	6	7	8	9	10+
Neural	no beam	8168	3229	1422	694	350	187	89	33	16	25
	beam 12	8378	3072	1320	615	305	151	75	46	14	15
Phrase	-	3403	2150	1227	825	530	360	282	212	157	774

Figure 5.4: The graph shows number of mispredicted *words*, categorized by lengths of the sequence of mispredicted words to which they belong. The table gives a breakdown of the number of *sequences* of each length.

we have a 1-word failure sequence. However, if it misses the next word also and only recovers after that, we have a 2-word failure sequence, and so on. Shorter failure sequences indicate better models (and ideally a correspondingly improved user experience). Figure 5.4 visualizes the sequences of word prediction failures by showing how many mispredicted words can be accounted for by each failure sequence length (mispredictions in shorter sequences are represented by light colors while mispredictions in long sequences are shown in darker red).

The methods show a stark contrast. The neural methods have a much higher number of 1-word failure sequences (8168 and 8378 vs. 3403) and 2-word failure sequences (3229 and 3072 vs. 2150, comprised of 6458 and 6144 vs. 4300 mispredicted words) but comparably very few long failure sequences. For instance, only a small fraction of the neural systems' mispredicted words occur in sequences of greater than 15 errors in a row, while 7129 of the phrase-based system's word prediction errors occur in misprediction sequences of length greater than 15 words. This is not simply a consequence of the greater word prediction accuracy of the neural systems; in particular, the phrased based model shows far more long misprediction sequences than one would expect were those errors distributed uniformly randomly (the neural systems also have more long misprediction sequences than would be expected if the errors occurred randomly, but to a lesser extent).

These numbers again suggest that the neural method recovers much better from failures, while the phrase-based system has more difficulty. Since the neural method considers every word in the vocabulary at any point in the sequence, it can always place the user's choice in a word prediction sequence, and does not have to resort to string edit distance to match up with the user's translation.

5.6.3 Synonyms

In both the example sentence (Figure 5.2) and the analysis of error recovery (Section 5.6.1), we observe interesting behavior when the model makes an error while still assigning relatively high probability to the correct token. Some of the failures are near-synonyms (*numbers* instead of *levels*, or *increasing* instead of *rising*) that we might expect would be accepted by real human users of the system. For the purposes of our evaluation, we count even these near-synonyms as incorrect (as they are not exact string matches).

For English, we can use WordNet (Fellbaum, 1998) as a resource for automatically determining synonymy. Words in WordNet are related to one another in terms of synonymy, hypernymy, hyponymy, meronymy, and other semantic relations. For our purposes in this section, we define words to be synonyms if their Wu-Palmer similarity (Wu and Palmer, 1994) in WordNet is equal to 1. This Wu-Palmer similarity between words x and y is computed using their least common superconcept c as $\frac{2*d(c)}{d(x)+d(y)}$, where d is the depth (number of nodes on the path to the root). We use the NLTK (Bird, Klein, and Loper, 2009) implementation.

It is worth noting the prevalence of these synonyms and near-synonyms: in the neural versions, we find that 21.0% of incorrect predictions (22.4% with beam search) are synonyms of the correct answer; in the phrase-based system, this drops to 17.7%. Were these to be accepted by a real translator, overall system accuracy scores would improve. We hypothesize that the difference between the prevalence of synonyms in the neural and phrase-based approaches could be partly due to the neural system having additional information about the semantics of words (as represented by their embeddings), while the search graph system treats synonyms and non-synonyms alike.

5.7 Speed Considerations

Having shown that the neural method delivers superior prediction accuracy, we turn to the issue of speed for use in a live system and address some details of implementation. To be used in an interactive user interface, the method has to quickly produce alternative sentence completion predictions. A common time limit in human computer interaction is 100 milliseconds for the system's response. Any longer feels sluggish and annoying to the user.

5.7.1 Implementation Details

In the initial description of neural interactive translation prediction, we showed the conditional probability equation for generating the single token immediately following a translator's prefix (Eq. 5.2). In practice, however, we generate more than just the next predicted token to show to the translator, so it is more accurately described as follows: given a translator prefix of length m, and some number of new tokens which we wish to show to the translator, we have two equations.

$$p(y_{m+1}|\{y_1^*, \cdots, y_m^*\}, \vec{x}) = g(y_m^*, c_t, s_t) \tag{5.3}$$

$$p(y_{m+n}|\{y_1^*, \cdots, y_m^*, \hat{y}_{m+1}, \cdots, \hat{y}_{m+n-1}\}, \vec{x}) = g(\hat{y}_{m+n-1}, c_t, s_t) \forall n > 1 \tag{5.4}$$

In Equation 5.3, we see that the word immediately following the user-generated prefix is conditioned on the user-generated prefix. In Equation 5.4, we see that all subsequent words are conditioned on a user-generated prefix followed by predicted words (until such time as the translator accepts or rejects them).

If a translator rejects a suggestion and provides their own, there are two possible cases: either the translator has added a complete word to the translation, or they have added a partial word. In the case of a complete word, we follow Equations 5.3 and 5.4. That word becomes part of the prefix, and the generation of the subsequent tokens is conditioned on it.

If, however, the translator has only generated a partial word (which we will call a character prefix), this is slightly more complicated. We provide some additional technical detail here. We must first determine whether this character prefix is the prefix to any item in our (subword) vocabulary. If it is the prefix of at least one vocabulary item, we predict the completion to this word (or subword) by selecting the highest probability item in the vocabulary that starts with our character prefix (this can be described as a modification to the softmax and/or as a mask applied to the distribution prior to performing the softmax). Given the character prefix r^*:

$$p(y_t|\{y_1^*, \ldots, y_{t-1}^*, r^*\}, \vec{x}) \propto \mathbb{1}(y_t) p(y_t|\{y_1^*, \ldots, y_{t-1}^*\}, \vec{x}) \tag{5.5}$$

where

$$\mathbb{1}(y_t) = \begin{cases} 1 & \text{if } y_t \text{ starts with the string } r^* \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

We then continue predicting the remaining tokens in the standard fashion.

In the case that the character prefix is *not* the prefix to any item in our vocabulary, we must first apply BPE to it.[7] Once BPE has been applied, we have the model consume (forced decode) all but the last subsegment. This last subsegment could be a complete vocabulary item on its own, or again a prefix to a vocabulary item. Thus we return to our approach of predicting the highest probability vocabulary item which has the last subsegment as a prefix, and then continue prediction.

5.7.2 Speed Measurements

In a basic setup, the neural machine translation decoder has to step through the user's prefix, and then produce predicted words until the end of the sentence. In other words, it has to translate the entire sentence for a response to a user interaction. Table 5.6 gives numbers for decoding speed, running on a multi-core CPU (32 core 3.20GHz Intel Xeon CPU E5-2667 v3, although only 2-3 cores are utilized on average) and a GPU (Tesla K80).

Decoding time is spent mostly on the matrix multiplications to compute hidden

[7] This has potentially interesting consequences, as the BPE segments produced here may not be the ones that would have been produced had the translator produced the entire vocabulary item in one go. That is, applying BPE to a prefix (and then a suffix) will not always result in the same segmentation of a word as applying BPE to the whole word at once.

Length	1-4	5-9	10-14	15-19	20-24	25-29	30-34	35-39	100-104
CPU	108.6	115.7	122.7	127.0	131.3	136.1	140.7	145.2	184.4
GPU	7.0	7.2	7.4	7.4	7.4	7.4	7.6	7.6	7.6

Table 5.6: Decoding speed per word in milliseconds (neural model, no beam search) for different sentence lengths.

and output layer vectors. The computational cost of the argmax operation to pick the best word is negligible, hence the computational cost is essentially the same for matching words in the user prefix and predicting new words.

To predict a single word, the CPU requires over 100 milliseconds, which is clearly too slow. The time it takes to translate a single word slightly increases with the length of the sentence, since the attention mechanism has to sum over a larger context of source language words.

On a Tesla K80 GPU, the cost to predict one word drops to 7 milliseconds. For a 20-word sentence, this means 140 milliseconds (7 × 20) which is also beyond our 100 millisecond time limit.

5.7.3 Optimizations

In CASMACAT, the server expects to receive full sentence translation output from the machine translation server each time that a new translation or prefix completion is requested and subsequently returned to it (Alabau et al., 2014). As we noted, CPU implementations of neural interactive translation prediction are too slow to be used in a real-life setting, and very long sentences may also be difficult to translate fast enough even on a GPU. In order to deliver translation predictions at an adequate speed, we perform translation using a GPU and implement several time-saving approaches.

We employ the following optimizations:

Precompute: We precompute the initial translation for each sentence. We allow a long time limit for this (5 seconds) as it is done in the background when the page opens, before translators begin translating. We also limit the output to 100 tokens.

Timeout: At any other point, when we are computing the predicted translation suffix for a translator-produced prefix, we only continue generating token predictions while fewer than 80ms have elapsed. This does mean that sometimes we will be left with only a partial sentence completion, which we attempt to turn into a full sentence using patching (described below).

Cache: We also perform caching to improve speed. As we produce a hypothesis translation, we also save the hidden states and probability distributions that were used to produce that hypothesis. That way, if the translator accepts part of the hypothesis but then diverges from it, we do not need to recompute those values, and can simply consume the new divergent continuation of the translation before predicting new tokens. Similarly, if the translator returns to edit an earlier part of the sentence which has been cached, this can save computation time as well. However, should they then return to editing the end of the sentence, having introduced new tokens in the middle, the system will still need to force-decode again until it reaches the end. One could also implement caching by translator and/or by document.

Patch: When the prediction of the remaining tokens in the sentence stops early due to timeout, we patch together the current tokens and the end of a previous longer (complete) translation. (If none exists, we simply return the partial translation without patching.) Assuming a longer previous translation exists, we select where to patch using KL-divergence between probability distributions. We describe patching in more detail below.

(1) Initial hypothesis	*A sovereign prel@@ ate of the Champions League season .*
(2) Translator prefix	A confident
(3) New prediction	`start to the`
(4) Alignment	`start to the` → *prel@@ ate of* ***the*** *Champions*
(5) New hypothesis	A confident `start to the` *Champions League season .*

Figure 5.5: Example of patching a 3-word prediction into the original sentence completion.

The patching method (without beam search) that we propose patches together a limited (say, 3 word) new prediction with the existing sentence completion each time that the translator diverges from the predicted translation. An example of this is shown in Figure 5.5 (we reference the row numbers parenthetically in the following description). We begin by precomputing a full translation when the document is uploaded (row 1). If and when the translator diverges from this (row 2), we compute predictions for the next 3 words only (row 3) and attempt to patch them together with the original translation.

We find the patch position by computing the KL divergence between the probability distribution that produced the last of the 3 new words and the stored probability distributions that produced the words in a 5-word window (following the position of the the last word in the translator prefix). This results in an alignment between the last of the 3 new words and the index of some word in the existing translation (row 4). The new translation hypothesis consists of concatenating the translator prefix, the 3 newly predicted words, and any words following the position of the index in the existing translation hypothesis that minimized the KL divergence (row 5).

By patching together earlier predictions with a short sequence of predictions based on new input from the translator, we can guarantee that we can serve the translator new predictions quickly. The new prediction and patching combined takes an average of 54.3 milliseconds to compute. This approach yields a word prediction accuracy of 56.4% and a letter prediction accuracy of 84.2% for German–English on the WMT 2016 news test set (a drop from the full search neural model by 5.2% and 2.4%, respectively, but still vastly outperforming the phrase-based search graph system).

In a real-life setting, we may sometimes have enough time to recompute the full sentence in the background, rather than relying on patching together different predictions, so we could expect performance closer to the performance noted earlier. Additionally, we could use beam search (or other improvements to the neural model)

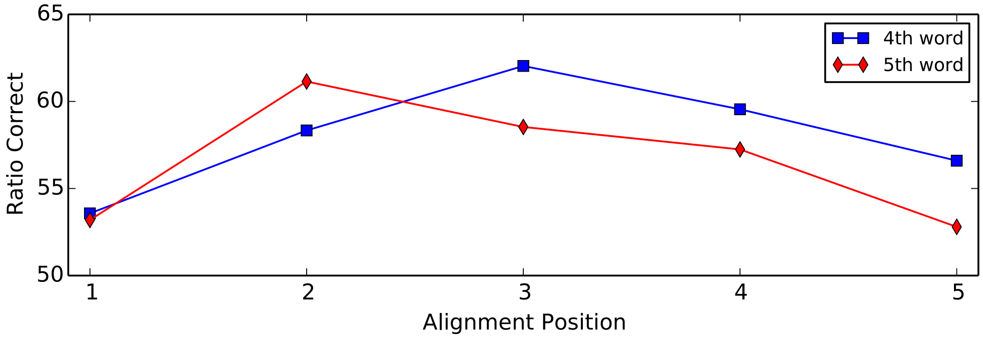


Figure 5.6: Ratio correct after first failure for the 4th and 5th words in the window. The horizontal axis represents the alignment position found by the patching heuristic.

in order to precompute better initial sequences, which we expect would also improve performance.

We analyze the performance of this patching heuristic. In the example in Figure 5.5, the new hypothesis is in fact the correct translation. If the initial error by the system is a single-token error (for example a synonym), we might expect the last of the 3 newly translated tokens to align to the token at the center of the window. In this case it (correctly) aligns one position to the right of this and produces the desired hypothesis.

An alignment position of 3 indicates that the 3rd newly translated token aligned with the 3rd token in the window (as would be expected if no reordering were needed). Similarly, an alignment position of 1 indicates that the 3rd newly translated token aligned to the 1st token following the failure, and so on. In the example in Figure 5.5, we are attempting to patch together some portion of the initial hypothesis (*A sovereign prel@@ ate of the Champions League season .*) with the translator prefix (<u>A confident</u>) and the new partial prediction (`start to the`). As there are two tokens in our translator prefix, we look for a token following the first two tokens of the initial hypothesis as a place to patch. Our window of five tokens is *prel@@ ate of the Champions*, with *prel@@* being in alignment position 1, *ate* in 2, *of* in 3, *the* in 4, and *Champions in 5.* After computing the KL divergence between the probability distribution that produced the last token (`the`) of the new partial prediction with each of the stored probability distributions that generated the tokens in the window of five tokens, we find that *the* in alignment position 4 had the lowest KL divergence of all five. This is then the position that we use for patching together our new hypothesis:

A confident `start to the` *Champions League season .*

When the alignment is close to the center of the window, this suggests that the sentence does not require much reordering. The patching heuristic is somewhat imprecise and has difficulty handling sentences with long-range reordering. In Figure 5.6 we compute the failure recovery ratios for the 4th and 5th words in the window following the first error,[8] conditioned on the alignment position. We see that when a longer-distance alignment occurs (aligning to position 1 or 5, rather than 3), the ratio drops, demonstrating either an error of alignment or the system's difficulty in handling long-distance reordering.

[8] We show only performance for the 4th and 5th words in the window; performance on the first three is identical to the no-beam-search values reported in Table 5.5, as the patching occurs after this sequence of 3 new predictions.

5.8 Conclusion

In this chapter, we demonstrate that neural machine translation systems can be effectively applied to interactive translation prediction, improving upon the performance of phrase-based statistical methods. We show that they recover well from errors, have shorter sequences of incorrect predictions, and, when they do make errors, more frequently predict synonyms than phrase-based systems do. We demonstrate that a combination of speed-related heuristics and use of GPU hardware can make them fast enough for practical application, though there remains room for improvement. In particular, most of our approaches assume that the translators generate their text from start to finish, without returning to edit earlier sections; while the caching heuristic may help with this, it is not guaranteed to solve all related challenges.

Chapter 6

User Study

6.1 Introduction

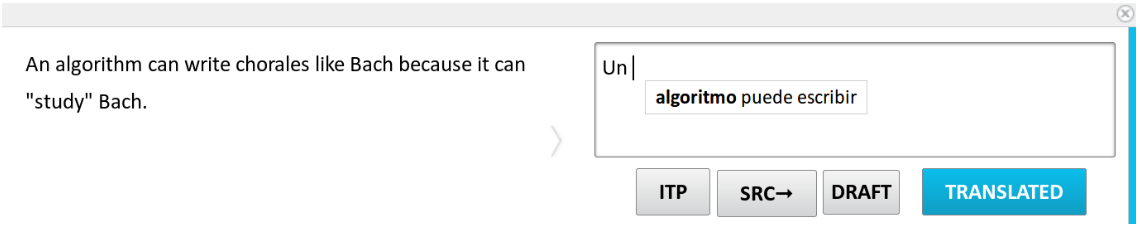
> *This chapter draws primarily on work published in: Knowles, Sanchez-Torron, and Koehn (2019). This paper was a collaboration, with equal contributions from the first two authors: I built the neural machine translation system used, integrated the neural interactive translation prediction system into the CAT interface, extracted user logs, and performed word prediction accuracy analysis. My coauthor, Marina Sanchez Torron, recruited the translators, selected the texts to be used in the study, developed the experimental protocol and instructions, and processed the logs and user survey responses.*

We investigate the use of neural interactive translation prediction through a user study with professional English–Spanish translators. We integrated our Nematus-based neural interactive translation prediction implementation into the open-source CASMACAT translation workbench (Alabau et al., 2014) and conducted a user study. The goals of the study were to examine productivity with interactive translation prediction, to determine whether translation productivity increased as translators became more familiar with the technology, and to collect translator impressions about interactive translation prediction.

The eight translators who participated in the study (who we refer to as TrA through TrH) were Castilian Spanish professional translators, experienced in translating from English into Spanish. All but two of them (TrA and TrB) had degrees in translation, and all had some level of higher education. With the exception of TrB, all translators had some prior post-editing (PE) experience. Details regarding their post-editing experience are shown in Table 6.2; additional detail and analysis can be found in Sanchez Torron (2017).

The study consisted of eight sessions spanning four weeks. During each session, translators were asked to translate one of eight news texts. In the first session, translators were asked to post-edit one news document each (N=231 sentences total). In the remaining seven sessions, they performed translation using the interactive translation prediction tool, translating one document per session (N=1377 sentences total). The post-editing session provided us with a baseline for each translator's productivity.

The news texts used in the study were selected controlling for length and syntactic complexity. Texts had on average 29.13 sentences (*SD*=1.24), 822.75 tokens (*SD*= 37.48), and a dependency length of 103 (*SD*= 2.99) and were assigned randomly to translators, while ensuring that each text was presented only once in each session



An algorithm can write chorales like Bach because it can "study" Bach.

Un

algoritmo puede escribir

Figure 6.1: Interactive translation prediction in CASMACAT: The system suggests to continue the translation with the words *algoritmo puede escribir*, which the user can accept by pressing the TAB key.

and only once to each translator throughout the study. Translators were asked to produce publication quality translations that used as much of the machine translation output as possible, and they were asked not to perform preferential changes that would not improve text quality. We used the English–Spanish neural machine translation system described in Section 4.2.3 to produce translations for post-editing as well as for interactive translation prediction.

The original sample size was reduced by about 17% due to technical issues (server down) invalidating two interactive translation prediction translator sessions and due to one translator (TrB) choosing not to follow instructions. We still report data on TrB's background and reactions to the tool, as those may shed light on interesting avenues to pursue when examining who will benefit the most from such tools.

6.2 Translator Interactions

Translators received instructions about the study (including compensation, the modes of interaction that they would use, the expectations of translation quality, etc.) and also had access to a help page through the CASMACAT interface. Before beginning the study proper, translators conducted short warm-up exercises consisting of 5 sentences to familiarize themselves with CASMACAT as well as both the interactive translation prediction and post-editing modes.

The CASMACAT system logs all keystrokes, mouse clicks, and movements between segments in the interface, along with timestamps. The system also logs requests to the translation server, source data, initial translation data from the machine translation system, and final translation output produced by the translators. While the underlying translation system vocabulary consists of subword segments, user interactions are performed at the character level (by typing individual characters) and at the whole-word level (by hitting TAB to accept a suggestion). All byte pair operations are performed behind the scenes and are not shown to the user.

In the user interface (UI), shown in interactive translation prediction mode in Figure 6.1, translators see a source sentence on the left and a space to enter their translation on the right. They translate the document sentence-by-sentence. During post-editing, the right side is initially populated with MT output, which the translator then edits, as in a standard word processor. During interactive translation prediction, a floating box to the right and below the translator-produced prefix shows the next three suggested words. The translator can accept a word using the TAB key, or type a new word one character at a time.

6.3 User Study Results

We evaluate neural interactive translation prediction both in terms of translator productivity and translator satisfaction. Productivity was measured in terms of temporal effort, technical effort, and final translation quality, across eleven variables derived from the CASMACAT logs and the final translation output. This evaluation is described in Section 6.3.1. We evaluated translator impressions using a post-study survey, as described in Section 6.3.2.

6.3.1 Sample Results

We considered three categories of translator productivity, measured by eleven unique variables. Temporal effort was measured in terms of processing time (seconds per source token). Technical effort was measured through the counts of Manual Insertions, Manual Deletions, Navigation and Special Key Presses (UP, DOWN, LEFT, RIGHT, CTRL, ALT, SHIFT, and TAB), Mouse Clicks, and Tokens of MT Origin (the count of tokens in the final translator output that were accepted by the translator exactly and then kept unchanged). The final translation quality was measured by manual annotations of MQM Score,[1] Accuracy Issues, Fluency Issues, Minor Issues, and Major Issues. We also separately examined word prediction accuracy for each translator.

Table 6.1 shows summary statistics (mean and standard deviation) for translation productivity indicators across all translators, broken down by translation condition. As Table 6.1 indicates, these results favor interactive translation prediction for eight out of the eleven productivity indicators. No critical issues were observed in any submitted translations (as expected given that the participants were professional translators).

We expected that we might observe consistent trends over time in interactive translation prediction, and examined this through exploratory graphs.[2] None of these exploratory graphs showed consistent trends that would indicate an increasing comfort or productivity with the tool, with the exception of Mouse Clicks. Mouse Clicks showed a steady decrease from the first interactive translation prediction session (M = 0.34, SD = 0.40) gradually to the last interactive translation prediction session (M = 0.28, SD = 0.46). This change may indicate that translators modify how they interact with the tool (decreasing mouse use) over time.

As Table 6.2 shows, the effect of interactive translation prediction on individual

[1] Multidimensional Quality Metrics (MQM) is a framework for translation quality assessment, which provides a set of issue (problem) types, guidance for categorizing those, and outlines severity levels (minor issues like extra spaces, major issues like spelling errors that do not make text uninterpretable but that could require extra reader effort, and critical issues that change text meaning). The MQM annotations for our output were produced by Marina Sanchez Torron, who analyzed the translations. Details of MQM are here: `http://www.qt21.eu/mqm-definition/definition-2015-12-30.html`

[2] These exploratory plots and additional analysis can be found in Sanchez Torron (2017).

	ITP		PE	
	Mean	SD	Mean	SD
↓ Processing Time (seconds per source token)	4.56	3.88	4.79	6.31
↓ Manual Insertions (count per source token)	2.55	2.31	3.52	3.85
↓ Manual Deletions (count per source token)	1.18	1.54	3.37	3.78
↓ Navigation and Special Key Presses (count per source token)	1.13	0.66	0.29	0.48
↓ Mouse Clicks (count per source token)	0.31	0.41	0.54	0.45
↑ Tokens of MT Origin (count per 100 source tokens)	61.93	30.22	59.36	33.33
↑ MQM Score (percentage)	98.52	4.01	98.25	6.02
↓ Fluency Issues (count per 1000 source tokens)	6.18	17.91	2.14	8.62
↓ Adequacy Issues (count per 1000 source tokens)	4.71	15.87	8.12	25.58
↓ Minor Issues (count per 1000 source tokens)	9.9	23.11	8.19	25.52
↓ Major Issues (count per 1000 source tokens)	0.97	6.3	2.08	12.22

Table 6.1: Summary statistics for translation productivity indicators in ITP and PE. Arrows indicate whether a decrease or increase indicates improvement (i.e., a lower processing time is better, so it is marked with ↓, while a higher MQM score is better, so it is marked with ↑).

		TrA	TrB	TrC	TrD	TrE	TrF	TrG	TrH
PE cert.		N	N	Y	N	N	Y	N	N
PE exp.(yrs)		2-5	0	2-5	2-5	5-10	5-10	<2	2-5
I prefer ITP		+	- -	+	++	+	+	-	=
I'd use ITP		+	- -	+	+	++	++	-	-
Processing Time	ITP	3.19	3.77	2.55	5.43	5.84	3.2	5.89	5.9
	PE	2.42	3.61	2.57	3.56	7.04	3.63	9.4	4.98
Manual Insertions	ITP	3.56	6.11	1.15	4.15	1.98	0.76	1.7	4.96
	PE	3.9	5.17	3.21	1.67	1.49	1.92	8.73	4.01
Manual Deletions	ITP	1.2	0.65	1.95	1.15	1.12	0.47	0.75	1.62
	PE	3.78	5.13	3.18	1.49	1.45	1.89	8.43	3.68
Nav. and Special Key	ITP	1.21	0.29	1.88	0.82	0.98	1.31	1.08	0.6
	PE	0.49	0.08	0.68	0.08	0.03	0.72	0.03	0.06
Mouse Clicks	ITP	0.14	0.19	0.37	0.67	0.14	0.11	0.32	0.49
	PE	0.32	0.32	0.3	0.45	0.6	0.34	0.91	0.85
Tokens of MT origin	ITP	55.63	10.9	81.73	35.85	68.91	86.48	61.8	37.75
	PE	53.62	26.82	59.33	74.9	79.17	75.23	21.51	49.68
WPA	ITP	65.92	52.02	78.82	59.31	76.58	83.92	68.36	61.01
	PE	68.04	55.32	69.38	79.27	76.56	76.50	37.32	68.51
MQM Score	ITP	99.51	96.48	98.22	99.23	98.01	97.95	98.52	98.42
	PE	99.4	99.42	98.65	99.25	98.51	97.13	96.05	98.6
Fluency	ITP	2.51	10.28	3.08	5.47	9.57	13	6.73	2.23
	PE	3.54	0	0.49	3.22	1.91	1.43	4.66	0
Adequacy	ITP	1.90	10.27	8.06	2.14	5.47	4.02	3.47	7.18
	PE	3.08	5.76	4.29	4.47	9.49	22.96	9.66	3.55
Minor	ITP	4.04	19.06	9.41	7.62	14.10	15.94	9.19	7.78
	PE	6.61	5.76	1.81	7.69	10.39	22.96	6.84	1.40
Major	ITP	0.37	1.48	1.72	0	0.98	0.99	1.01	1.55
	PE	0	0	2.96	0	1.01	1.43	7.48	2.15

Table 6.2: Translators' main translation productivity indicators and impressions. Processing Time is measured in seconds per source token; Manual Insertions, Manual Deletions, Navigation and Special Key Presses, and Mouse Clicks are measured as counts per source token. Tokens of MT origin are measured as counts per 100 source tokens; MQM Score as a percentage, and translation issues as count per 1000 source tokens. Word Prediction Accuracy (WPA) is a percentage. Likert responses are ranked from most negative to most positive: - -; -; =; +; ++.

translators' productivity indicators varies. All translators made more Navigation and Special Key presses and fewer Manual Deletions in interactive translation prediction, and all but two (TrC and TrD) made fewer Mouse Clicks in interactive translation prediction. The increase in Special Key presses is directly attributable to the use of the TAB key to accept translation suggestions in the interactive translation prediction interface. The decrease in Mouse Clicks seems intuitive given the interaction modes; a mouse click should only be required in interactive translation prediction if a translator chooses to return to an early part of the sentence to make changes, whereas it may be a more natural way to navigate and interact during post-editing (just as it is during editing of a standard text document in most word processors). Additionally, all but one translator (TrA) produced texts with more Fluency Issues in interactive translation prediction, and all but one translator (TrC) produced texts with fewer Adequacy Issues in interactive translation prediction.[3]

We observe a wide range of word prediction accuracy scores (obtained by rerunning neural interactive translation prediction as a simulation on the final translator output) for both interactive translation and post-editing, showing (as also shown in the Tokens of MT Origin) that the usefulness of the suggestions varies by translator. In all cases

[3] In addition to the lack of spell checking in our interface, a bug in tokenization for ITP may have introduced some spelling errors when the translator's spacing (for example, leaving whitespace between a number and the character "%") did not match the automatic detokenization performed by the system on the backend. This resulted in system suggestions of words with a character missing. These errors were quite rare and only reported by one translator. Spelling errors (including Spanish vs. Catalan spelling differences) were also introduced naturally by translators. It is possible that translators did not catch these errors before continuing to the next sentence, perhaps due to the lack of spell checker or if they were less thorough than they would typically be in checking their translations.

(except TrB), the word prediction accuracy for a translator using interactive translation prediction is higher than the reference-simulated overall word prediction accuracy (59.1%), indicating that this automatic metric underestimates the tool's usefulness to translators. While there is not a strict correlation between the positivity of translator reactions to interactive translation prediction and word prediction accuracy or Tokens of MT Origin, the three translators with the highest word prediction accuracy do agree strongly or agree that they would use interactive translation prediction in real-life scenarios, while the translator with the lowest word prediction accuracy strongly disagreed (this is TrB, who also chose not to follow task instructions due to dissatisfaction with the translation mode). The two translators with the most post-editing experience agreed that they would use interactive translation prediction in their work and both have high word prediction accuracy scores; this may suggest that they are adept at using machine translation output in their translations.

Four translators were faster in interactive translation prediction, the same number (though not the exact same set) that applied fewer Manual Insertions and made more use of MT in interactive translation prediction, as measured by Tokens of MT Origin.[4] This is similar to earlier studies that have found notable between-translator variation. We discuss potential reasons for variation in Section 6.3.2.

[4] In particular, TrG's "outlying" indicators in post-editing are partly due to replacing English quotation marks in the translation by guillemets: TrG copied the guillemets from an outside source and pasted them into CASMACAT's interface, but in the process also pasted whitespaces and line breaks (adding to the count of Manual Insertions), some of which TrG then manually deleted, hence the higher temporal and technical effort indicators.

6.3.2 Translators' Impressions

We used a 5-level Likert scale questionnaire to collect translators' impressions of the following statements:

- *I prefer ITP to PE*
- *ITP is less tiring than PE*
- *As the study progressed, I took better advantage of the ITP suggestions*
- *ITP helps me translate faster than PE*
- *ITP helps me translate to better quality than PE*
- *I would use ITP in real-life scenarios.*

The survey also provided them the opportunity to answer open questions: *Do you have any suggestions for improvement of any aspect of interactive translation's use?* and *Please provide any additional comments about your experience with interactive translation prediction.*

Translators' impressions of ITP were very positive overall. Of the eight translators, five agreed (TrA, TrC, TrE, TrF) or strongly agreed (TrD) that they preferred the interactive translation prediction mode over post-editing. Five agreed (TrA, TrC, TrD) or strongly agreed (TrE, TrF) that they would use interactive translation prediction in real-life translation scenarios. Six translators agreed (TrD, TrE, TrF, TrG,TrH) or strongly agreed (TrA) that they felt that they took better advantage of the system's suggestions as the study progressed. Three translators agreed (TrG,TrH) or strongly agreed (TrD) that interactive translation prediction was less tiring than post-editing, with one strongly disagreeing (TrB), and the rest giving neutral answers.

Our sample size of translators is quite small, so we should be cautious about drawing strong conclusions or overly broad generalizations. Nevertheless, we do see some patterns emerge which are worth noting as a basis for future consideration and study. In their study on post-editing, Moorkens and O'Brien (2015) find that experienced translators have more negative views of post-editing than novices (translation students). In our study, we do not observe a comparable pattern of more experienced translators expressing more negative views of interactive translation prediction. Instead, we observe that the participants with more post-editing experience in their backgrounds generally tended to have more favorable views of interactive translation prediction. Both of the translators with 5-10 years of experience (TrE, TrF) expressed positive views of interactive translation prediction, preferring it to post-editing. In fact, the most experienced translator (TrA), both in terms of length of experience ($>$ 10 yrs) and translation volume in the previous 12 months ($>$ 55k words) – who had 2-5 years of post-editing experience – expressed, as detailed above, consistently positive views of interactive translation prediction. In terms of translation productivity indicators, as shown in Table 6.2, TrA logged the fastest post-editing time and the second fastest interactive translation prediction time of all translators. TrA also produced the highest quality texts in the post-editing condition and the highest quality texts of all translators in the interactive translation prediction condition. Similarly, three of the four translators with 2-5 years of experience (TrA, TrC, TrD) preferred it to post-editing and would consider using it in their work. The remaining translator with 2-5 years of experience (TrH) was neutral in terms of preference and weakly negative towards the use of interactive translation prediction in their work. The two translators with the most negative impressions of interactive translation prediction (TrB, TrG) were the two translators with the least post-editing experience (0 and $<$ 2 years, respectively). One of these translators (TrB) strongly rejected interactive translation prediction, as evidenced by strongly disagreeing to all Likert scale questions and expressing negative views in the open questions. TrB chose to ignore the interactive translation prediction assistance altogether after just one session, not accepting a single token the interactive translation prediction system suggested afterwards (instead typing all translations character-by-character, even those that matched the suggestions). This reaction may be related to TrB experiencing prediction delays (see Section 6.3.3). The translation activity data produced by TrB was deemed invalid and discarded, as any measures collected would not be representative of working in interactive translation prediction (and because the translator had ignored the task instructions to use as much machine translation output as possible), but rather of unassisted translation.[5] Finally, there is some indication that translators who have formal post-editing training or provide post-editing services frequently benefited the most from interactive translation

[5] We did compute word prediction accuracy on TrB's translations in simulation; with a score of 52.0%, TrB was the only translator who had a lower word prediction accuracy than the simulated WMT test set word prediction accuracy.

prediction. In fact, of the four translators who were faster in interactive translation prediction than in post-editing, two have post-editing industry certifications (TrC [TAUS]; TrF [SDL]) and one (TrE) provides frequent post-editing services.

This all raises a number of questions for future study.

- Are the translators with more post-editing experience more open to using machine-produced output?
- Do they see it as an improvement compared to post-editing *and* translating from scratch, or simply the former?
- The translators in this study chose to participate in a study of new translation technologies; are they more comfortable with or positively predisposed towards machine translation technologies, as compared to the general population of translators?

It may be that some translators are not willing to engage with post-editing or interactive translation prediction, possibly because they already have a working routine they are comfortable with. In this sense, the views expressed by Vasconcellos and León (1985), O'Brien (2002), Rico Pérez and Torrejón (2012) and De Sutter (2011) that post-editing requires that the translator has a positive or open attitude towards machine translation also seem to resonate for interactive translation prediction. All of these – along with the question of whether there really is meaningful correlation between post-editing experience and positive reactions to interactive translation prediction – are questions that could be examined in a larger scale study.

In addition to sharing positive and negative impressions, translators commented on their experiences with the interface and the machine translation system. Three translators (TrA,TrD,TrE) identified desirable UI features such as keyboard shortcut customization and search and replace options. Four translators (TrB, TrC, TrD, TrH) noted machine translation issues in terms of orthography, grammar, translation, style, and discourse and three translators (TrC, TrE, TrF) pointed out variation in machine translation quality level from sentence to sentence. One translator reacted to this by treating confusing suggestions as a post-editing task (accepting all suggestions and then post-editing a full sentence). Two translators (TrB, TrF) felt that not being able to see the whole machine-translated text (only being shown the next three suggested words in interactive translation prediction) slowed their overall translation workflow, because otherwise they could quickly perform triage to determine whether or not the MT output was going to be helpful.[6] This highlights two potential ways of improving interactive translation prediction: providing confidence or quality estimation information (perhaps even to allow translators to determine which way they would prefer to interact with the output) and improving underlying machine translation system quality. Knowles and Koehn (2018b) provides initial steps towards confidence estimation for neural interactive translation prediction.

[6] Showing the full sentence is a display option, which we did not examine in this study.

Translators also commented on the differences between interactive translation prediction and post-editing. Two translators (TrA, TrG) noted the cognitive and translation process differences between the two, such as interactive translation prediction resulting in “less time researching terminology” (TrG) and it involving “a mental process different to PE, consisting of constantly comparing ITP’s suggestions to the translator’s own mental translations, a process that, while seemingly complex, nevertheless sped up translation times” (TrA). These two translators also expressed their worries about the translator’s role: machine translation priming may mean that “the voice of the translator is lost” (TrG), and the user-friendliness and speed of the interactive translation prediction system may generate overconfidence and “lead to mistakes or wrong decisions if the required exigence and rigor levels are not there, on the user’s side” (TrA). It would be helpful to perform larger-scale comparisons between unassisted translation, post-editing, and interactive translation prediction in order to determine the level of influence of the machine translation system, and indeed whether translators do place too much trust in the systems. We observed similar overall quality between post-editing and interactive translation prediction, and observed mixed results (varying by translator) as to the tokens of machine translation origin that appeared in their output. While we did not observe clear trends over time, three translators did feel that some time had to elapse before making the most out of interactive translation prediction:

> *“As the experience went on [ITP] helped me finish the tasks in a shorter time*

and with a higher level of confidence in the quality of my work." (TrA);

"By the end of the study I found [ITP] to be a user friendly and straightforward tool" (TrF);

"I had the distinct feeling that, on average, the suggestions were more and more spot on as I proceeded"[7] (TrD).

6.3.3 Speed

As noted in Section 5.7.2 and shown in Table 5.6, it is necessary to use a GPU and other time-saving heuristics in order to perform computation quickly enough to make neural interactive translation prediction usable. While we used a Tesla K80 GPU for benchmarking in our simulation work, we used a machine with an NVIDIA GeForce GTX 1080 GPU for our user study. On this GPU, our speed increases to an average of one token every 3.7ms, an improvement over the 7 or more milliseconds we observed with the Tesla K80.

We aimed to return each suggestion to the translator in under 100ms, in order to avoid the user sensing a lag. In our study, 73.5% of the suggestion requests during valid ITP sessions were returned to the user in under 100ms (99.1% in under 300ms). Nevertheless, two translators (TrB, TrH) explicitly reported experiencing delays or concerns about the speed of translation suggestions. This is likely due to: (1) having

[7] While our setup did not include adaptation to translator corrections, future work could create additional gains by doing so, as in Kothur, Knowles, and Koehn (2018) and Peris and Casacuberta (2019). We examine this in simulation in Chapter 9.

experienced one of the few instances of slower response times, (2) accepting all tokens and reaching the end of the current prediction (after which point new suggestions will not be generated until the translator makes a change to the prefix), or (3) network lag (the server was located in the United States, and the translators, based in Europe, accessed the tool through a web interface). To mitigate the first, future work could use a faster NMT decoder adapted for ITP, or set lower thresholds. For the second, we could change the interaction between the user interface and the MT backend such that accepting a token triggers additional translation (if the suggestion produced so far has not yet reached an end-of-sentence token).

6.4 Conclusion

In this chapter, we presented the results of a user study of interactive translation prediction. Eight translators participated in the study, translating news story documents from English to (Castilian) Spanish using post-editing and interactive translation prediction. Most of the translators had positive reactions to interactive translation prediction, and about half of them translated more quickly using interactive translation prediction. Collecting both logging data regarding tool use as well as user reactions, we show that this approach is a viable tool for some translators, and raise additional questions for future research into computer aided translation tools.

Part IV

Fine-Grained Adaptation

This section of the dissertation examines fine-grained adaptation for improving neural machine translation performance on individual documents. We examine approaches that incrementally adapt to single sentences (as a translator would produce while working through the translation of a document) and to document-specific dictionaries of novel words. As background and motivation for this, in Chapter 7 we first provide analysis of neural machine translation performance on rare and novel words, and then examine how neural machine translation systems copy observed and novel words. After demonstrating the challenges that these types of words pose to neural machine translation systems (as well as where they may find success), in Chapter 8 we show that we can quickly and effectively adapt neural machine translation systems to perform better overall as well as specifically on these challenging words.

Chapter 7

Analysis of Word-Level Performance

7.1 Introduction

We begin by demonstrating and analyzing the challenge that different types of rare words pose. In particular, we first consider words that were unobserved or rarely observed during training (Section 7.2). One particular case stands out among these: words that can or should be copied from the source to the target. We examine copying behavior in neural machine translation systems in Section 7.3. The analysis of rare and copied words provides a justification for research into improving translation of rare and novel words, including through fine-grained adaptation.

One important issue to note is that of the distinction between words and subwords.

Human users of machine translation will generally experience translation output – either when reading translated text, post-editing, or performing some form of interactive translation – at the whole word level, while in the background the systems are operating at the subword level. Since our interest is in the human use case, in this chapter we focus on measuring the translation accuracy of whole words, *not* the subword units on which the models are trained. The consequence of this is that when we refer to the frequency of a word in the training corpus, we may actually be referring to the frequency of a *full sequence of subword tokens*, of which some subsequences may be more frequent than the sequence as a whole. In some instances, the greater frequency of individual subsequences may be what enables successful translation (e.g., in the case of inflected words), while in other cases translation can be successful without that (e.g., in the case of certain proper nouns). We examine these phenomena in more depth in Sections 7.2.2 and 7.3.

7.2 Rare Words

This section draws on work from Koehn and Knowles (2017).

Conventional wisdom states that neural machine translation (NMT) models perform particularly poorly on rare words, (Luong et al., 2015; Sennrich, Haddow, and Birch, 2016c; Arthur, Neubig, and Nakamura, 2016) due in part to the smaller vocabularies used by NMT systems.[1] We examine this claim by comparing performance on rare

[1] The fact that NMT models require large amounts of training data to perform comparably to

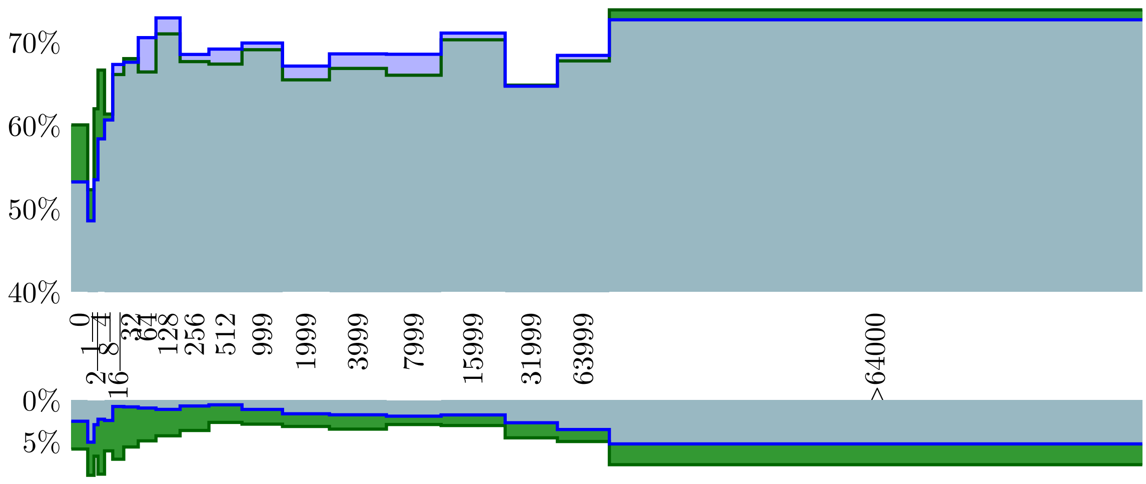


Figure 7.1: Precision of translation and deletion rates by frequency of the source word's type. SMT (light blue) and NMT (dark green). The horizontal axis represents the corpus frequency of the source types, with the axis labels showing the upper end of the bin. Bin width is proportional to the number of word types in that frequency range. The upper part of the graph shows the precision averaged across all word types in the bin. The lower part shows the proportion of source tokens in the bin that were deleted.

word translation between NMT and SMT (phrase-based statistical machine translation) systems of similar quality for German–English[2] and find that NMT systems actually outperform SMT systems on translation of very infrequent words. However, both NMT and SMT systems do continue to have difficulty translating some infrequent words, particularly those belonging to highly-inflected categories.

Both models have case-sensitive BLEU scores of 34.5 on the WMT 2016 news test set (for the NMT model, this reflects the BLEU score resulting from translation with a beam size of 1). We use a single corpus for computing our lexical frequency counts

SMT systems (as we do note in the Amount of Training Data section of Koehn and Knowles (2017)) may also give weight to this idea.

[2] These are the same models described in Sections 4.2.1 and 4.2.2 and used in Chapter 5 to compare phrase-based and neural interactive translation prediction.

(a concatenation of Common Crawl, Europarl, and News Commentary).

7.2.1 Examining the Effect of Source Word Frequency

We follow the method described by Koehn and Haddow (2012) for examining the effect of source word frequency on translation accuracy.

First, we automatically align the source sentence and the machine translation output. We use fast-align (Dyer, Chahuneau, and Smith, 2013) to align the full training corpus (source and reference) along with the test source and MT output. We use the suggested standard options for alignment and then symmetrize the resulting alignment with `grow-diag-final-and`.

Each source word is either unaligned ("dropped") or aligned to one or more target language words. For each target word to which the source word is aligned, we check if that target word appears in the reference translation. If the target word appears the same number of times in the MT output as in the reference, we award that alignment a score of one. If the target word appears more times in the MT output than in the reference, we award fractional credit. If the target word does not appear in the reference, we award zero credit. We then average these scores over the full set of target words aligned to the given source word to compute the accuracy for that source word. Source words can then be binned by their frequency in the training corpus and average translation accuracies can be computed.

7.2.2 Results

The overall average accuracy is quite similar between the NMT and SMT systems, with the SMT system scoring 70.1% overall and the NMT system scoring 70.3%. This reflects the similar overall quality of the MT systems. Figure 7.1 gives a detailed breakdown. The values above the horizontal axis represent accuracies, while the lower portion represents what proportion of the words were deleted. The first item of note is that the NMT system has an overall higher proportion of deleted words. Of the $64,379$ words examined, the NMT system is estimated to have deleted 3769 of them, while the SMT system deleted 2274. Both the NMT and SMT systems delete very frequent and very infrequent words at higher proportions than words that fall into the middle range. Across frequencies, the NMT system deletes a higher proportion of words than the SMT system does. This finding is consistent with the results regarding sentence length discussed in the Long Sentences section (3.4) of Koehn and Knowles (2017); those experiments showed that SMT outperformed NMT on sentences of length 60 or greater, with the NMT system producing output that was too short.[3]

The next observation of interest is what happens with unknown words (words which were never observed in the training corpus). The SMT system translates these

[3] It is worth noting several factors that can contribute to this. Preprocessing for NMT training typically removes very long sentences from the training data. Also, unlike the SMT systems, the NMT systems examined did not incorporate any coverage mechanism.

Label	**Unobserved**	**Observed Once**
Adjective	4	10
Named Entity	40	42
Noun	35	35
Number	12	4
Verb	3	6
Other	6	3

Table 7.1: Breakdown of the first 100 tokens that were unobserved in training or observed once in training, by hand-annotated category.

correctly 53.2% of the time, while the NMT system translates them correctly 60.1% of the time. This is reflected in Figure 7.1, where the large gap in performance on unknown words is visible.

Both SMT and NMT systems actually have their worst performance on words that were observed a single time in the training corpus, dropping to 48.6% and 52.2%, respectively; this is even worse than for unobserved words. Table 7.1 shows a breakdown of the categories of words that were unobserved in the training corpus or observed only once. The most common categories across both are named entity (including entity and location names) and nouns. The named entities can often be passed through unchanged (for example, the surname “Elabdellaoui” is broken into “E@@ lab@@ d@@ ell@@ a@@ oui” by the byte pair encoding for NMT and is correctly passed through unchanged by both the NMT and SMT systems). We delve into this phenomenon of copying more deeply in Section 7.3. Many of the nouns are compound nouns; when these are correctly translated, it may be attributed to compound-splitting (SMT) or byte pair encoding (NMT), respectively. For example, consider the word “Sozialstiftung” (“social foundation”, successfully segmented by byte pair encoding for NMT along the morphological boundary "Sozial@@ stiftung"); despite never having been observed in training, it is successfully translated by both the NMT and SMT systems. Byte pair encoding is not guaranteed to always provide morphology-respecting segmentations, but the morphological soundness of a segmentation is not a prerequisite for successful translation; the NMT system successfully translates the word "Ligaspielen" ("league games") despite the fact that it was segmented as "Lig@@ asp@@ ielen" (a more morphologically meaningful segmentation would be splitting the compound noun into "Liga | spielen" or even to break off the inflected ending as well).[4] In contrast to overall performance, we find that for the NMT systems, overall performance on nouns (NN tag[5]) is quite high. The worst performance is on NN words that were observed just once (55.7%), but this represents a smaller drop than is observed across all words; the performance on unknown words is only slightly higher (59.7%). The SMT system has lower performance for rare NN words, as shown in Figure 7.2.

The categories which involve more extensive inflection (adjectives and verbs) are arguably the most interesting. Adjectives and verbs have worse accuracy rates and higher deletion rates than nouns across most word frequencies. We show this in figures 7.3 and 7.4.

The factored SMT system also has access to the stemmed form of words, which can play a similar role to byte pair encoding in enabling translation of unobserved

[4] The SMT system also translates this correctly, perhaps on the basis of compound-splitting.

[5] All tags used here were generated as part of the Moses preprocessing pipeline for the source side of the data.

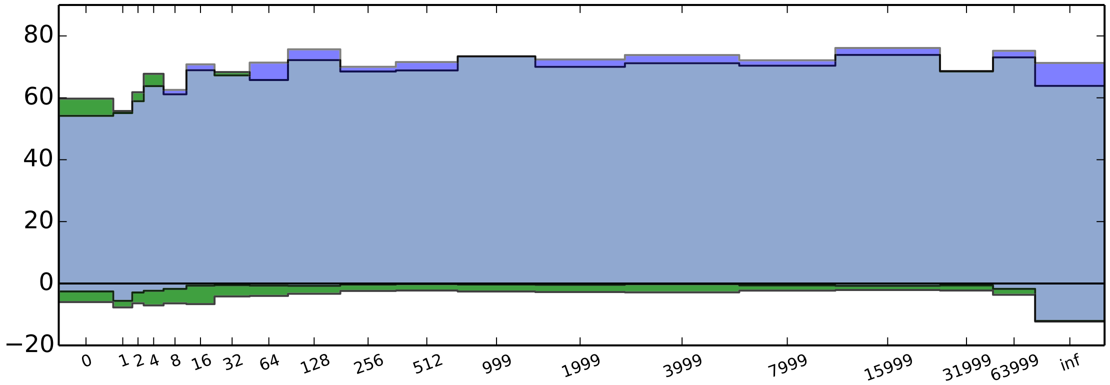


Figure 7.2: Accuracy of translation and deletion rates of NN (noun) tokens by frequency of the source word's type. SMT (light blue) and NMT (dark green).

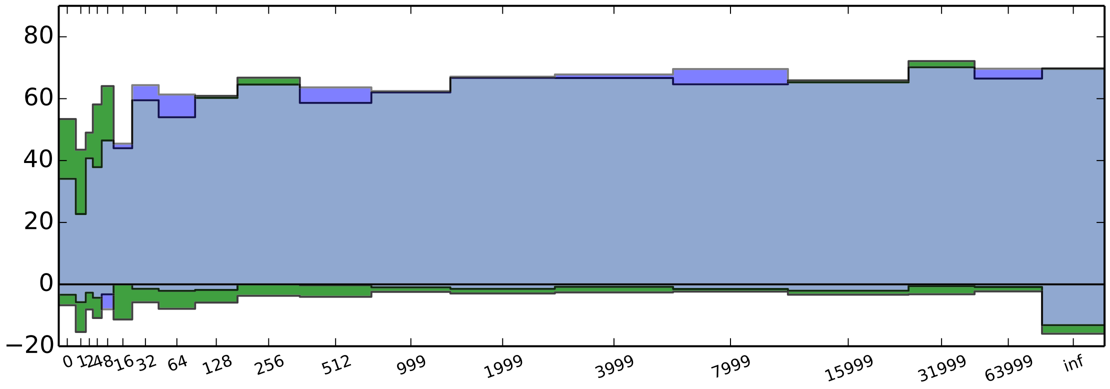


Figure 7.3: Accuracy of translation and deletion rates of ADJ (adjective) tokens by frequency of the source word's type. SMT (light blue) and NMT (dark green).

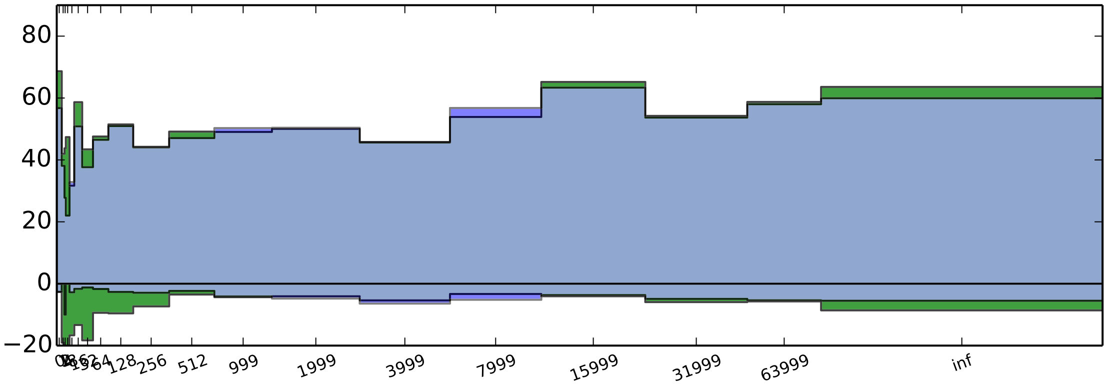


Figure 7.4: Accuracy of translation and deletion rates of V (verb) tokens by frequency of the source word's type. SMT (light blue) and NMT (dark green).

inflected forms (e.g., adjectives, verbs). Consider, for example, this inflected adjective "hochgiftiges"[6] meaning "highly toxic" (segmented as "hoch@@ gif@@ tiges", though a more morphologically meaningful segmentation would be "hoch | gift | ig | es"). Both the NMT and SMT systems accurately translate this word, despite never having observed it in training. While this particular inflection was unobserved, other inflections of the word were observed in training, and all of these share the same stemmed form ("hochgiftig"), allowing the SMT system to successfully translate the novel form. The NMT system is also successful, despite breaking up morphemes in the byte pair encoding segmentation. There are also many numbers that were unobserved in the training data; these tend to be translated correctly (with occasional errors due to formatting of commas and periods, resolvable by post-processing).

We show examples in Figures 7.5 and 7.6 of situations where the NMT system

[6] Neuter, nominative/accusative, singular form of "hochgiftig".

Source	... **choreographiertes** Gesamtkunstwerk ...
BPE	**chore@@ ograph@@ iertes**
NMT	... **choreographed** overall artwork ...
SMT	... **choreographiertes** total work of art ...
Reference	... **choreographed** complete work of art ...

Figure 7.5: Example 1: a word that was unobserved in the training corpus, successfully translated by NMT.

Source	... die Polizei ihn **einkesselte**.
BPE	**ein@@ kes@@ sel@@ te**
NMT	... police **stabbed** him.
SMT	... police **einkesselte** him.
Reference	... police **closed in on** him.

Figure 7.6: Example 2: a word that was unobserved in the training corpus, unsuccessfully translated by NMT and SMT.

succeeds and fails, and contrast it with the failures of the SMT system. In Example 1, the NMT system successfully translates the unobserved adjective *choreographiertes* (choreographed), while the SMT system does not. In Example 2, the SMT system simply passes the German verb *einkesselte* (closed in on) unchanged into the output, while the NMT system fails silently, selecting the fluent-sounding but semantically inappropriate “stabbed” instead.

While there remains room for improvement, NMT systems (at least those using byte pair encoding) perform better on very low-frequency words than SMT systems do. Byte pair encoding is sometimes sufficient (much like stemming or compound-splitting) to allow the successful translation of rare words even though it does not necessarily split words at morphological boundaries. As with the fluent-sounding but semantically inappropriate examples that have been observed for domain mismatch (Koehn and

Knowles, 2017), NMT may sometimes fail similarly when it encounters unknown words even in-domain. This phenomenon (which we observe in the "einkesselte" example) has also been noted by Arthur, Neubig, and Nakamura (2016), who show examples of mistranslations of low frequency content words (e.g., substituting "Tunisia" for "Norway"). In the following section, we provide analysis of two related issues to those considered in this section: how context influences translation, and when tokens are copied rather than translated.

7.3 Context and Copying

This section contains work published in the following: Knowles and Koehn (2018a).

In translation, certain tokens – such as names and numbers – should almost always be copied from the source sentence to the target sentence. As observed in Section 7.2, word copying is fairly straightforward in phrase-based statistical machine translation, where unknown words can be left untranslated (copied to the target side – one of the ways that statistical machine translation systems could succeed at translating out-of-vocabulary words).[7] It poses more of a challenge in neural machine translation systems, which often use limited or subword vocabularies and soft attention rather than strict alignment. The use of subword vocabularies means that in order for words to be copied, a whole sequence of tokens must be copied, one subword at a time.

[7] Due to the coverage tracking in SMT systems, they are encouraged or required to produce a translation for every word, even if that simply means copying a source word.

Nevertheless, neural machine translation models that use subword vocabularies to perform (near) open-vocabulary translation have been observed to correctly translate unknown words or copy words even when the full word to be translated or copied was not observed in training. As described in Section 7.2, we found that neural machine translation systems using subword vocabularies outperformed phrase-based statistical machine translation systems on the translation of unknown words, which does include copying (Koehn and Knowles, 2017).

The challenge of copying in neural machine translation has resulted in a variety of approaches to copying, which make use of pre-/post-processing and/or network modifications (e.g., explicit switching between generation and copying). By modifying the available training data rather than the neural architecture, Currey, Miceli Barone, and Heafield (2017) find that training a neural machine translation system to do both translation and copying of target language text improves results on low-resource neural machine translation and learns to pass untranslated words through to the target. They do this by mixing monolingual target data (as source-target pairs) with parallel training data. In contrast, Khayrallah and Koehn (2018) find that this dramatically hurts performance (in a higher-resource setting). These network- and data-modifying approaches are discussed in Section 7.3.1.

Together, these observations raise questions that we seek to answer here: to what extent does byte pair encoding (Sennrich, Haddow, and Birch, 2016c) solve the copying problem (without requiring modifications to the network structure)? More generally, what are subword neural machine translation models learning about copying?

We address this by focusing on two main questions: (1) Do certain *contexts* encourage copying? (2) Do certain *words* exhibit features that make them more likely to be copied (regardless of context)?

In this section, we do not modify the machine translation system to influence copying performance; instead we provide an analysis of standard existing systems. We find that neural machine translation systems (with attention, trained on joint source-target subword vocabularies) learn to copy words (both novel and observed) based on their sentential contexts. Additionally, though the models have no knowledge about the components of each subword unit, they learn that certain categories of tokens (e.g., capitalized tokens) tend to be copied. We use quantitative and qualitative evaluations to shed light on what these models learn about copying tokens and about the contexts in which copying occurs.

7.3.1 Related Work on Copying

Quite a bit of prior work has focused on the challenge that rare or unknown words pose to neural machine translation systems, as well as copying words in particular. We provide an additional discussion of other rare word issues in Section 8.2. Broadly, this can be divided into work that modifies the training data and work that modifies the network. We first discuss data augmentation techniques and then describe network modification.

Luong et al. (2015) augment data with word alignments to train neural machine translation systems (without attention) that emit both a translation and source word positions for any out-of-vocabulary (OOV) tokens emitted. Using automatic alignments between source and target training sentences, they propose three data augmentation schemes. Their "copyable" model assignes a unique UNK token to each unknown word in the source sentence. Any unknown target side word aligned to a source unknown token is assigned the same unique UNK token as its aligned source word. The remaining target side unknown words are assigned a special UNK_{null} token. Their "positional all" model uses a single UNK token, but inserts a positional token after each word, which indicates the relative position of its aligned source word. To limit the number of positional tokens used, aligned words more than 7 tokens apart are considered unaligned and assigned a *null* positional token. The best performing of their models is the "positional UNK" model, which uses a single UNK token on the source side and then uses unique target side UNK_i tokens to indicate that the aligned source word is i tokens away (for $i \in \{-7, \ldots, 7\}$). They then post-process OOVs with a dictionary lookup or copying of the aligned source word.

Currey, Miceli Barone, and Heafield (2017) augment training data with monolingual target language text as bitext (where the source and target are identical target language sentences). They find that in a low-resource setting this training data combination produces BLEU score gains and improves accuracy for copied words like named entities for translation between the following language pairs: English–Turkish, English–

Romanian, and English–German (in both translation directions). Ott et al. (2018) point to copied *source* sentences (where the source and target are identical source language text) as a source of degradation for overall machine translation quality in experiments on English–German and English–French translation. Khayrallah and Koehn (2018) found that such copied source sentences resulted in major BLEU score quality decreases in a higher-resource setting; in fact, they found them to be the worst possible kind of noise in terms of their impact on translation quality for German–English translation.

The work described so far has only involved modifications to (or observations about) the training data. We turn now to approaches that involve network modification. Both Gu et al. (2016) and Gulcehre et al. (2016) modify neural sequence to sequence models to explicitly perform copying. Gu et al. (2016) focus on monolingual tasks (dialogue systems and summarization), proposing a model that can both generate and copy text. In CopyNet, the output vocabulary for a particular sentence consists of the standard vocabulary, an UNK token, and the full source sentence vocabulary (which may contain tokens that would otherwise be considered out-of-vocabulary). At each timestep, the probability of generating a token is the combined probability of producing the token in the standard manner and the probability of copying it from the source sentence; they describe this as the two modes "competing through a softmax function." Gulcehre et al. (2016) perform experiments on neural machine translation (with attention), using whole-word vocabularies (and an UNK token to represent unknown words). Their model incorporates a switching variable that determines whether to copy or generate a translation, and two softmax layers to do the copying and generation: one to predict a source sentence token location (for copying) and one to predict the output word from a shortlist vocabulary. They find that using the pointer softmax model improves BLEU scores for English–French translation.

In this chapter's work, we focus on subword vocabularies for neural machine translation, using byte pair encoding (BPE; Sennrich, Haddow, and Birch (2016c)). The other approaches described above are somewhat orthogonal to the use of subword vocabularies, but may require modifications to handle subwords.

7.3.2 Data and Models

We train German–English (DE–EN) and English–German (EN–DE) neural machine translation models with attention, similar to the University of Edinburgh's WMT 2016 submissions (Sennrich, Haddow, and Birch, 2016a). Models are trained using the Marian toolkit (Junczys-Dowmunt et al., 2018). We use recommended settings and early stopping,[8] with results comparable to WMT 2016 systems: BLEU scores of 39.9 (DE–EN) and 33.2 (EN–DE) on the 2016 test set. We use the WMT parallel text[9] (Europarl, News Commentary, and CommonCrawl) along with synthetic backtranslated

[8] These include: model type `amun`, vocabulary of 85000, embedding dimension 512, RNN dimension 1024, one layer GRU encoder and decoder, layer normalization, dropout, early-stopping, and Adam (Kingma and Ba, 2014) for optimization. Decoding was performed with beam size 6 and length normalization (set to the default of 0.6).

[9] `http://www.statmt.org/wmt16/translation-task.html`

data.[10] The system is trained with a joint source-target vocabulary, but without tied embeddings.

7.3.3 Initial Analysis

We analyze the training data to learn about the prevalence and characteristics of words that should be copied in translation and the contexts in which they occur. We consider both the full training data (including backtranslations and CommonCrawl) and cleaner subsets. We restrict our search for copied words to tokens of length 3 or more characters. This has the benefit of removing words like *in* which are the same in German and English, but may nonetheless be considered translations rather than copies. Our heuristic for detecting copied tokens is this: a word is a "copied token" if it appears the same number of times in both the source and target sentence.[11] As we will show, copied words tend to belong to specific categories (proper nouns, numbers, etc.) which coincide with their repeated appearance in certain contexts (e.g., names following titles like "Ms" or "Prime Minister").

[10] `http://data.statmt.org/rsennrich/wmt16_backtranslations/`

[11] In DE–EN, we find one notable exception to this heuristic – *was* – which is a homograph, not a copy. It makes up $< 1\%$ of copied tokens in Europarl/News Commentary.

Data	**% Tokens Copied**	
	DE	*EN*
Europarl	1.8%	2.0%
News Commentary	2.9%	3.3%
Full Training (EN–DE)	7.6%	8.1%
Full Training (DE–EN)	8.6%	9.2%

Table 7.2: Percentage of tokens which should be copied, as measured across training data sources. The *DE* column indicates the percentage of the tokens in the German text that it was determined should be copied, while the *EN* column indicates that for the English side of the given data.

7.3.3.1 Where do copied words appear?

In Table 7.2, we see that between 1.8% and 9.2% of tokens are copied.[12] Though the majority (or near-majority) of sentences do not contain any copied words (of length 3 or more), copied words are still quite prevalent: approximately 18% of sentences in each full training dataset contain one, 4% to 5% contain four, and there is a long tail (one sentence contains 70). Sentences with many copied words often contain direct quotations, third language text (not source/target), or a sequence of copied words (e.g., comma-separated numbers or names).

The cleaner Europarl and News Commentary corpora have lower percentages of copied tokens than the overall training data. Of particular note, the backtranslated data contains some examples of copying that we'd prefer for the system *not* to learn, such as target language words appearing untranslated in the (backtranslated) source side data.

[12] The two full training sets differ due to the synthetic backtranslated data; the rest of the corpora are identical.

7.3.3.2 What words are copied?

We first examine the part-of-speech (POS) tags[13] of copied words. In the EN–DE training data, most copied words are tagged on the English side as NNP (proper noun, singular), including names of individuals, places, or organizations (e.g., González, Wales, Union). The next most frequent categories are CD (cardinal number) – including numbers like *42* that should be copied and ones like *seven* which should be translated – and NN (noun, singular or mass). The results are similar for DE–EN training data (tagged on German with a different tag set): PROPN (proper noun) is the most frequent tag for copied words, followed by NUM (numbers) and NOUN. Punctuation would rank highly if we included short tokens.

7.3.4 Contexts

In this section, we address our first question of interest: Do certain *contexts* encourage copying? Working from the intuition that certain contexts indicate that copying should occur – for example, a name following a title like “Ms” or “Frau” should often be copied – we examine the relationship between context and copying. We show that the machine translation system learns that certain contexts are so indicative of copying that it will even copy (not translate) words that it has *learned to translate* if they are seen in a sufficiently copy-prone context. We use left bigram

[13] POS tags are generated by the Stanford POS tagger (Toutanova et al., 2003). For English: english-left3words-distsim.tagger. For German: german-ud.tagger.

contexts as a proxy to evaluate the contexts in which words are copied.[14] For each POS, we collect the full set of left bigram contexts that ever precede a word with that tag, then filter by frequency and subsequent token diversity. We then filter by frequency and diversity of tokens following the bigram. The following section describes this process in more detail.

7.3.4.1 Collection of Contexts

For each POS, first, we find all left bigram contexts $(tok0, tok1, copiedword)$ that occur in the training data (where the copied word was tagged with the given POS). We then filter this set so that it only contains contexts $(tok0, tok1)$ that appeared at least 1000 times (left of NNP/PROPN) or 500 times (left of CD/NN/NUM/NOUN) in the training data. To ensure that we're not simply capturing collocations ("European Union"), we filter out left bigram contexts that have been followed by fewer than 150 unique types with that particular POS. This results in between 53 and 276 contexts, depending on POS, as shown in Table 7.3.

Each context is then associated with a copying rate, calculated as the number of times the token (with the given POS tag) following $(tok0, tok1)$ is copied, divided by the total number of times $(tok0, tok1)$ was observed to be followed by a token with that POS tag. In Table 7.4, we show the most- and least-copy-prone contexts for EN–DE (those with the highest and lowest copying rates).

[14] Since neural machine translation systems have access to both left and right context, there is reason to expect that right context also plays a role, but we leave that for future study.

POS	**Num. Contexts**
NNP	176
NN	82
CD	74
PROPN	276
NOUN	66
NUM	53

Table 7.3: Context counts by POS tag (NNP, NN, CD for EN–DE; PROPN, NOUN, NUM for DE–EN), selected as described in Section 7.3.4.

POS	**Context**	**Copy Rate**
NNP	Finance Minister	94.5%
	rates for	94.0%
	congratulate Mr	91.7%
	between the	10.5%
	President ,	7.7%
CD	updated on	94.0%
	the B	0.1%
NN	notified when	97.3%
	the first	0.6%

Table 7.4: Left bigram contexts with the highest/lowest copying rates (EN–DE), by POS tag.

For each context-POS pair, we select 50 random templates from the training data containing the bigram context followed by a word with that POS. We select contexts and templates from the full training data, rather than only the cleaner Europarl/News Commentary data, because we are interested in what patterns the model is learning from *all* data to which it has been exposed. Each context-POS pair is associated with a percentage that represents how often it exhibited copying in the training data. For example, in the copy-prone context "Finance Minister [NNP]" the NNP was copied

S: Therefore, Mrs Ashton, your role in this is invaluable.
R: Darum, Frau Ashton, ist Ihre Aufgabe in diesem Zusammenhang von unschätzbarem Wert.
T: Therefore, Mrs [NNP], your role in this is invaluable.
E1: Therefore, Mrs BBC, your role in this is invaluable.
D1: Deshalb, Frau BBC, ist Ihre Rolle hierbei von [...]
E2: Therefore, Mrs June, your role in this is invaluable.
D2: Deshalb, Frau June, ist Ihre Rolle dabei von [...]
E3: Therefore, Mrs Lutreo, your role in this is invaluable.
D3: Daher, Frau Lutreo, ist Ihre Rolle hierbei von [...]

Table 7.5: Source, reference, template, and examples of template-token combinations. *E1* has a word usually (76.0% of the time) copied in training, *E2* has one rarely (0.8% of the time) copied, and *E3* has a novel one. In training, 84.8% of NNPs with this left bigram context (", Mrs") were copied.

94.5% of the time, compared to 10.5% of the time in "between the [NNP]". For DE–EN translation, we see similar patterns: two of the three most copy-prone PROPN left bigram contexts are "sagte Frau" and "sagte Herr" ("said Ms/Mr"), while many less copy-prone ones end with articles.

7.3.4.2 Collection and Labeling of Copy/Non-Copy Words

We take all word types with a given POS tag from the WMT 2016 test set, dividing them into four categories based on two binary distinctions: *observed* (in training data) or *novel* (not observed in training), and *copy* (typically copied) or *non-copy* (not typically copied) and filter the observed ones based on training frequency. We count words as *non-copy* if they were copied $\leq 30\%$ of the time, and as *copy* if they were copied $\geq 70\%$ of the time.

All words that we examine are labeled as either *copy* or *non-copy*. For words

	Novel		Observed	
POS	**Copy**	**Non-C.**	**Copy**	**Non-C.**
NNP	96	22	251	263
NN	14	16	13	1664
CD	3	29	60	44
PROPN	92	76	463	418
NOUN	12	222	29	2176
NUM	2	29	55	68

Table 7.6: Counts of each word type by novel/observed, copy/non-copy distinction and POS tag (NNP, NN, CD are EN–DE; PROPN, NOUN, NUM are DE–EN).

that were observed in training, we discard those that appeared fewer than 1000 times. We label the remainder as *copy* if they were copied ≥ 70% of the time in training data (according to the heuristic described in Section 7.3.3), and as *non-copy* if they were copied ≤ 30% of the time in training data. For words that were unobserved in training, we used the same copying threshold but calculate it over all instances in the test data (with no requirement that they appear a certain number of times). Table 7.6 shows the number of words selected after filtering and thresholding.

7.3.4.3 Translation Experiments

We then combine each word with each POS-appropriate example template and perform preprocessing (including BPE) and translation.[15] Table 7.5 shows examples.

For each context, we calculate the percentage (across all example templates for that context and all words, separated by observed/novel and copy/non-copy categories)

[15] We use the Marian batch decoder, with recommended settings: beam size 6 and length normalization penalty of 0.6.

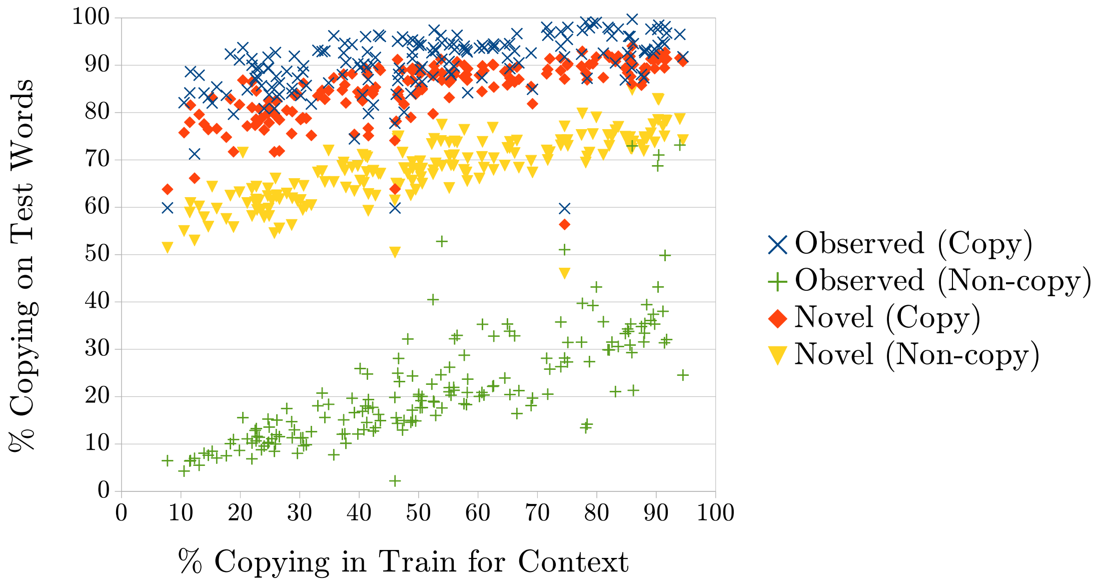


Figure 7.7: Percent of NNP (EN–DE) tokens copied by how copy-prone the context is, by category. Each point is the percentage of copying for all within-category words, across all example templates for one particular context (averaged over between 1,100 (novel-non-copy) and 13,150 (observed-non-copy) binary copy values).

of the time that the words in that context were copied. We then compare it to the percentage of the time that copying occurred for that context-POS tag pair in training.

Figure 7.7 shows NNP (EN–DE) results. Both observed-copy and novel-copy words behave almost identically, with copying percentages generally above 80%, and a slight trend upward as contexts become more copy-prone (moving to the right along the horizontal axis). Novel-non-copy words shadow these, but with a drop in copying percentage (see Section 7.3.5.2). Most interesting is the observed-non-copy category. In contexts that are not copy-prone, minimal copying occurs.[16] However, as they

[16] Note that some of the non-copy words were sometimes copied in training data, even if only in backtranslations.

are placed in increasingly copy-prone contexts, even these words that the system has learned it should *translate* are being copied. We observe the same trend for words tagged NN and CD, and for PROPN, NOUN, and NUM words in the DE–EN direction. Fig. 7.8 shows DE–EN PROPN (proper nouns). It shows similar trends to Fig. 7.7, but with a greater gap between novel- copy/non-copy words. This demonstrates that the machine translation system has learned that certain contexts are copy-prone.

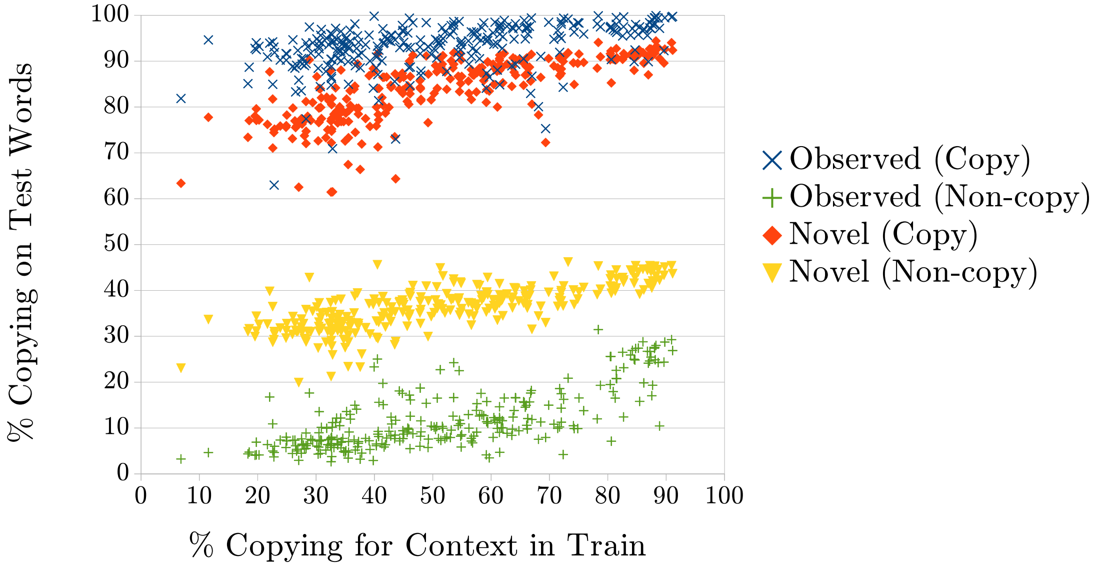


Figure 7.8: Percent of PROPN (DE–EN) tokens copied by how copy-prone the context is, by category. Each point is the percentage of copying for all within-category words, averaged across all example templates for one particular context.

We manually analyze outliers that appear much more or less copy-prone than expected. In both cases, the cause appears the same: the context occurred repeatedly in many very similar sentences in the training data. Highly copy-prone contexts that produced copying percentages greater than 70% even in observed-non-copy tokens

	Drop	**Change**	**Other**
Novel-Copy	24	102	60
Novel-Non-Copy	14	128	50
Observed-Copy	51	6	126
Observed-Non-Copy	12	1	186

Table 7.7: Counts of automatically detected output categories (*drop*, *change*, and *other*) for a sample of NNP tokens (EN–DE) that were not copied.

often appeared in common boilerplate text (e.g., "stay at [NNP]" or "rates for [NNP]" followed by "Hotel").[17] Where we observe lower than expected rates (e.g., ") of [NNP]"), we find that the system may have memorized training sentences.

7.3.5 Words

In this section, we examine what is happening to words when they are not copied, and take a closer look at both the types of translation behavior occurring, as well as features of the words themselves. This enables us to consider the second question of interest: Do certain *words* exhibit features that make them more likely to be copied (regardless of context)?

7.3.5.1 Analysis of Words That Are Not Copied

When words are not copied, what sort of output is the system producing? We find that it typically falls into one of four categories: *drop* (no target token aligns with the source token), *change* (the word is changed: partially translated, transliterated, or

[17] Since hidden representations contain whole sentence information, right side context may influence copying too, though we leave a more detailed analysis to future work.

inflected even if it is not a target language word), *substitution* (the word is replaced with a fluent but not adequate substitute), or *translation* (translated into a target language word).

We begin with an automatic analysis. We randomly sample 200 examples each of sentences containing words that were not copied for novel-copy, novel-non-copy, observed-copy, and observed-non-copy NNPs (EN–DE). We retranslate each sentence and produce a soft alignment matrix from the attention mechanism, then convert the soft alignments between BPE segments into hard alignments between the source word and one or more target words. We produce soft alignments (the attention matrix) using the AmuNMT decoder with the "return-nematus-alignment" flag set (Junczys-Dowmunt, Dwojak, and Hoang, 2016). It performs normalization differently than Marian's decoder (producing slightly different outputs for many sentences, including sometimes copying words that were not copied in our original translations).

For each target (subword) token, we align it to the source (subword) token with the highest soft alignment weight. Given our source word of interest s (composed of subword segments $s_1 \dots s_n$), we define its translation to be the list of all target words t (composed of subword segments $t_1 \dots t_m$) for which any subword t_i was aligned to a subword s_j of s.

A word has been *dropped* if it is unaligned. We count a word as being *changed* if any words it is aligned to have any subword (BPE segment) overlap with the original word's subwords. Both *substitution* and *translation* fall under *other*; we analyze those manually.

Results are shown in Table 7.7.[18] For all novel words, the most frequent output type is *change*. For example, the novel NNP *Bishnu* is changed into *Bischnu* in German.[19] Other changes include translations of parts of the word, and concatenation with other tokens. The output token often starts with the same character or sequence of characters as the source token.

We manually inspect examples in the *other* category. For observed-non-copy words, almost all are translations (e.g., *Sea* translated correctly as *Meer*), as expected. For observed-copy words, we see a mix of translations and other changes to the words, which are almost evenly split between substitutions and small changes. These include inflections (e.g., *Bremen magazine* reasonably translated as *Bremer Magazin*[20]).

There are also partial translations when BPE segments are full source language words – like *Thneed* (segmented "Th@@ need") becoming *ThNotwendigkeit* (segmented "Th@@ Notwendigkeit" – *Notwendigkeit* is a valid translation of *need*). Sometimes, a token *is* copied but then concatenated with another token.

Even without overlap of BPE segments between the source and the translation, changed words sometimes share a number of characters (especially at the beginning or end of a word). Half of the *other* category output of *Thneed* ("Th@@ need") begin with the letter "T" (but not the BPE token "Th@@"). This may suggest some level of

[18] Rows do not sum to 200 because some words in our random sample were copied by the the AmuNMT decoder.

[19] A near-transliteration – the "sh"/"sch" transformation is seen in EN–DE cognates, e.g., "ship" and "Schiff".

[20] *Bremen* and *Bremer* are unique BPE segments, so the *change* heuristic could not be applied.

character-awareness in the representations of BPE segments, produced as a byproduct of training, but we do not examine this in this work.

Within the *other* category, perhaps the most interesting cases are those where words appear to be substituted with a fluent but not adequate alternative. Many substitutions occur when the rare word is inserted next to a word that often forms a collocation (like “United States” – in sentences that include “in the [NNP] States” the translation sometimes defaults to a translation of “United States” regardless of the actual NNP inserted in place of “United”). Others have a less common NNP swapped for one that belongs to a similar semantic category (e.g., the place name *Dublin* being generated instead of the less common *Halle* – as Arthur, Neubig, and Nakamura (2016) and others observed). These findings provide additional support and nuance to the study of this phenomenon of neural machine translation system errors. Many substitutions occur when the rare word is inserted next to a word that often forms a collocation (like “United States” or “European Union” or “Madam President”). For example, in a template where “in the [NNP]” is followed by “States”, inserting the NNP *Accies* results in “in the *Accies* States” – which was then translated by the system as “in den Vereinigten Staaten” (gloss: “in the United States”). We also observe examples that may have to do with a combination of (in)frequency of tokens and the context. For example, we have the novel NNP *Sloveina* (perhaps a misspelling of *Slovenia*), which is often replaced with *Slowaken* (*Slovakia*) when translated to German in various different context templates. In another sentence, we find that “this year, *Angela* expects” is translated to “in diesem Jahr erwartet *Merkel*” despite *Merkel* appearing nowhere in the source text. The first and last names of German chancellor Angela Merkel appear frequently together in training data, and thus likely have sufficiently similar representations. We see other similar substitutions: *Mitt* for *Romney*, *US* for *Obama*, and *Thomas* for *Sarah*. Sometimes a specific name is replaced with a title, such as “your prime minister, *York*” being translated as “ihr Premierminister, *Herr Präsident*” (glossed as “your prime minister, *Mr. President*”). For novel-copy words labeled as *other*, three quarters are substitutions and one quarter exhibit small changes. The reverse is true for novel-non-copy words: the majority exhibit small changes while almost thirty percent are substitutions.

7.3.5.2 Properties of Copied Words

Certain words exhibit properties that make them more likely to be copied, regardless of context. At first glance, it seems unintuitive that the rate of copying of novel-copy words and novel-non-copy words differs (Fig. 7.7) – the model has never observed any of these words, and they are being presented in identical contexts to one another – why does it differentiate between them? Doing so indicates that the model has learned what makes a sequence of subwords likely to be copied.

Belinkov et al. (2017) observe that neural machine translation models may encode information about part-of-speech, which could be used when determining whether or not to copy (but does not explain within-POS differences). For numbers, it mainly

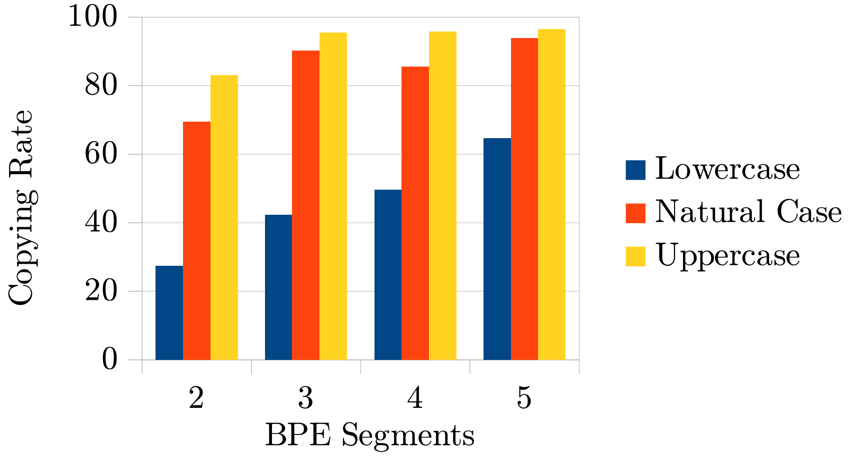


Figure 7.9: Copying rate based on casing and number of BPE segments for novel NNP words (EN–DE), averaged across all NNP contexts.

learns to copy numerical portions while changing commas to periods and vice versa (as required by the target language's conventions). Nouns and proper nouns are more interesting: some should be translated (e.g., novel noun compounds like *hallmate*), or, in the case of misspellings (e.g., *manfacturer*), corrected, while others should be copied. For novel NN words, there is another striking difference between copy and non-copy: most of the former contain capital letters and most of the latter do not.

7.3.5.3 Capitalization and Copying

To experiment with the influence of capitalization on copying, we take each novel NNP word (96 copy and 22 non-copy) and convert it to all lowercase, leave it in its natural case (all have at least one uppercase letter), or convert it to all uppercase letters. We then translate all of them in all NNP contexts (from previous EN–DE experiments). Using only novel words sidesteps the issue of truecasing.

Lowercase words are the least frequently copied (average copy rate of 40.2%), uppercase words are the most copied (94.4%), and the natural case falls in the middle (81.7%). However, changing casing changes the BPE segmentation, and uppercase words tend to be split into more pieces: a mean of 4.4 segments, as compared to means of 3.1 (lowercase) and 2.9 (natural case). The number of subword segments correlates positively with copying rate (Fig. 7.9), but, controlling for that, we still find that NNP words that are completely capitalized tend to be copied more than those with the same number of subword segments but only lowercased letters, suggesting that the system is encoding information about the connection between capitalization and copying. We also perform this experiment with PROPN words in the DE–EN direction, and find that increased capitalization increases copying, though we do *not* find there that an increase in the number of BPE segments increases copying. This occurs despite the capitalization of all nouns in German. Figure 7.10 shows these results for DE–EN. The true casing of the word consistently falls between these two extremes. The high copying rate of fully-capitalized words is intuitive: acronyms are often both uppercased and copied from source to target. That is not to say that the model always learns to copy acronyms; it also learns to translate them when appropriate (such as *GDP* to *BIP*). There is always an interplay between learned translations and features that may encourage copying.

The connection between copying rate and capitalization provides one explanation

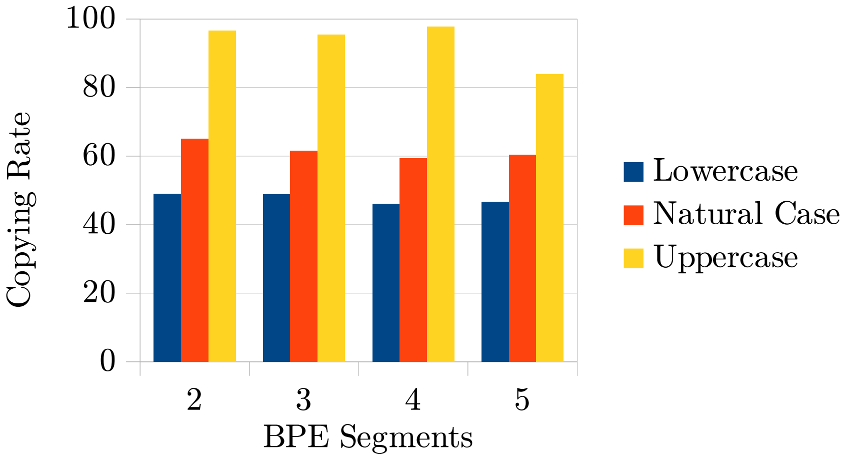


Figure 7.10: Copying rate based on casing and number of BPE segments for novel PROPN words (DE–EN), averaged across all PROPN contexts.

for the gap in behavior of the two novel word types, and demonstrates that features of words influence copying. Note that it learns this behavior based on parallel training data, without access to information at a finer granularity (character-level) than the subword units; the model is never explicitly told that certain subwords contain capitalization.

7.3.6 Conclusion

We show that subword vocabulary neural machine translation systems learn about copying from context and the subwords themselves. The effect of context is strong enough to cause words that would otherwise be translated to be copied. Characteristics of subword tokens play a role in copying behavior, with capitalized tokens more likely to be copied. We leave as future work a deeper analysis of the level of character-awareness encoded in representations of the BPE segments as a byproduct of training. We provide an analysis of what happens when words are not copied, showing expected differences between novel words and words that were observed during training. Additionally, we provide more examples and evidence of the problem of substituting fluent but non-adequate translations for rare or unknown words. All of this provides useful context for examining how neural models translate or copy known and novel words, as a starting point for examining how to adapt them to perform better on those challenging words.

7.4 Consistency

In any given document, a translator may encounter a number of words or phrases for which there exist multiple valid translation options.[21] The choice of a particular translation option may be influenced by many factors: formality, fixed terminological resources or style guides that they are required to adhere to, dialect, intended audience, and so on. Within a document or translation project, these choices are not made in a vacuum. As Carpuat (2009) observed, there is a tendency for translators to produce translations such that the "one translation per discourse" hypothesis holds within a particular document.[22] That is, human translators tend to prefer consistent translations of individual terms throughout a document. Other work on "translationese"

[21] A portion of this section draws on work published in Kothur, Knowles, and Koehn (2018).

[22] This work follows from "one sense per discourse" (Gale, Church, and Yarowsky, 1992), which found that the vast majority of polysemous words share only one sense within a given document.

has also found that translations show regularities in syntax and punctuation (Baroni and Bernardini, 2005).

We examine human translator consistency with an eye towards improving machine translation consistency in a computer aided translation setting. Even expanding beyond words with multiple senses or synonymous translation options, we expect that learning from the translator's lexical, syntactic, and stylistic choices at the beginning of a document should result in a well-tailored system that is better at translating subsequent sentences. We can think of fine-grained adaptation over a document as producing a document-specific machine translation system that encodes or highlights document context (even as the machine translation system still performs translation of each sentence individually).

7.4.1 Consistency Case Study

As a case study of consistency in translation, we examine the translation of the "(Laughter)" annotation in TED talk data. We first examine human consistency in translation, and then compare this to machine translation. This particular phrase is selected because of its frequency across many documents.

The Multitarget TED talk dataset (Duh, 2018) consists of transcriptions of TED talks (in English) along with their translations. As these talks are delivered in front of live audiences, the speaker's words are sometimes interspersed with audience laughter or applause, which is then included in the transcript. For example:

> "It was unequivocally not something a squirrel could chew on. *(Laughter)* But that in fact seemed to be the case."[23]

Subsequent translations of these transcripts often (but not always) include translations of these transcriptions of non-speech phenomena. Transcription guides generally have fixed style guides for how to incorporate these phenomena, and transcribers are encourage to be consistent. In the English TED training data (transcriptions in English of spoken English), transcribers are quite consistent in how they annotate laughter, typically using "(Laughter)" to indicate it. This follows official TED guidance on how to transcribe non-speech phenomenon like sounds (represented in parentheses) and on-screen text (in square brackets). In fact, "(Laughter)" is one of the examples shown in a training video on TED transcriptions. There are 4545 instances of "(Laughter)" in the $152,606$ lines of English TED training data and fewer than 5 instances in which a misspelled variant appears.[24]

There are multiple possible translations for "(Laughter)" when translating from English to German. If there is no specific translation lexicon enforced, we might find multiple translations in the data, and we do find this to be the case in English–German translation. Table 7.8 shows the distribution of the translations of the 4545 instances of "(Laughter)" in the training data. The list of translations is collected by manually examining a sample of German sentences aligned to English sentences containing

[23] From Andrew Blum's 2012 TED talk, *Discover the physical side of the internet*: `https://www.ted.com/talks/andrew_blum_what_is_the_internet_really`

[24] There are also 28 instances of "(Laughs)", which typically appears to be used to indicate the speaker's own laughter, rather than audience laughter. We actually compute these statistics over tokenized and lowercased text, but show the raw text here for readability.

"(Laughter)" and then the counts are obtained automatically by string matching in the set of German sentences aligned to English sentences containing "(Laughter)". The most common translation, "(Gelächter)", appears 49.97% of the time, while the second most common translation, "(Lachen)", appears 44.33% of the time. In some cases, no translation appears at all, or another infrequent translation is used.

We would, however, expect that human translations of individual documents would be *internally* consistent in how they translate. That is, a single document's translation will likely use the same translation for every instance of "(Laughter)" in the source document, but the translation may vary from document to document or translator to translator (unless there is a clear language-specific guideline for that translation). Observing that there are two very common translations of "(Laughter)" in the English–German data, we now examine whether they are used consistently within specific documents. We begin with the training data, which contains 1212 unique documents. Of these, 876 include at least one instance of "(Laughter)" and 686 include more than one instance. For each of the documents containing more than one instance, we count the number of appearances of each of the translations (as listed in Table 7.8). We label a document consistent if only one of the translations appeared in the German translation of the document. We label a document as inconsistent if multiple different translations from the list appeared. A total of 648 documents (94.5%) of the documents with more than one instance of "(Laughter)" on the source side in the training data were consistent, while only 38 (5.5%) were inconsistent. In the majority of inconsistent documents, exactly two different translations were used. Thus we observe that, consistent with expectations about human translators, most documents contain consistent translations of “(Laughter)”.

We now examine how machine translation consistency compares to human translation consistency. The model is trained using Sockeye (Hieber et al., 2017) on a concatenation of general domain data (WMT 2017) and Open Subtitles 2018 (Lison, Tiedemann, and Kouylekov, 2018) and then domain-adapted with continued training to TED data. Data is preprocessed with the Moses tokenizer (Koehn et al., 2007), lowercasing, and byte pair encoding with a vocabulary size of $30,000$ (trained on the general domain data and applied to all data). Since we use standard sentence-level NMT systems, each sentence in a document is translated independently from every other sentence in the document. This is, of course, in contrast to the human translator, who is translating each sentence with knowledge of its surrounding context. To examine this, we consider the test set, which contains 12 documents with more than one instance of “(Laughter)”. The human translators are completely consistent in each of these documents. The neural machine translation system, however, only translates 5 of these documents consistently. The remaining 7 documents are translated inconsistently by the neural machine translation system. In all inconsistent documents, translations are divided between “(Gelächter)” and “(Lachen)”, the two most common translations in the training data. In 4 of the 7 inconsistent documents the majority of instances are “(Gelächter)”, in 2 documents there is one of each, and in one document “(Lachen)” is

Translation of "(Laughter)"	**Percent of Examples**
(Gelächter)	49.97%
(Lachen)	44.33%
(Schallendes Gelächter)	0.35%
(Lacht)	0.31%
(Gelächter.)	0.20%
(Gelaechter)	0.18%
(Lachten)	0.07%
None/Other	4.60%

Table 7.8: Prevalence of various translations of the 4545 instances of "(Laughter)" in the English–German TED talk training data.

used more than "(Gelächter)".

The inconsistency of the neural machine translation system is a concern from a computer aided translation perspective. First, we know that human translators will prefer to use consistent translations within a document, so inconsistencies here will be errors that the translator will need to correct. Secondly, translators may be working within a particular style guide that may or may not match training data. Ideally, the system should be able to perform translations in accordance with the desired style guide or translation lexicon. We examine approaches to incorporating lexicons and improving consistency in Chapter 8.

7.4.2 Apparent Inconsistency

While examples of consistency are relatively easy to examine through automatic means – the "(Laughter)" example is particularly helped by the fact that it is always enclosed in parentheses – examples of inconsistency provide more of a challenge. We performed manual qualitative analyses of repeated source language tokens in English–German TED documents (ignoring verbs, which we know are more heavily inflected in German than in English, and whose consistency is therefore not expected[25]). In general, repeated nouns and noun phrases are translated quite consistently within a single document. Adverbs like “actually”, which can be translated as “eigentlich”, “wirklich”, or “tatsächlich” (among other translations) do not always appear to be translated consistently, because of subtle semantic distinctions between various uses of “actually”.[26]

In Table 7.9, we show examples of apparently inconsistent human translations of the English word “music” across several documents and how they compare to machine translations of the word. In most cases where the word “music” appears in the English sentence, the corresponding word in the German sentence is “Musik”. We also observe cases where we instead see compound nouns: “Musikformen” (in the context of “traditional dress and dance and music” translating to “traditionellen Trachten , Tänzen und Musikformen”), “Musikvideos” (“music videos”), or “Musiksendern” (“music channels”). Thus while it might appear that the word “music” is being translated inconsistently, it is more appropriate to consider that the multi-word phrase is being translated as a unit. We also observe cases where no translation appears, often due to differences of sentence structure that allow the word to be implied rather than explicitly stated. Finally, we see one case where “music” is translated as “Takt”

[25] We might still expect consistency in terms of the choice of verb itself, but the inflections are determined grammatically.

[26] This could arguably provide a counterexample to the “one sense per discourse” claim.

Doc.	**German (ref.)**	**German (MT)**
684	Musik (3), Musikformen	Musik (4)
710	Musik (2), Musikvideos, Musiksendern	Musik (2), Musikvideos, Musikkanälen
755	Musik (2), Takt (1)	Musik (3)
805	Musik (15), [none] (2)	Musik (15), [none] (2)

Table 7.9: Examples of translations of the word "music" in documents from the English–German TED development set, manually analyzed. Parenthetical numbers indicate counts when a particular translation appeared multiple times.

("beat" or "rhythm"), in a situation where "Takt zählen" likely sounds more fluent and appropriate than using "Musik" would. This occurs in the following sentence pair:

> "So, you can hear it not just in the phrasing, but the way they count off their music: two, three, four, one."
>
> "Das kann man nicht nur in der Phrasierung hören, sondern auch in der Art, wie sie ihren Takt zählen. Zwei, drei, vier, eins."

All of these examples suggest that human translators do tend toward consistency in translation, except when there are clear reasons (multi-word expressions, specific contexts, differences in cross-lingual semantics, etc.) to deviate.

The machine translation system, on the other hand, tends to be more consistent and literal than the human translator. In the example of "music", we sometimes see the machine translation system using "Musik" even when a different option might be more appropriate in the translation, with the notable exception of several of the compound nouns, which it also generates (or generates variations of).

Manual exploration uncovered one common example of easily measurable apparent inconsistency in human translation: the translation of the English word “percent”,[27] which is sometimes rendered as “Prozent” and sometimes rendered as “%”. There is some internal consistency in this behavior: “%” only ever appears after numerical tokens (tokens ending in an integer character in the range from 0 to 9). “Prozent”, which is overall more common in the training and test data, appears after both numerical and non-numerical tokens. Of the 340 training documents that contain multiple instances of “percent” on the source side, 228 are consistent (always using “Prozent” or always using “%”), but 112 are inconsistent (using a mix of the two). Initial intuition suggested that this might tie in with the fact that “%” only appears following numerical tokens – perhaps translators are following some internally consistent rules and alternating on the basis of the previous token. However, this is not generally supported by the evidence – only 17 of the 112 inconsistent documents follow that pattern. The machine translation system is more consistent than the human translators on test documents, remaining entirely consistent and always using the more common “Prozent”, while 5 of 7 human translated documents in the test set were consistent. While this is evidence of a human inconsistency in notation, the question of whether it should be considered an inconsistency in translation is more complicated: both “%” and “Prozent” are read aloud identically.

[27] The English word “percent” is much more common than “%” in the training source data. The latter appears there in fewer than 10 documents.

7.4.3 Conclusion

We observe through automatic metrics that human translators (but not machine translation systems) are extremely consistent within document (or within translator) in their translations of certain special tokens. In seeking to find cases of real inconsistency by human translators, the cases of apparent examples of human inconsistency that we did find ("actually" and similar adverbs, "music", and "percent") still suggest that human translators are quite consistent within the bounds of a single document, as described by Carpuat (2009). When they appear to vary from a consistent translation, there is often an alternate explanation to be found, such as a larger phrase being translated or a different grammatical structure being used in the target language. We find that in some cases the neural machine translation system is more consistent than the human (biased towards the more frequent of two nearly-interchangeable translations, as in the case of "percent"), while at other times it is less consistent (in the case of "(Laughter)").

Chapter 8

Fine-Grained Adaptation

This chapter draws on work from Kothur, Knowles, and Koehn (2018). The paper represents a collaboration, with equal contributions from the first two authors: I wrote the code for and ran the experiments on dictionary adaptation, while my coauthor, Sachith Sri Ram Kothur, wrote the code for and ran the experiments on single-sentence adaptation. We both contributed to the experimental design and analysis.

8.1 Introduction

The challenge of adapting a machine translation model to a new domain is a well-studied one, but even a strong domain-adapted system may be able to perform better on a particular document if it were to learn from a translator's corrections within the document itself. In fact, each new document may pose unique challenges due to novelty of vocabulary, word senses, style, and more.[1] It stands to reason that

[1] For example, Carpuat et al. (2012) decompose errors into seen, sense, score, and search; the first two are most relevant to this work.

fine-grained adaptation using information from within a document (for example, as it is being translated by a human translator in a computer aided translation (CAT) environment) could provide the added benefit of a closer in-domain match than existing approaches that use data from other documents within the same domain. We focus on adaptation within a single document – appropriate for an interactive translation scenario where a model adapts to a human translator's input over the course of a document. We propose two methods: *single-sentence adaptation* (which performs online adaptation one sentence at a time) and *dictionary adaptation* (which specifically addresses the issue of translating novel words). These two approaches are complementary, and we show that the combination of approaches outperforms baselines as well as each approach individually, resulting in an improvement of +1.8 BLEU points and +23.3% novel word translation accuracy on WMT news data and an improvement of +2.7 BLEU points and +49.2% novel word translation accuracy on EMEA data (descriptions of medications). Both approaches address aspects of consistency in translation, and the dictionary adaptation approach in particular focuses on improving recurring machine translation errors, as desired by human translators (Moorkens and O'Brien, 2017).

Continued training of neural machine translation (NMT) systems has been shown to be an effective and efficient way to tune them for a specific target domain (Luong and Manning, 2015). One such technique is incremental updating – comparing the system's predicted translation of an input sentence to a reference translation and

Source	**Reference**	**Baseline MT Output**
Ambirix (Ambi | rix)	Ambirix (Ambi | rix)	Hampshire, Glaurix, Tandemrix, ...
Prepandemic (Prep | an | demic)	Präpandemischer (Prä | pandem | ischer)	Proteasehemmer
Cataplexy (Cat | ap | lex | y)	Kataplexie (Kat | ap | lex | ie)	Cataplexy
hormone-dependent (hormon | e- | dependent)	hormonabhängig (hormon | abhängig)	hormonell

Table 8.1: Examples of novel words and their mistranslations. The subword segmentation (in parentheses) is indicated by "|" for the source and reference.

then updating the model parameters to improve future predictions. Though this is typically done in batches during training, a single-sentence pair or even a word and its translation can be treated as a training instance.

Computer aided translation provides an ideal use case for exploring model adaptation at such a fine granularity. As a human translator works, each sentence that they translate (or each novel word for which they provide a translation) can then be used as a new training example for a neural machine translation system. In an interactive translation setting or a post-editing scenario, rapid incremental updating of the neural model will allow the neural system to adapt to an individual translator, a particular new domain, or novel vocabulary over the course of a document.

As discussed in Sections 7.2 and 7.3, novel words are split by byte pair encoding into subwords, which the machine translation system may either copy or attempt to translate, with varying levels of success and with varying levels of consistency. We examine translation consistency in Section 7.4. Table 8.1 shows example mistranslations of novel words in the EMEA dataset.

We test our approaches (dictionary training and single-sentence adaptation) to fine-grained NMT adaptation on two very different domains: news and formal descriptions of medications, each of which provide their own challenges. In our datasets, just under 80% of news documents and just over 90% of medical documents contain at least one word that was unobserved in the training data. In the news documents, 12.8% of lines contain at least one novel word, whereas in the medical data, 38.3% of lines contain at least one novel word. We show that models can learn to correctly translate novel vocabulary items and can adapt to document-specific terminology usage and style, even in short documents.

8.2 Related Work on Rare Words and Adaptation

In this section we describe related work on rare words and fine-grained adaptation. Much like the approaches used for copying (as discussed in Section 7.3.1), the work on rare words can broadly be divided into approaches that require network modifications and approaches that involve data augmentation. We first describe network modifying approaches and then examine data augmentation. With the exception of some copying work, these approaches require knowledge of the rare words during training, meaning they are not applicable to novel words. We then describe work on neural machine translation adaptation using very small datasets, including approaches that can handle novel words (with the use of a subword vocabulary).

Arthur, Neubig, and Nakamura (2016) propose to improve the translation of rare (low-frequency) content words through the incorporation of translation probabilities from discrete lexicons into neural machine translation models. To begin with, values in the lexicon need to be converted into translation probabilities. For a count-based lexicon (such as one extracted through automatic alignment of a corpus), this can be done by computing expected counts and normalizing. For a dictionary-based lexicon, this can be done by assigning a uniform distribution to all translations of a word, and 0 to all words that are not a translation. These probabilities $p_l(y|x)$ (the probability of the target being y, given the source word x) are referred to as "lexicon probabilities" in the paper. Neural machine translation systems, however, use probabilities conditioned on the source sentence $\vec{x}$ and the previously translated words $\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}$ in order to determine the probability of the target $\hat{y}_{t-1}$; this is the model probability $p_m(y_t|\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}, \vec{x})$. For each source sentence, a matrix of lexicon probabilities (rows corresponding to output vocabulary items and columns corresponding to each token in the source sentence) can be computed, and the attention mechanism can be used to provide a lexicon probability from the weighted average of the columns. This can then be used to bias the model's standard probability distribution in favor of the lexicon probabilities. They find that their methods improve translation for English–Japanese in terms of BLEU score, in both directions.

Nguyen and Chiang (2018) propose to train a feed-forward neural network to generate a target word based directly on a source word. Like Arthur, Neubig, and Nakamura (2016) they then weight these probabilities using the attention mechanism and combine them with the standard translation approach. They perform their evaluation on eight language pairs, finding substantial BLEU score improvement especially for low-resource languages. As described in Section 7.3.1, Gu et al. (2016) propose a (monolingual) sequence-to-sequence model, CopyNet, that can select input sequences to copy to the output within the course of generating a single sequence. All of these approaches require modifications to the neural network architecture.

Fadaee, Bisazza, and Monz (2017) propose to learn better translations of rare words by generating new sentences that include them to add to the training data. Taking existing sentences from the training data and a list of rare words and their translations, a language model is used to score plausible replacements of words in training sentences with words from the rare word list. If a plausible replacement is found, automatic alignment is performed over the source and target sentences, and the aligned word in the target sentence is replaced by the translation of the rare word that has been inserted into the source sentence. This new modified sentence pair is added to the training corpus. They find that this improves translation quality and increases the number of rare words produced during translation (for both translation directions of English–German).

Work like Freitag and Al-Onaizan (2016) and Luong and Manning (2015) provide an approach to domain adaptation for neural machine translation by simply fine-tuning existing models on new – often smaller – datasets. Taking this idea to the extreme conclusion, we can consider ways of adapting to just a handful of sentences or even a single sentence, a task that is often relevant for computer aided translation or document-level translation.

Farajian et al. (2017) propose such a model in the context of handling translation requests from multiple domains on the fly with a single model. Given a new sentence to translate, they select sentence pairs from the training data that are similar to this new sentence, adapt the general model to those sentences through fine-tuning (dynamically adapting parameters on the basis of the similarity between the sentence to be translated and those source sentences retrieved from the training data), and then reset to the original model before translating the next sentence in the same manner. They find that this instance-based adaptation approach outperforms phrase-based statistical machine translation systems, generic neural machine translation systems, and oracle domain-specific machine translation systems on English–French translation. Li, Zhang, and Zong (2018) also proposes to learn a general domain model, select a batch of between 1 and 128 sentence pairs from the training data for which the source sentence is similar to the new sentence to be translated, and then fine-tune on this batch of sentences. They evaluate their approach on Chinese–English translation.

Turchi et al. (2017) examine several combinations of an instance-based adaptation approach like those described above and an approach where a model is incrementally updated on new sentence pairs (like our single-sentence adaptation approach). They demonstrate BLEU score improvements on a general domain English–German neural machine translation system used to translate information technology data, as well as a domain-specific English–Latvian system for translating data in the medical domain (EMEA data). In both cases, they use existing datasets of post-edits, though there is not a guarantee that their initial machine translation output matches the machine translation output that was used for the post-editing. Our experiments are complementary to theirs; they explore a high-resource but domain-mismatched scenario (information technology) and a lower-resource but domain-matched scenario (EMEA), while we examine a high-resource domain-matched scenario (WMT) and a high-resource domain-adapted scenario (EMEA). Karimova, Simianer, and Riezler (2018) have recently shown in a user study with translation students that an online adaptation approach (similar to our single-sentence adaptation approach and to Turchi et al. (2017)) can decrease post-editing effort.

8.3 Approaches

We propose two complementary approaches for adapting an NMT model over the course of a single document's translation and the combination of the two. For each approach, adaptation is done at or within the document level and the model is reset to baseline between documents.[2]

8.3.1 Single-Sentence Adaptation

In this approach, the model is iteratively adapted over the previous translated sentence (and its reference), then the updated model is used to translate the next sentence. Thus, line n of the document is translated by a model which has been incrementally adapted to all previous lines (1 through $n-1$) of the document. See Algorithm 2 for details. Such an approach could be applied in a computer aided translation tool, which would allow the machine translation system to adapt to translator corrections as produced by post-editing, through an interactive translation prediction interface, or any other computer aided translation approach. Single-sentence adaptation allows the model to learn the translator's preferred translations, which may be specific to the particular document. For example, the system might initially produce a valid translation for a word in the document, while the translator prefers an alternate translation; after single-sentence adaptation, the system can learn to produce the translator's preferred translation in future sentences.

[2] In cases where the domain is fairly homogeneous, it may be beneficial *not* to reset the model between documents, while in heterogeneous domains it may be desirable to always reset the model (or maintain several models, each of which is fine-tuned to a particular subdomain).

Algorithm 2 Single-Sentence Adaptation

```
$m_0$ : baseline model
$s, r$ : document (source language) and reference translation (target language)
$s_i, r_i$ : line $i$ of $s$ and $r$, respectively; zero-indexed
$n$ : number of lines in $s$ (and $r$)
$t$ : translation output (initially empty)

$t_0 \leftarrow \textsc{Translate}(m_0, s_0)$
for $i \in \{1 \dots n\}$ do
    $m_i \leftarrow \textsc{Adapt}(m_{i-1}, s_{i-1}, r_{i-1})$
    $t_i \leftarrow \textsc{Translate}(m_i, s_i)$
end for
$\triangleright$ We compute BLEU score between $t$ and $r$.
```

8.3.2 Dictionary Training

This approach aims to adapt models with the specific goal of better translating novel words. Given a new document to translate, we identify words that are novel (have not appeared in any training or adaptation data). Next, we obtain a single translation for each of these words (in a computer aided translation setting, this might consist of asking a human translator to provide translations; along the lines of terminology curation). In this work, we simulate the collection of such dictionaries (or terminology banks) using the reference. We then treat the list of novel words and their respective translations as bitext and continue model training, producing a model specifically adapted to this document's novel vocabulary, which we can then use to decode the complete document. Note that this is a very small bitext to train on, and each line of the bitext contains a single word (segmented into multiple tokens by byte pair encoding).

To simulate a translator-produced dictionary, we build a dictionary of novel word translations from the source and reference. First we run fast-align (Dyer, Chahuneau, and Smith, 2013) over the byte pair encoded representations of the source and reference sentences.[3] The target-side token whose subword segments most frequently align to the subword segments of the source-side token is selected as a candidate translation, and a single final translation is selected based on the most common candidate translation within the document. Note that, particularly for words with morphological variants in the target language, there may have been more than one correct translation. We account for this in evaluation, but only train on one translation option.

8.3.3 Single-Sentence Adaptation with Dictionary Training

Dictionary training and sentence adaptation offer distinct benefits when adapting over a document. Dictionary training helps the model learn the right translations for novel words and single-sentence adaptation can provide a more general adaptation. The latter can also learn correct translations of repeated novel words, but may require multiple instances to do so. Doing dictionary adaptation before adding single-sentence adaptation could ensure that the novel terminology is correctly and consistently translated from the beginning of the document, which could eliminate

[3] The fast-align model is trained over the byte pair encoded representations of the full training data: WMT data, backtranslations released by Sennrich, Haddow, and Birch (2016c), and EMEA data used for adaptation.

a pain point for human translators. In this combined approach, we begin with the document's dictionary trained model and use that as the initial model for single-sentence adaptation.

8.4 Data and Models

We use two datasets and baseline models to evaluate our approaches, translating from English into German. We evaluate on WMT news data and EMEA medical data using baseline WMT and EMEA domain adapted models, respectively. The different domains (news vs. medical) allow us to evaluate our approaches in different scenarios. While the data and models have been introduced in Sections 4.1.1 and 4.1.2, we elaborate on details like the document splits in the following sections. The news data is a very commonly used dataset, and features a range of news domains, with a wide range of vocabulary and styles, making it challenging for MT in general. The EMEA data, on the other hand, is highly repetitive and structured, but comes with a challenge of frequently repeated novel vocabulary in almost every document. Experimenting with both of these datasets lets us see the strengths and weaknesses of each of our approaches. For example, we would expect the dictionary adaptation approach to be particularly useful in the EMEA scenario (due to the high frequency of those novel words), so testing it on WMT as well gives us a chance to see how it performs in a more average scenario and not just its best case scenario.

8.4.1 WMT

WMT Data: We test on the full English–German WMT 2017 news translation test set (Bojar et al., 2017), splitting it into 130 unique documents (derived from the document splits in the original SGM file). Each document is a short news story. These stories are drawn from a number of news sources, covering a wide range of topics. While all documents are in the "news" domain, this is a fairly heterogeneous dataset. The documents range in length from 2 to 64 lines, with an average length of 22.1 lines (median 20).

We used the first 20 documents from the 2016 WMT news translation test set (Bojar et al., 2016) as a development set for selecting training parameters for dictionary training experiments, and a subset of 8 of these documents for selecting parameters for the single-sentence training experiments. The development set documents had a similar range of lengths (3 lines to 62 lines, with an average of 19.0).

The number of novel word types per document in our test set ranged from 0 (no novel words; no dictionary adaptation) to 15 novel words. There are 295 novel types (across all documents combined) and 442 novel tokens. Across the test set, 12.8% of lines contain at least one novel word. In some cases, up to 75% of the lines within a single document contain at least one novel word.

WMT Baseline Model: We use the University of Edinburgh's publicly available WMT 2016 English–German model, as described in Section 4.2.2.[4] As this was trained

[4] http://data.statmt.org/rsennrich/wmt16_systems

for the 2016 WMT evaluation, both the 2016 and 2017 test sets can be safely used for development and testing, respectively, as they were not included in training data.

8.4.2 EMEA

EMEA Data: We use a subset of the English–German portion of the European Medicines Agency (EMEA) parallel corpus[5] of documents focusing on medical products (Tiedemann, 2009). The corpus contains high levels of domain-specific terminology and repetition, making it appropriate for this task. For more detail about the corpus, see Section 4.1.2.

We select only those documents labeled as "humandocs" and then filter out very long and very short documents and those that contain only or primarily highly-repetitive dosage information. In particular, we removed all documents that contained in their names "Annex", "RQ", or "de2", as those tended to contain tables or very repetitive dosage information. Each document describes a new medication, meaning that new documents contain novel vocabulary. The medication name is typically repeated frequently within the document. Other novel vocabulary items include highly-specific medical terminology; these tend to appear fewer times within the document.

We divide the documents into training, development, and test sets such that all documents about a particular medication are in the same set. Thus most novel

[5] `http://opus.lingfil.uu.se/EMEA.php`

medication names in the development and test data will have been unobserved in the training data. We use four splits of the data: 500 document pairs ($375,000$ sentence pairs) for training a baseline EMEA-adapted model, 22 document pairs (5000 sentence pairs) as validation for that training, 5 document pairs (285 sentence pairs) for a small grid search over parameters, and 47 documents (2755 sentence pairs) for testing.

Test documents ranged in length from 48 lines to 95 lines. In general, the EMEA documents have a greater variation in length than this (with some having 1000 or more lines). For data with several hundred or more lines, considerable BLEU improvements have been documented with online adaptation and continued training (Servan, Crego, and Senellart, 2016). However, we seek to demonstrate that adaptation can be done with even shorter documents, and so focus this test set on documents with fewer than 100 lines.

The number of novel types per document in our test set ranged from 0 (no novel words; no dictionary adaptation) to 10 novel words. There are a total of 151 novel types (all documents combined) and 1129 novel tokens. Across the test set, 38.3% of lines contain at least one novel word. In some cases, up to 63.5% of the lines within a single document contain at least one novel word. Some novel word types occurred more than 30 times within a single document.

EMEA Baseline Model: The WMT model is trained on data which is significantly different from the EMEA data's medical domain. We see considerable differences in terms of vocabulary and sentence lengths. If we were to use the unadapted WMT model as our baseline, we might expect high gains from very small amounts of data due to the domain differences. Instead, in order to determine what marginal gains are possible in a real-life use scenario where a client already has access to a domain-specific model, we first adapt the WMT model on the EMEA train data so that it is familiar with the general style and vocabulary of the new dataset. Thus, improvements are attributable to document-specific adaptation rather than general domain adaptation.

We use the $375,000$ sentence pair training set, validating on the 5000 sentence pair development set, to perform continued training (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015). We use the same subword vocabulary and preprocessing pipeline as the WMT model. We limit sentence lengths to 50 tokens and train with a batch size of 80 over 15 epochs. We use a learning rate of 0.001 with the Adam optimizer (Kingma and Ba, 2014).

While training, external validation is done every 1000 batches and models are saved accordingly. We choose the model that gives the best validation score over the development set. Results are consistent with prior work: performance on the new domain peaks around the first few epochs and then tails off (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015).

The performance of the baseline WMT model on the EMEA development set gives a BLEU score of 18.2. Our best adapted model gives a BLEU of 51.5. With an increase of over 30 BLEU points, the adapted model is well-tuned to the EMEA corpus. We use this adapted model as the baseline for further document-level adaptation.

Source	This was a massive and , at the same time , very delicate operation .
Reference	Dies war eine massive und gleichzeitig sehr heikle Tätigkeit .
Baseline	Das war ein massiver und zugleich sehr delikater Betrieb .
Dict.-Adapt.	Das war ein massiver und zugleich sehr delikater Betrieb .
Sent.-Adapt.	Das war eine massive und zugleich sehr heikle Tätigkeit .
Dict.+Sent.	Dies war eine massive und zugleich sehr heikle Tätigkeit .

Table 8.2: Example of improvement from single-sentence adaptation and dictionary-and-single-sentence adaptation. The preferred translation of "delicate operation" ("heikle Tätigkeit") is observed in an earlier sentence in the document, and the model learns to reproduce it.

Source	Breast-feeding should be stopped while taking Siklos .
Reference	Das Stillen sollte während der Behandlung mit Siklos eingestellt werden .
Baseline	Während der Einnahme von Xenlos sollte abgestillt werden .
Dict.-Ad.	Während der Einnahme von Siklos sollte abgestillt werden .
Sent.-Ad.	Während der Behandlung mit Ivlos sollte abgestillt werden .
Dict.+Sent.	Während der Behandlung mit Siklos sollte abgestillt werden .

Table 8.3: Complementary nature of two approaches: single-sentence approach learns the preferred translation of "while taking" ("Während der Behandlung"), but mistranslates *Siklos* as *Ivlos*. Dictionary training produces *Siklos* correctly, but makes no other changes. Combined, the overall translation is improved, though it would still require post-editing for correctness. (Note that all translations use *abgestillt* rather than the noun *Stillen* for *breast-feeding*.)

8.5 Experiments

The two domains and their respective baseline models provide us two distinct scenarios to evaluate our methodology. Both simulate a relatively data-rich realistic setting in which translators have completed translations of in-domain data and continue to work on new documents (with novel terminology) within the same domain. Each domain provides its own challenges: the WMT data covers a wide range of topics and sources of news stories, while the EMEA data includes highly technical medical vocabulary, presented in fairly consistent ways. Due to the way our EMEA data splits were produced,[6] this in particular means that the new EMEA documents will likely contain novel vocabulary (such as names of medications and other specific terminology). Similarly, we expect news stories to cover new names, locations, and more as news breaks over time.

8.5.1 Single-Sentence Adaptation

For hyperparameter optimization, we did a complete grid search over a number of learning rates (0.1, 0.01, 0.001, 0.0001, 0.00001), training epochs (1, 5, 10, 20), and optimizers (*Adam*, *SGD*) on WMT data and a partial search on EMEA data. The batch size is set to 1, so setting the training epochs to n means that the model trains for n iterations on each individual sentence (before decoding the full remainder of the document and moving on to train on the next sentence). We use BLEU (Papineni et al., 2002) to measure the effect of adaptation. We found the optimum configurations (*optimizer, learning rate, epochs*) of (*SGD*, 0.01, 5) for EMEA[7] and (*SGD*, 0.1, 20) for WMT. The difference in optimum configurations can be partly attributed to the different domains of the two datasets. We note that the best EMEA configuration

[6] When there exists more than one document about a given medication, all of the documents about that particular medication are placed in the same split. For example, if one document about a medication is in the test set, all other documents about the drug will also be in the test set. This ensures that we do not train on documents about that medication during domain adaptation, and neither do we gain unfair additional knowledge about it during our parameter search. This also mimics a real-world scenario, in which we would expect that we have trained on existing medical documents and now need to continue adapting as new medications enter the market.

[7] During hyperparameter selection, document lengths were clipped to the first 60 lines.

matched the second-best WMT configuration.

8.5.2 Dictionary Training

For the EMEA dictionary experiments, we completed a grid search over number of epochs (1, 2, 5, 10) and learning rate (0.1, 0.5, 1.0) using SGD as the optimizer.[8] Finding consistent results, we ran a smaller grid search (epochs: 2 and 5 and learning rates 0.1, 0.5, and 1.0) over a development set of the first 20 documents from WMT 2016. Setting the learning rate and/or number of epochs too low resulted in minimal changes, while setting them too high resulted in pathological overfitting (loops of repeated tokens, etc.). Based on these initial experiments, we set a learning rate of 0.5 for both datasets, with 5 epochs for EMEA data and 2 epochs for WMT data. The parameters chosen were those that maximized BLEU score on the appropriate development set.

8.5.3 Lexically Constrained Decoding

We compare our dictionary training approach against an approach that uses the same dictionaries and enforces a lexical constraint: if one of the dictionary entries appears in the source, its translation (acquired as described in Section 8.3.2) must appear in the translated output. We do this using the grid beam search approach

[8] We also considered lower learning rates (0.01, 0.001, 0.0001), but found that they did not result in much, if any, change to the model.

Model	**BLEU**	**Nov. Acc.**
EMEA-Adapt. Baseline	51.1	39.9%
Single-Sent. Adapt.	52.8	62.3%
Lex. Const. Decoding	50.4	86.5%
Dictionary Training	53.3	87.9%
Dict. + Single-Sent.	53.8	89.1%

Table 8.4: Results of baseline and dictionary training across the full set of EMEA test documents. Accuracy is computed for novel words only.

described in Hokamp and Liu (2017). Rather than adapting the underlying machine translation model, this approach constrains the search space to translations containing specified subsequences (in this case, the byte pair encoded representations of the translation of any words from the dictionary which appears in the source sentence). We use the publicly released implementation for Nematus, with a beam size of 12.

8.5.4 Single-Sentence Adaptation with Dictionary Training

Here we combine the approaches: for every document, we first do dictionary training. Using that as the starting point, we perform single-sentence adaptation. We use the best hyperparameters obtained from the grid search for the individual methods.

Model	BLEU	Nov. Acc.
WMT Baseline	25.1	48.9%
Single-Sent. Adapt.	26.7	58.4%
Lex. Const. Decoding	25.0	76.9%
Dictionary Training	25.1	71.7%
Dict. + Single-Sent.	26.9	72.2%

Table 8.5: Results of baseline and dictionary training across the full set of WMT test documents. Accuracy is computed for novel words only.

8.6 Results and Analysis

We evaluate on two metrics: BLEU and novel word accuracy. First, we compute BLEU over the full set of test documents and compare against the baseline translations. Across both domains, single-sentence adaptation provides consistent improvements in BLEU score (1.6 BLEU points on WMT data and 1.7 BLEU points on EMEA data). The dictionary training approach has more varied results. We see no clear improvement on the WMT data, but training on these small dictionaries does not *hurt* BLEU score overall. However, for the EMEA data, dictionary training produces a 2.2 BLEU point improvement. This gain can be primarily attributed to producing correct translations of the novel vocabulary, which can make a large difference in *n*-gram matches.[9]

The lexically constrained decoding approach results in a decrease in BLEU score on both domains. In their work on an alternate constrained decoding algorithm, Post and Vilar (2018) describe a phenomenon that they call *reference aversion*, where

[9] Consider the case of the baseline translation *Was ist AFluntis ?* and the (correct) dictionary-adapted version *Was ist Aflunov ?* – the former contains no 4-gram matches.

forcing the output to contain a word or words from the reference increases BLEU score while decreasing model score. They also note that another effect of reference aversion is that their beam often contains weak competing hypotheses, arguing that when the decoder is forced into generating low-probability sequences of tokens, it may revert to generating high probability target language output (like a language model), without clear connection to the input. What we observe here is closer to the latter: both the model score and the BLEU score are decreasing, particularly for EMEA data. We suspect that the model assigns low probability to the novel vocabulary items, especially drug names (which are quite different from standard German or English words), and subsequently suffers when forced to generate them. We show this to be the case in Chapter 9, and examine it in more detail there.

Combining both dictionary training and single-sentence adaptation results in modest improvements (0.2 on WMT and 0.5 on EMEA) over the best single approach for each domain. Full results are shown in Tables 8.4 and 8.5. The combined approach produces BLEU score improvements over the baseline for 79.2% of the WMT documents and 83.0% of the EMEA documents.

Figure 8.1 shows how single-sentence adaptation improves translation quality (as measured by BLEU score) over the course of adapting to EMEA documents. We examine this as follows: for each sentence (indexed by n) in a document, we take the model trained on the first n sentences and decode all remaining sentences in the document (all sentences s_i in the document, where $i > n$), compute the BLEU

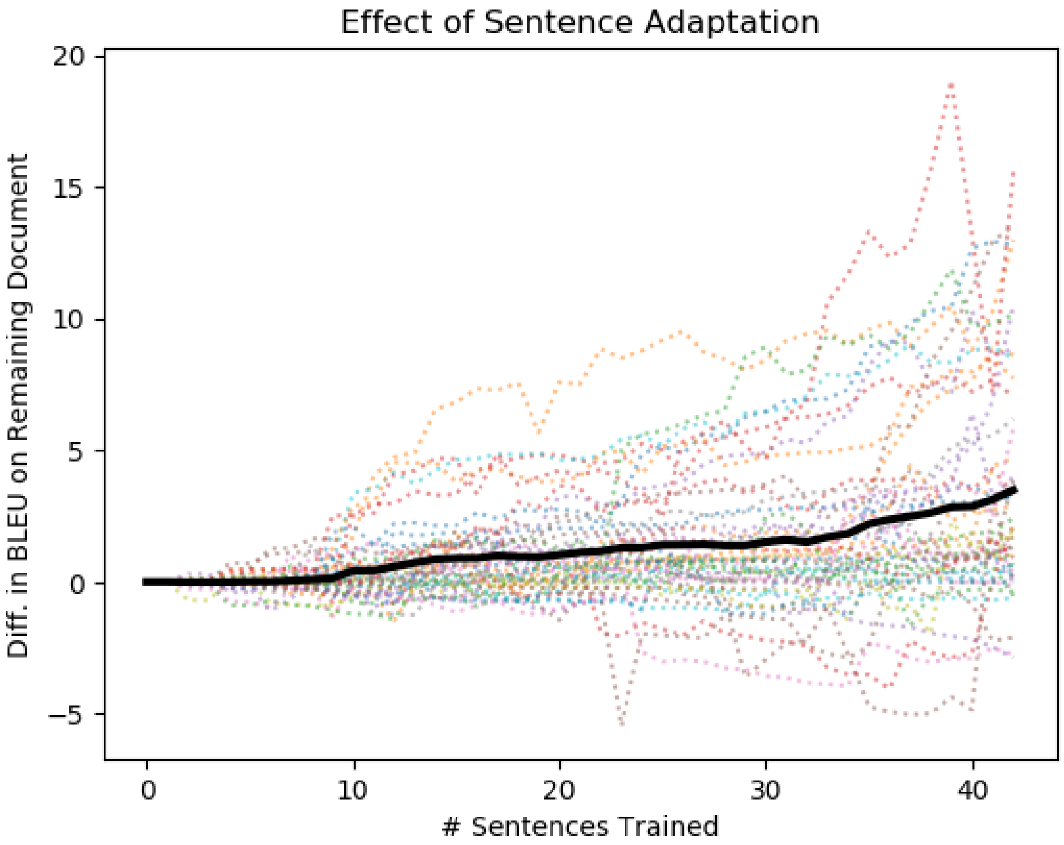


Figure 8.1: The X-axis shows the number of sentences to which the model has been adapted. The Y-axis shows the difference in BLEU score between this adapted model and the baseline on the document's remaining lines (higher Y values indicate greater improvement over the baseline). Dotted lines represent individual documents; the average trend is shown in bold. This plot displays EMEA results.

score over that document subset, and compare it to the BLEU score of the baseline model (computed over the same subset of sentences). We then calculate and plot the difference in BLEU score (baseline score subtracted from adapted score). The overall trend is an increasing improvement in the BLEU score difference of the remaining lines of the document, starting from around the 10 sentence mark.

We observe qualitative results that suggest that single-sentence adaptation is performing as expected, learning document- or translator-specific translations. For example, as shown in Table 8.2 the baseline WMT system initially translates the English bigram "delicate operation" as "delikater Betrieb" while the reference translation prefers "heikle Tätigkeit" as the translation. In the next sentence in which "delicate operation" is observed, the sentence-adapted model successfully translates it as "heikle Tätigkeit" instead. Table 8.3 shows another example in which the two approaches combine to produce improvements: the single-sentence adaptation produces the desired translation "Behandlung" for the word "taking", while the dictionary adaptation correctly copies the medication name ("Siklos"). Together they successfully produce both of these corrections. Even with this adaptation, the translation may still contain non-adaptive errors (the reference uses the noun "Stillen" for "breast-feeding", while the machine translation output uses an inflected form of the verb "abgestillen", meaning "to wean" or "to stop breast-feeding").

We also compute accuracy for the translations of novel words.[10] To compute accuracy, we first run a trained fast-align model over the byte pair encoded source and the byte pair encoded reference. We use this alignment to map full tokens from the source to full tokens in the reference (as was done for producing the dictionaries). We then align the source sentence and the machine translation output the same way. For each instance of a novel word, we score its aligned machine translated token as correct if it matches the aligned reference token. The dictionary training approach shows,

[10] Since the publication of this work, Simianer, Wuebker, and DeNero (2019) have also proposed approaches to measuring zero-shot and one-shot adaptation effects for lexical items.

as expected, a major jump in translation accuracy. The single-sentence adaptation approach shows results that fall between the baseline and the dictionary approach. Lexically constrained decoding underperforms dictionary training on EMEA data (in part because it sometimes produces medication names that are concatenated with other subwords, or produces the medication name more times than required), while it outperforms other methods on the WMT data (at a cost to the overall BLEU score, whereas all other methods produce improvements in BLEU). Table 8.4 shows that EMEA improves from a baseline accuracy of 39.9% to an accuracy of 87.9% after dictionary training, and Table 8.5 shows a slightly smaller jump from 48.9% to 71.7% for WMT. Both show slight improvements after combining single-sentence adaptation and dictionary training.

With this increase in accuracy comes an increase in consistency of translating the novel words. In the baseline EMEA-adapted model, the average type-token ratio[11] for translations of novel words that occur at least 3 times (in the source text) is 0.29. With dictionary adaptation, this drops to 0.14 – lower than the reference type-token ration of 0.16 – meaning that the new model produces the exact translation from the dictionary even when a variant (e.g., different case ending) may be appropriate. As we use only one translation per novel source token in the dictionaries used for training, the model overfits slightly. This issue could potentially be alleviated by training on multiple translation options, at the risk of introducing errors from incorrect

[11] The number of different machine translation outputs for the source type, divided by the number of times that source type appears.

Model	**WMT**		**EMEA**	
	Copy	*Trans.*	*Copy*	*Trans.*
Baseline	80.8%	11.3%	41.9%	28.4%
Single-Sent. Adapt.	87.9%	23.6%	67.2%	32.7%
Dictionary Adapt.	92.5%	47.3%	92.5%	60.5%
Dict. + Single-Sent.	94.6%	45.8%	92.9%	66.7%

Table 8.6: Novel word accuracy divided into tokens to be copied (*Copy*) vs. translated (*Trans.*).

alignments.

We perform more detailed analysis across two kinds of novel words: those which should simply be copied from source to target (e.g., medication names) and those which must be translated. Table 8.6 shows results for the baseline and our approaches. WMT data is almost evenly split between these: 46.8% of novel types (54.1% of tokens) must by copied, while EMEA data is skewed towards words that should be copied, with 51.7% of novel types (85.7% of tokens). On WMT data, baseline accuracy of terms to be copied is already quite high, but accuracy of terms to be translated is very low. The EMEA baseline has a much harder time with tokens that should be copied, but does better on non-copied terms. The analysis in Section 7.3 provides additional insight into why this may be. The set of WMT novel words contains many names of people or places, as well as some morphological variants of known words. We observed that context can influence copying rate, and in particular that certain contexts (including those in which names often appear) tend to result in more copying of novel vocabulary. Some of these names do appear near job titles and other such potentially copy-prone contexts, which may increase their successful copying. On the other hand, novel words in EMEA tend to be a mix of medication names, which tend to contain character sequences not frequent in either source or target language, or highly-specialized medical terminology. The medication names in particular are quite morphologically and orthographically distinct from either German or English. We observe that for many of the medication names, it takes 10 or more instances of the name being observed for the single-sentence adaptation approach alone to successfully learn to copy the word (if ever). Though there remains a gap between novel word accuracy on tokens that should be copied and those that should be translated, our approaches demonstrate improvements for *both* types of novel words.

A concern with training on a dictionary as bitext is that the model may overfit to the sentence length; we do not find that to be the case here, as the difference between the full hypothesis lengths is $48,641$ tokens for the EMEA-adapted data compared to $48,627$ for the dictionary-trained models. However, this is dependent on choosing the correct learning rate and number of epochs. Similarly, there's a potential concern that single-sentence training on the previous sentence may cause some type of overfitting (memorization of the sentence, etc.). We do not observe that to be the case either.

8.7 Conclusions

We propose two approaches to document-level adaptation of NMT systems (single-sentence adaptation, dictionary training) and their combination, which can be effectively used to improve performance, both in terms of BLEU score and in the translation of novel words. Both approaches have minimal training data requirements, can be effectively applied with an existing NMT architecture, and show considerable improvements even for short documents.

Part V

Applications to CAT

Chapter 9

Applications of Adaptation to CAT

9.1 Introduction

This section of the dissertation brings together the two main contributions of this thesis: combining neural interactive translation prediction and fine-grained adaptation of neural machine translation systems. We perform this work in simulation, evaluating the potential for success using several metrics: BLEU score (as examined in prior work), word prediction accuracy (WPA), and examining the probability assigned by the model to various tokens of interest. In examining adaptation in the interactive translation prediction setting, we observe that the model probability assigned to novel words does increase and that this results in higher word prediction accuracy. This also ties back in to Chapter 5; we know that interactive translation prediction recovers more quickly from making an error when the correct token received relatively high model probability. Increasing the model probability of these novel words serves to increase the chance that they are correctly generated, but also has the potential to positively influence the word prediction accuracy of the remainder of the sentence.

9.2 Experimental Setup and Evaluation

We reuse the WMT and EMEA English–German models and data from Chapter 8. These consist of a publicly released model from the University of Edinburgh's submission to the WMT news translation task as a baseline for WMT (Sennrich, Haddow, and Birch, 2016a) and the same model domain-adapted to EMEA data for use as a baseline for EMEA. In particular, we focus on the baseline model, the dictionary adapted model, and the combined dictionary and single-sentence adaptation model. The combined model is the best performing across both datasets, while examining the dictionary adapted model in this setting allows us to closely examine the effects on the translation of novel words. Instead of standard decoding, we simulate interactive translation prediction (beam size 1) as described in Chapter 5. In our evaluation, we consider novel words; again, these match those from Chapter 8.

9.2.1 Word Prediction Accuracy

One way we can measure improved performance (in addition to BLEU score) is to consider the effects of adaptation on word prediction accuracy. We expect that a system that has successfully performed adaptation should demonstrate this improvement through an increase in word prediction accuracy. For example, after performing dictionary training, we would expect the system to be more likely to output target language dictionary items in sentences containing source side dictionary items, increasing word prediction accuracy just as it increased BLEU. Similarly, in single-sentence adaptation (or the combination of the two), we would expect to be more likely to produce the translator's preferred phrases, thus increasing word prediction accuracy.

9.2.2 Novel Word Prediction Accuracy

We measure word prediction accuracy restricted only to words of interest, such as the novel words that were trained on for dictionary training (which we call novel word prediction accuracy). That is, whenever a target language word from the dictionary used for dictionary training appears in the reference output (which we are using to simulate a human translator), we check whether the interactive translation prediction system produced it, and calculate this targeted word prediction accuracy by dividing the number of times the system correctly generated such a word by the total number of such words in the reference. This differs from our earlier analysis, which used word alignments on free translations. In the case of interactive translation prediction, we can dispense with the need for alignments because the reference itself indicates precisely where the target words should appear.[1]

9.2.3 Model Probabilities

Our second approach to examining the generation of tokens of interest is to examine the probability assigned to these novel words of interest. We expect to see this increase with adaptation. While we could measure this in free translation output, measuring it with a constrained prefix and forced decoding in the interactive translation prediction setting provides a more consistent context for measuring the probability assigned to these novel tokens. We plot the average negative log probability (averaged by type) to examine the changes that the adaptation methods produce in individual novel words. We also consider the average negative log probability as averaged across three sets: the full test data, the novel words only, and the complement of the novel words only. By comparing each of these sets across the different adaptation scenarios, we can see the effect of adaptation in general, on the novel words, and on the remainder of the words, which helps to explain the performance improvements observed in BLEU score and word prediction accuracy.

[1] We can connect interactive translation prediction and constrained decoding: checking the word prediction accuracy here is similar to checking the word prediction accuracy following a constraint, where the constraint includes the beginning of sentence token. Similarly, interactive translation prediction in simulation has a strong connection to forced decoding, the only difference being the generation of predicted tokens at each timestep (to compare to the reference), rather than decoding the reference only.

9.3 EMEA Experiments

9.3.1 WPA and NWPA

In Table 9.1 we show results for word prediction accuracy, novel word prediction accuracy, letter prediction accuracy, and novel letter prediction accuracy on the EMEA data. We saw in Chapter 8 that the EMEA baseline begins at quite a high BLEU score (51.1). This corresponds to a high baseline word prediction accuracy (74.5%). Dictionary adaptation increases this more than one percentage point (to 75.6%), while single-sentence training provides a slight improvement above this (to 75.9%). The letter prediction accuracy improvements are smaller but follow the same trend.

When we restrict to novel words only, the scores and improvements track almost identically with the novel word accuracy. The novel word accuracy starts low, at 37.2%, rising to 87.0% with dictionary training, and increasing slightly to 89.1% with combined dictionary and single-sentence adaptation. The novel letter prediction accuracy follows a similar trend. The fact that novel word prediction accuracy under interactive translation prediction so closely tracks the novel word accuracy in free translation is not in itself surprising; the underlying model is the same, it is merely the generation process that differs.

In Table 9.2 we compute word and letter prediction accuracy restricted to the words immediately following instances of novel words in the simulated interactive translation prediction output. We know that errors in interactive translation prediction can

Model	BLEU	N. Acc.	WPA	N. WPA	LPA	N. LPA
EMEA-Ad. Base.	51.1	39.9%	74.5%	37.2%	93.0%	79.0%
Dict. Training	53.3	87.9%	75.6%	87.0%	93.5%	95.6%
Combined	53.8	89.1%	75.9%	89.1%	93.5%	96.0%

Table 9.1: Results of EMEA-adapted baseline and dictionary training across the full set of EMEA test documents. BLEU and novel word accuracy (Nov. Acc.) are computed with standard decoding (from Table 8.4), while WPA, Novel Word WPA, and Letter Prediction Accuracy (LPA) are computed with interactive translation prediction (using the reference for simulation).

set off subsequent (though often short) cascades of other errors, especially when the translator's desired translation was assigned low probability by the model. As such, we might expect that word prediction accuracy would suffer for the words immediately following novel words (particularly for the baseline system which struggles when translating them). However, we don't find that to be the case for EMEA data: the word prediction accuracy computed over words following novel words is in some cases slightly higher than the overall word prediction accuracy. A manual examination of this suggests that it may partly be a quirk of the structure of EMEA documents. In particular, for a given medication (which we'll represent as "MEDICATION") they often begin with "What is [MEDICATION] ?" (translated as "Was ist [MEDICATION] ?"), and then state "[MEDICATION] is ..." (translated as "[MEDICATION] ist ..."). So, perhaps even in the case where the system has failed to correctly translate the medication name, it is still able to compensate through the sheer repetitiveness of this particular data; we do not observe the same trend with WMT data. This effect could also be related to the high proportion of novel words in the training data lowering the

Model	Following Word WPA	Following Word LPA
Baseline	75.7%	93.2%
Dict. Training	75.4%	93.3%
Combined	77.9%	93.9%

Table 9.2: Results of baseline, dictionary training, and combined dictionary with single-sentence adaptation across the full set of EMEA test documents. WPA and LPA are computed over each of the first tokens *following* a novel token.

overall word prediction accuracy.

9.3.2 Model Probabilities

We now examine the probability assigned to the words in the reference by the model during interactive translation prediction. We begin with the novel words. The dictionary adaptation approach in particular was designed to improve translation of these novel words, and we show that this does increase the probability that the model assigns to those words (with minimal negative consequences for other words). Figure 9.1 shows the change in negative log probability averaged over each novel type after dictionary adaptation. In the figure, types are sorted by their average negative log probability under the baseline model; it is clear to see that in almost all cases, the adapted models show a decrease in negative log probability, indicating that the adapted models now assign higher probability to the novel words than the baseline model did. The values for the combined dictionary and single-sentence adapted models had near identical performance.

Table 9.3 examines model average negative log probabilities across all tokens, just

Model	All Tokens	Novel Tokens	Other Tokens
EMEA-Adapted Base.	1.78	3.45	1.68
Dictionary Training	1.64	0.82	1.69
Combined	1.59	0.74	1.65

Table 9.3: Average negative log probability of all tokens, novel tokens, and all other tokens (non-novel tokens). Lower values indicate improvement.

the novel tokens, and other tokens (the complement of the novel tokens). This allows us to tease apart the impact of the different adaptation techniques on different words. The most drastic improvement is from the baseline to dictionary training as measured over the novel words, a relative change of 76.2%. While there is improvement for all tokens and for novel words, it comes at a cost to the non-novel tokens. However, this cost is quite small in comparison, a relative change of only 0.6%. The combined dictionary and single-sentence adaptation shows improvements across all subsets of the tokens (over both the baseline and the dictionary training).

9.4 WMT Experiments

9.4.1 WPA and NWPA

Table 9.4 shows word prediction accuracy results for the 2017 WMT news test data in English–German. In contrast to the highly-repetitive EMEA data, WMT data consists of a variety of news stories, covering a range of topics. The baseline begins with a lower overall BLEU score (25.1) and corresponding word prediction accuracy (43.4%)

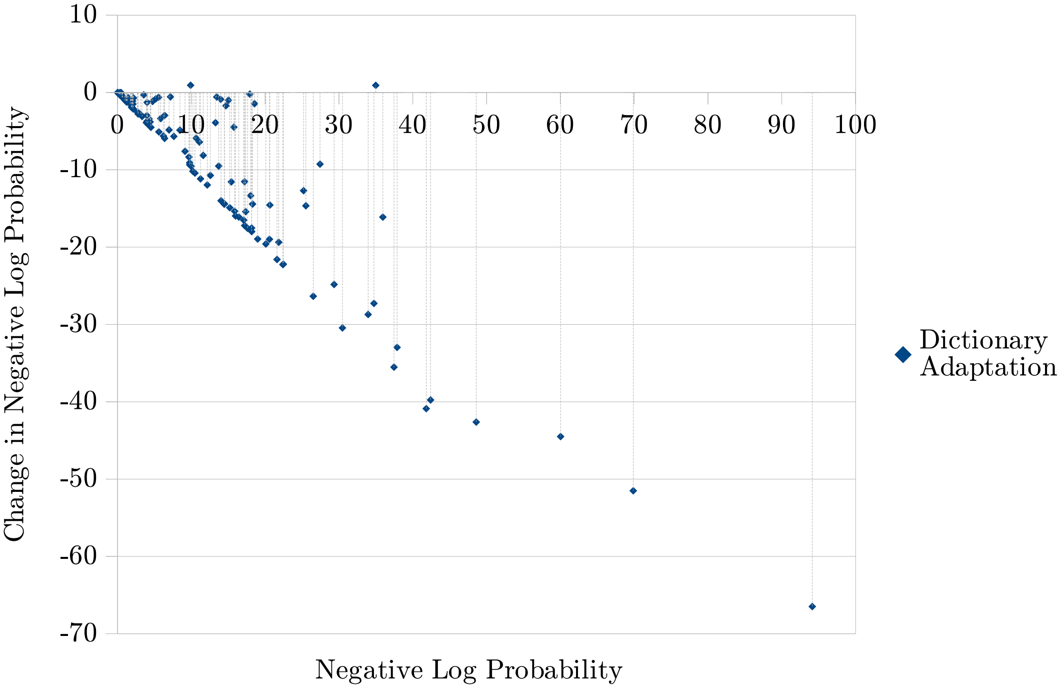


Figure 9.1: Change in average negative log probability assigned to novel words in the EMEA dataset (averages computed over types, with each type represented by a point in the plot), with horizontal axis sorted by baseline average negative log probability. Each point shows the change in average negative log likelihood after dictionary adaptation (combined dictionary and single-sentence adaptation values are nearly identical). All points below the horizontal axis show improvement (decrease in average negative log probability).

Model	**BLEU**	**N. Acc.**	**WPA**	**N. WPA**	**LPA**	**N. LPA**
Baseline	25.1	48.9%	43.4%	27.1%	78.0%	68.9%
Dict. Training	25.1	71.7%	56.3%	71.6%	86.6%	92.5%
Combined	26.9	72.2%	59.2%	71.8%	88.3%	93.4%

Table 9.4: Results of baseline, dictionary training, and combined dictionary with single-sentence adaptation across the full set of WMT test documents. BLEU and novel word accuracy (Nov. Acc.) are computed with standard decoding (from Table 8.5), while WPA, Novel Word WPA, and Letter Prediction Accuracy (LPA) are computed with interactive translation prediction (using the reference for simulation).

than the EMEA dataset. While we saw no change in BLEU score through dictionary adaptation, we did see an improvement in novel word translation accuracy in Chapter 8 (repeated here in Table 9.4). This corresponds to an (even greater) improvement in novel word prediction accuracy (jumping from 27.1% using the baseline model to 71.6% with the dictionary adaptation model). We also see an increase in word prediction accuracy with dictionary training, which may be primarily attributable to the large increase in novel word prediction accuracy. Moving from dictionary training alone to the combined dictionary and single-sentence adaptation, we again see increases in both word prediction accuracy and novel word prediction accuracy. This time, however, the magnitude of the change is different; the novel word prediction accuracy only increases by 0.2 percentage points (0.3% relative improvement), while the overall word prediction accuracy increases 2.9 percentage points (5.2% relative), suggesting that the single-sentence adaptation is improving translation of other words in the document or making other stylistic improvements. Letter prediction accuracy follows similar trends.

Model	Following Word WPA	Following Word LPA
Baseline	41.5%	76.0%
Dict. Training	58.5%	87.2%
Combined	58.1%	87.6%

Table 9.5: Results of baseline, dictionary training, and combined dictionary with single-sentence adaptation across the full set of WMT test documents. WPA and LPA are computed over each of the first tokens *following* a novel token.

In Table 9.5 we show results for the word prediction accuracy restricted to only the word immediately following a novel word. For these words, we see a drop in word prediction accuracy with the baseline model, then an increase in the adapted models. This does suggest that, for this dataset, improving the translation of novel words has a positive impact on the translation of subsequent tokens, as we might expect from evidence in neural interactive translation prediction simulations.

9.4.2 Model Probabilities

Table 9.6 shows average negative log probabilities of all tokens in the WMT test set, as well as those same tokens separated into novel tokens (which were trained on in the dictionary adaptation approach) and all other tokens. We find that dictionary training decreases average negative log probabilities across all sets of words examined, most strongly for the novel tokens (which were the focus of that training), which saw a 77.1% relative drop. However, in contrast to what we observed on the EMEA data, dictionary training also resulted in a noticeable drop in the average negative log probability of the non-novel tokens. This difference could be related to the very

Model	**All Tokens**	**Novel Tokens**	**Other Tokens**
Baseline	5.55	6.06	5.54
Dictionary Training	3.31	1.39	3.35
Combined	3.06	1.33	3.10

Table 9.6: Average negative log probability of all tokens, novel tokens, and all other tokens (non-novel tokens) for WMT data.

different types of novel words that exist in the two datasets. In EMEA, most novel words are names of medications, while in WMT novel words span a wider range of categories: numbers, proper nouns, compound nouns, previously unobserved inflections of adjectives and verbs, and so on. Some of these may have subword overlap with other words in the document, something which is less likely to be the case in EMEA data where the novel drug names are quite different from English or German text. For example, we see that when training on *2015-16* (with BPE segments *20@@ 15-@@ 16* – in the context of "in der Saison 2015-16" or "in [the] 2015-16 [season]"), the probability of *20@@* in the context *20@@ 13-@@ 14* also improves. Improving the probability assigned to novel tokens could also be improving the probability of the subsequent tokens.

9.5 Conclusion

In this section, we examine the combination of neural interactive translation prediction and fine-grained adaptation in simulation experiments. We show that improvements from both dictionary training and combined dictionary training with single-sentence adaptation do correspond to improvements in interactive translation prediction metrics. These benefits do extend beyond just improvements to novel token translation, showing their benefits for two very different types of datasets. This result has positive implications for the potential usefulness of adaptation in an interactive translation prediction setting.

Part VI

Conclusion

Chapter 10

Conclusions

10.1 Conclusions

In this dissertation, we have described work in the areas of machine translation and computer aided translation. In particular, we have shown that neural interactive translation prediction outperforms phrase-based statistical machine translation approaches, even when the underlying machine translation systems are of similar quality. By performing a user study with professional translators, we have demonstrated the feasibility of neural interactive translation prediction as a competitive alternative to post-editing, and have observed primarily positive translator reactions to the tool. We have provided an analysis of challenges facing machine translation systems in terms of the translation of rare words as well as translation consistency, topics that are highly relevant to computer aided translation work. We show that dictionary and single-sentence adaptation can successfully perform extremely fine-grained adaptation, improving already-strong neural machine translation systems. In simulation, we apply those adaptation tools to a computer aided translation task and demonstrate improvements on metrics that may correlate with translator efficiency and satisfaction.

Machine translation is playing an increasingly large role in the world – free online machine translation services are used to translate hundreds of billions of words every day.[1] However, for companies, governments, and individuals interested in localization or publishing content in multiple languages, the quality of machine translation output is often insufficient. For this, they turn to translators and language service providers, who often use computer aided translation tools to produce high-quality translations efficiently. In an increasingly connected world where over 7000 languages are spoken (and for only a small portion of which machine translation technology currently exists),[2] translation is a growing industry with a major impact in a wide range of areas, from government to technology to medicine to communication and beyond.[3] The ability to access information in one's own language also has a more qualitative impact on individuals. The work in this dissertation seeks to make progress towards both improved machine translation quality and improved experiences for translators using computer aided translation tools in their daily lives.

[1] https://www.blog.google/products/translate/ten-years-of-google-translate/

[2] https://www.ethnologue.com/guides/how-many-languages/

[3] To measure this impact financially, the global market of "language services and technology," including translation was estimated at $46.5 billion in 2018, and is expected to continue growing, according to the Globalization and Localization Association. https://www.gala-global.org/industry/industry-facts-and-data/

10.2 Future Work

There remain, as ever, a number of promising directions for future work in this area. As machine translation systems continue to improve, the question of how best to form a human and machine partnership continues to evolve. We face questions about whether the high quality of neural machine translation may produce errors that are harder to catch or more inconsistent than those made by phrase-based statistical machine translation systems. We must also consider what it means for translator interaction modes: is post-editing a near-perfect translation faster than using interactive translation prediction? Is there a threshold of quality that makes one tool better than the other?

10.2.1 Human Variability

If experience is any guide, the answer to many questions about the usefulness of any given computer aided translation tool is that it will vary greatly by translator. One of the areas ripest for exploration is to examine which tools work best for individual translators on the basis of their strengths, areas of expertise, and personal preferences. In Chapter 5, we observed possible correlation between post-editing experience and higher word prediction accuracy scores for interactive translation prediction, but our small sample size made it difficult to draw concrete conclusions. Future study would benefit from larger groups of study participants (though this also comes with greater costs) in order to be able to confidently measure correlations between metrics of tool usefulness and translator-specific variables.

There are a number of translator-specific variables that would be worth exploring in large-scale studies of interactive translation prediction (or other CAT tools). We examined experience, education, and certification, particularly with respect to post-editing background. We did not measure typing speed, but it is very plausible that this might play a role; slower typists may benefit from word and sentence completion (or even just from post-editing text), while faster typists might see different benefits. Other factors such as a translator's receptive and productive vocabulary may also come into play. A translator with a strong receptive vocabulary but weaker productive vocabulary might benefit from being presented with translation options, which they can then verify. On the other hand, a system that incorrectly primes a translator who has a weaker vocabulary has the potential to backfire by leading them astray.

The choice of participants for a user study is also important; many studies choose to use students (of translation or in any field), while others use professional translators. The selection of participants should take into account the intended audience for the tool. If the tool is being built for professional translators, evaluation by non-professionals may not be representative of the results that would be obtained with professional translators. Moorkens and O'Brien (2015) discuss these and other differences between student and professional translators acting as user study participants. If the tool is being built with the goal of use by a wide range of professional and amateur translators, the study participants should reflect that. The types of tools that are most useful to each of these various groups are likely to be both task-specific and translator-specific, and translators' experience with and preference for certain workflows is likely to have an impact on their perception of a CAT tool, their interest in trying new tools, and the tool's usefulness for them as part of a new or existing workflow.

One major challenge in evaluation is the tradeoff between examining the impact of a particular computer aided translation technology in isolation as opposed to having a more realistic setting. Teixeira and O'Brien (2017) perform eyetracking, screen recording, key logging, and interviews with translators to observe their translation processes, a complicated data collection process in itself. They find that translators often switch between different tools in the course of their work, sometimes even using tools other than the computer, such as their phones. In our work, we perform key and mouse click logging in a simple computer aided translation interface. Each of these approaches has its own strengths: the Teixeira and O'Brien (2017) approach can help draw conclusions about translator workflows through detailed analysis of the tools a translator uses in their typical process, while our approach allows for much more automatic analysis of productivity with a particular tool in a constrained setting. Recent work (Daems and Macken, 2019) has compared statistical and neural interactive prediction in the commercial interface Lilt, examining the approaches in a tool with more of the industry-standard bells and whistles enabled (in the case of this study, though, Lilt was a new tool for most of the participants).

While it may be challenging to run sufficiently large-scale user studies to untangle the influences of human variables and specific tools (or improvements to tools), large-scale user surveys like Moorkens and O'Brien (2017) may provide some guidance for high-impact areas of future research and development. Features desired by translators interviewed in that work include dynamically updating machine translation systems to improve recurring errors (as we examined in the fine-grained adaptation work in Chapter 8), interactive machine translation (as we explored in Chapters 5 and 6), and confidence scores (which we will discuss in Section 10.2.3). As suggested in the description of a *translator's amanuensis* in Kay (1980), translators today would still prefer that these interfaces be customizable, allowing them to pick and choose which features they would like to use.

10.2.2 CAT-Specific Architectures

Much of the core of machine translation research is aimed at improving translation in an assimilative context (i.e., translating webpages for a reader to access in their preferred language), and architectures are designed with that in mind. The work in this dissertation has taken existing machine translation architectures and used them, with occasional modifications, for the task of computer aided translation. An alternative approach would be to build machine translation architectures for the specific goal of use in computer aided translation. What could a CAT-specific machine translation architecture look like? What factors should a CAT-specific architecture take into account?

Tools like CASMACAT offer a number of different interaction modes for translators, including interactive translation prediction, post-editing, translation by selecting words from a visualization of a search graph, and so on. There are also a number of additional optional visualizations, like alignment, source coverage based on the text translated so far, and confidence. Some of these are intimately tied to phrase-based statistical machine translation approaches, where information about alignment and coverage are produced as byproducts of the translation process. For example, one may wish to either see which source words have already been translated or to see what source words are being translated by the system as a particular target word. In the phrase-based statistical approach, this is relatively simple: the alignment delivers it. As shown in Koehn and Knowles (2017), though, this may be more challenging in neural machine translation systems because attention is not always alignment and thus cannot be relied upon as such (and attention in architectures like transformer (Vaswani et al., 2017) may be even less interpretable). Recent work in Ding, Xu, and Koehn (2019), however, has proposed a potential solution to this, through the use of saliency, which can be applied to a range of architectures. Alternatives would include training architectures to explicitly model alignment as well as translation.

Architectures designed with CAT-related goals in mind may also contribute back to the core machine translation literature. In Chapter 8, we examined ways to adapt machine translation models to perform better on documents by adapting to document-specific vocabulary or by performing sentence-level adaptation throughout the process of translating a document. Document-level machine translation and evaluation has seen growing interest in recent years (Läubli, Sennrich, and Volk, 2018; Maruf and Haffari, 2018; Barrault et al., 2019), with recent success at WMT (Junczys-Dowmunt, 2019). There are a number of potential approaches to document-level neural machine translation, including conditioning translation on one or more previous sentences or, as in Junczys-Dowmunt (2019), treating documents as very long sequences with sentence separating tokens. Both approaches have the potential to be more computationally expensive than standard decoding, especially if multiple passes over the document are required. In the computer aided translation setting, one would also want to be able to take into account any corrections made by a translator to early portions of the document in later portions (potentially by a sentence-adapation approach or a constrained decoding approach). Approaches that condition on previous sentences (and their translations) may appear to be more easily suited to this, as one could simply condition on post-edited or interactively translated output from earlier sentences in the document. With the increasingly high quality of neural machine translation output, producing document-level coherence is an exciting challenge, one that will have benefits for a range of machine translation use cases, including but not limited to computer aided translation.

Due to the need for and challenges of human evaluation of CAT tools, it will often take time for certain CAT tools to adapt to novel architectures. While any machine translation architecture can be used for post-editing, human translators will need to learn how to adjust to the kinds of errors the systems make. Interactive translation prediction and other types of CAT tools tend to be intimately tied to a particular architecture, so changing architectures may require new approaches to interactive translation prediction, as has been the case for neural machine translation. Torregrosa, Pérez-Ortiz, and Forcada (2017) propose black-box approaches to interactive translation prediction to resolve this, but their approach involves translating segments of sentences, which is a task that may be quite different from the training data and may result in differing performance between different paradigms of systems. As an alternative to building CAT-specific architectures, one could also seek to build architecture-agnostic CAT tools, though the risk is that new machine translation paradigms may fail to adhere to the assumptions with which the CAT tools were built.

10.2.3 Trust, Confidence, and Mistakes

When neural machine translation models make mistakes, they typically do so in ways that are quite different from phrase-based statistical machine translation systems. For example, they may start to generate fluent text that has no apparent connection to the source (what is sometimes described as "switching to language modeling mode") or they may replace words with grammatically reasonable but semantically incorrect substitutions. While we have some understanding of certain aspects of performance, such as why they typically produce output of the correct length (Shi, Knight, and Yuret, 2016), they can be somewhat opaque otherwise. The unique paradigm of mistakes and the challenges associated with explaining them may result in user mistrust in a neural machine translation system. On the other hand, the high rates of fluency may lure readers or translators into a false sense of security, causing them to miss errors. Martindale and Carpuat (2018) provide an overview of how readers of machine translation output may react to fluency or adequacy errors, finding that they react more negatively to fluency errors.

When translators use translation memories in a computer aided translation tool, they are typically presented along with a "fuzzy match score" which lets the translator know how close they should expect the translation to be to a correct translation. Many translators would also like to see a confidence score for machine translation output that they are expected to post-edit (Moorkens and O'Brien, 2017), and this could also be implemented for individual words in interactive translation prediction. There is a rich body of literature on quality and confidence estimation for machine translation (Callison-Burch et al., 2012), including for interactive translation prediction (Gandrabur and Foster, 2003; González-Rubio, Ortiz-Martínez, and Casacuberta, 2010). We examined simple approaches to word-level confidence for neural interactive translation prediction in Knowles and Koehn (2018b) and found that the score assigned to the current token by the neural machine translation model could be used (via thresholding or in a regression model) for confidence estimation. It is by no means perfect, though, and sometimes the model assigned very high probability to completely incorrect tokens. Building machine translation systems that are penalized more heavily for overconfidence in incorrect translations, or that are rewarded for emitting some information to indicate when they may need human oversight could potentially help to ameliorate this.

For translators accustomed to working with translation memories or phrase-based statistical machine translation output, the different types of errors made by neural systems may require a period of adjustment or changes to their workflow to ensure that they catch adequacy errors in otherwise fluent translations. In general, producing more interpretable models (or finding ways to make existing models more interpretable, such as in Ding, Xu, and Koehn (2019), or through better indications of confidence) could enhance user trust and understanding. It comes with a risk, though: poor estimates of confidence could cause users to miss errors or to waste time examining correct tokens, either of which could also result in either an overabundance of trust or a loss of trust in the system.

Bibliography

Alabau, Vicent, Alberto Sanchis, and Francisco Casacuberta (2011). "Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Techologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 389–394. URL: https://www.aclweb.org/anthology/P11-2068.

Alabau, Vicent, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Germán Sanchis Trilles, and Chara Tsoukala (2014). "CASMACAT: A Computer-assisted Translation Workbench". In: *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 25–28. URL: https://www.aclweb.org/anthology/E14-2007.

BIBLIOGRAPHY

ALPAC, Automatic Language Processing Advisory Committee (1966). *Language and Machines: Computers in Translation and Linguistics; A Report.* National Academy of Sciences, National Research Council.

Arthur, Philip, Graham Neubig, and Satoshi Nakamura (2016). "Incorporating Discrete Translation Lexicons into Neural Machine Translation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* Austin, Texas: Association for Computational Linguistics, pp. 1557–1567. URL: `https://aclweb.org/anthology/D16-1162`.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *ICLR.* URL: `http://arxiv.org/pdf/1409.0473v6.pdf`.

Banerjee, Satanjeev and Alon Lavie (2005). "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.* Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: `https://www.aclweb.org/anthology/W05-0909`.

Baroni, Marco and Silvia Bernardini (2005). "A new approach to the study of translationese: Machine-learning the difference between original and translated text". In: *Literary and Linguistic Computing* 21.3, pp. 259–274.

Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar (2009). “Statistical Approaches to Computer-Assisted Translation”. In: *Computational Linguistics* 35.1, pp. 3–28. URL: `https://www.aclweb.org/anthology/J09-1002`.

Barrault, Loïc, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri (2019). “Findings of the 2019 Conference on Machine Translation (WMT19)”. In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1).* Florence, Italy: Association for Computational Linguistics, pp. 1–61. URL: `https://www.aclweb.org/anthology/W19-5301`.

Barrière, Caroline and Pierre Isabelle (2011). “Searching Parallel Corpora for Contextually Equivalent Terms”. In: *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT).* Ed. by Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste. Leuven, Belgium, pp. 105–112.

Belinkov, Yonatan, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass (2017). “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). Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 1–10. URL: `https://www.aclweb.org/anthology/I17-1001`.

Beller, Charley, Rebecca Knowles, Craig Harman, Shane Bergsma, Margaret Mitchell, and Benjamin Van Durme (2014). "I'm a Belieber: Social Roles via Self-identification and Conceptual Attributes". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 181–186. URL: `https://www.aclweb.org/anthology/P14-2030`.

Bender, Oliver, Saša Hasan, David Vilar, Richard Zens, and Hermann Ney (2005). "Comparison of generation strategies for interactive machine translation". In: *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*. Budapest, Hungary. URL: `http://www-i6.informatik.rwth-aachen.de/publications/download/276/Bender-EAMT-2005.pdf`.

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc.

Blain, Frédéric, Holger Schwenk, and Jean Senellart (2012). "Incremental adaptation using translation information and post-editing analysis". In: *IWSLT*. URL: `http://hltc.cse.ust.hk/iwslt/proceedings/paper_32.pdf`.

Blain, Frédéric, Fethi Bougares, Amir Hazem, Loïc Barrault, and Holger Schwenk (2015). "Continuous Adaptation to User Feedback for Statistical Machine Translation". In: *Proceedings of the 2015 Conference of the North American Chapter* *of the Association for Computational Linguistics: Human Language Technologies.* Denver, Colorado: Association for Computational Linguistics, pp. 1001–1005. URL: `https://www.aclweb.org/anthology/N15-1103`.

Blunsom, Phil and Miles Osborne (2008). "Probabilistic Inference for Machine Translation". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.* Honolulu, Hawaii: Association for Computational Linguistics, pp. 215–223. URL: `https://www.aclweb.org/anthology/D08-1023`.

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.* Berlin, Germany: Association for Computational Linguistics, pp. 131–198. URL: `https://www.aclweb.org/anthology/W16-2301`.

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 Conference on Machine Translation (WMT17)". In: *Proceedings of the Second Conference on Machine Translation.*

Copenhagen, Denmark: Association for Computational Linguistics, pp. 169–214. URL: https://www.aclweb.org/anthology/W17-4717.

Bojar, Ondřej, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz (2018). “Findings of the 2018 Conference on Machine Translation (WMT18)”. In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, pp. 272–303. URL: https://www.aclweb.org/anthology/W18-6401.

Bowker, Lynne (2002). *Computer-aided Translation Technology: A Practical Introduction*. Didactics of translation series. University of Ottawa Press. URL: https://books.google.com/books?id=ly29-mc6dOOC.

Cadwell, Patrick, Sharon O’Brien, and Carlos S. C. Teixeira (2018). “Resistance and accommodation: factors for the (non-) adoption of machine translation among professional translators”. In: *Perspectives* 26.3, pp. 301–321. eprint: https://doi.org/10.1080/0907676X.2017.1337210. URL: https://doi.org/10.1080/0907676X.2017.1337210.

Callison-Burch, Chris, Colin Bannard, and Josh Schroeder (2004). “Searchable Translation Memories”. In: *Proceedings of ASLIB Translating and the Computer 26*. URL: http://mt-archive.info/Aslib-2004-Callison-Burch.pdf.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn (2006). “Re-evaluation the Role of BLEU in Machine Translation Research”. In: *11th Conference of the*

European Chapter of the Association for Computational Linguistics. Trento, Italy: Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/E06-1032`.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia (2012). "Findings of the 2012 Workshop on Statistical Machine Translation". In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, pp. 10–51. URL: `https://www.aclweb.org/anthology/W12-3102`.

Carpuat, Marine (2009). "One Translation Per Discourse". In: *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*. Boulder, Colorado: Association for Computational Linguistics, pp. 19–27. URL: `https://www.aclweb.org/anthology/W09-2404`.

Carpuat, Marine, Hal Daumé III, Alexander Fraser, Chris Quirk, Fabienne Braune, Ann Clifton, Ann Irvine, Jagadeesh Jagarlamudi, John Morgan, Majid Razmara, Aleš Tamchyna, Katharine Henry, and Rachel Rudinger (2012). "Domain Adaptation in Machine Translation: Final Report". In: *2012 Johns Hopkins Summer Workshop Final Report*. URL: `http://hal3.name/damt/`.

Cettolo, Mauro, Nicola Bertoldi, Marcello Federico, Holger Schwenk, Loïc Barrault, and Christophe Servan (2014). "Translation project adaptation for MT-enhanced computer assisted translation". In: *Machine Translation* 28.2, pp. 127–150.

Chan, Sin-wai (2002). *Translation and Information Technology*. Translation studies. Chinese University Press.

Chatterjee, Rajen, Matteo Negri, Raphael Rubino, and Marco Turchi (2018). "Findings of the WMT 2018 Shared Task on Automatic Post-Editing". In: *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, pp. 723–738. URL: `https://www.aclweb.org/anthology/W18-6453`.

Cherry, Colin and George Foster (2012). "Batch Tuning Strategies for Statistical Machine Translation". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, pp. 427–436. URL: `https://www.aclweb.org/anthology/N12-1047`.

Chiang, David (2007). "Hierarchical Phrase-Based Translation". In: *Comput. Linguist.* 33.2, pp. 201–228. URL: `http://dx.doi.org/10.1162/coli.2007.33.2.201`.

Chiang, David, Kevin Knight, and Wei Wang (2009). "11,001 New Features for Statistical Machine Translation". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, pp. 218–226. URL: `https://www.aclweb.org/anthology/N/N09/N09-1025`.

Cho, Kyunghyun (2015). *Neural Machine Translation Tutorial (DL4MT Winter School)*. Tech. rep. Dublin, Ireland.

Chu, Chenhui, Raj Dabre, and Sadao Kurohashi (2017). “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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 385–391. URL: https://www.aclweb.org/anthology/P17-2061.

Church, Kenneth W. and William A. Gale (1991). “Concordances for Parallel Text”. In: *Seventh Annual Conference of the UW Centre for the New OED and Text Research*. Oxford, England. URL: http://www.cs.jhu.edu/~kchurch/wwwfiles/bilingual_concordances.ps.

Currey, Anna, Antonio Valerio Miceli Barone, and Kenneth Heafield (2017). “Copied Monolingual Data Improves Low-Resource Neural Machine Translation”. In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 148–156. URL: https://www.aclweb.org/anthology/W17-4715.

Daems, Joke and Lieve Macken (2019). “Interactive adaptive SMT versus interactive adaptive NMT: a user experience evaluation”. In: *Machine Translation*. URL: https://doi.org/10.1007/s10590-019-09230-z.

Dandapat, Sandipan, Sara Morrissey, Andy Way, and Mikel L. Forcada (2011). “Using Example-Based MT to Support Statistical MT when Translating Homogeneous

Data in a Resource-Poor Setting". In: *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*. Ed. by Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste. Leuven, Belgium, pp. 201–208.

De Sutter, Nathalie (2011). "MT Evaluation based on post-editing: a proposal". In: *Perspectives on translation quality*. Ed. by Ilse Depraetere. Berlin and Boston: De Gruyter Mouton, pp. 125–144.

Denkowski, Michael (2015). "Machine Translation for Human Translators". PhD thesis. Carnegie Mellon University.

Denkowski, Michael, Chris Dyer, and Alon Lavie (2014). "Learning from Post-Editing: Online Model Adaptation for Statistical Machine Translation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 395–404. URL: `https://www.aclweb.org/anthology/E14-1042`.

Ding, Shuoyang, Hainan Xu, and Philipp Koehn (2019). "Saliency-driven Word Alignment Interpretation for Neural Machine Translation". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, pp. 1–12. URL: `https://www.aclweb.org/anthology/W19-5201`.

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: Volume 2, Shared Task Papers. Berlin, Germany: Association for Computational Linguistics, pp. 272–280. URL: https://www.aclweb.org/anthology/W16-2310.

Domingo, Miguel, Álvaro Peris, and Francisco Casacuberta (2017). "Segment-based interactive-predictive machine translation". In: *Machine Translation* 31.4, pp. 163–185. URL: https://doi.org/10.1007/s10590-017-9213-3.

Dorr, Bonnie J., Pamela W. Jordan, and John W. Benoit (1999). "A survey of current paradigms in machine translation". In: *Advances in computers*. Vol. 49. Elsevier, pp. 1–68.

Duh, Kevin (2018). *The Multitarget TED Talks Task*. http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/.

Durrani, Nadir, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn (2013). "Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?" In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 399–405. URL: https://www.aclweb.org/anthology/P13-2071.

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. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: `https://www.aclweb.org/anthology/N13-1073`.

Esplà, Miquel, Felipe Sánchez-Martínez, and Mikel L. Forcada (2011). "Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited". In: *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*. Ed. by Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste. Leuven, Belgium, pp. 81–88.

Fadaee, Marzieh, Arianna Bisazza, and Christof Monz (2017). "Data Augmentation for Low-Resource Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 567–573. URL: `https://www.aclweb.org/anthology/P17-2090`.

Farajian, M. Amin, Marco Turchi, Matteo Negri, and Marcello Federico (2017). "Multi-Domain Neural Machine Translation through Unsupervised Adaptation". In: *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Paper*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 127–137. URL: `https://www.aclweb.org/anthology/W17-4713`.

Federico, Marcello, Alessandro Cattelan, and Marco Trombetti (2012). "Measuring User Productivity in Machine Translation Enhanced Computer Assisted Translation". In: *Proceedings of the Tenth Conference of the Association for Machine Translation* *in the Americas (AMTA)*. San Diego, California. URL: http://www.mt-archive.info/AMTA-2012-Federico.pdf.

Federico, Marcello, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frederic Blain, Philipp Koehn, Christian Buck, and Ulrich Germann (2014). "THE MATECAT TOOL". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 129–132. URL: https://www.aclweb.org/anthology/C14-2028.

Fellbaum, Christiane, ed. (1998). *WordNet: An Electronic Lexical Database*. MIT Press. URL: https://wordnet.princeton.edu/.

Foster, George (2002). "Text Prediction for Translators". PhD thesis. Université de Montréal.

Foster, George, Pierre Isabelle, and Pierre Plamondon (1997). "Target-Text Mediated Interactive Machine Translation". In: *Machine Translation* 12.1, pp. 175–194. URL: https://doi.org/10.1023/A:1007999327580.

Foster, George, Philippe Langlais, and Guy Lapalme (2002). "User-Friendly Text Prediction For Translators". In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, pp. 148–155. URL: https://www.aclweb.org/anthology/W02-1020.

Freitag, Markus and Yaser Al-Onaizan (2016). “Fast Domain Adaptation for Neural Machine Translation”. In: *CoRR* abs/1612.06897. arXiv: 1612.06897. URL: http://arxiv.org/abs/1612.06897.

Gage, Philip (1994). “A New Algorithm for Data Compression”. In: *C Users J.* 12(2):23–38.

Gale, William A., Kenneth W. Church, and David Yarowsky (1992). “One Sense Per Discourse”. In: *Proceedings of the Workshop on Speech and Natural Language*. HLT ’91. Harriman, New York: Association for Computational Linguistics, pp. 233–237. URL: https://doi.org/10.3115/1075527.1075579.

Galley, Michel and Christopher D. Manning (2008). “A Simple and Effective Hierarchical Phrase Reordering Model”. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 848–856. URL: https://www.aclweb.org/anthology/D08-1089.

Gandrabur, Simona and George Foster (2003). “Confidence estimation for translation prediction”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. CONLL ’03. Edmonton, Canada, pp. 95–102. URL: https://www.aclweb.org/anthology/W03-0413.

González-Rubio, Jesús, Daniel Ortiz-Martínez, and Francisco Casacuberta (2010). "Balancing User Effort and Translation Error in Interactive Machine Translation via Confidence Measures". In: *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, pp. 173–177. URL: https://www.aclweb.org/anthology/P10-2032.

Green, Spence (2014). "Mixed-Initiative Natural Language Translation". PhD thesis. Stanford University.

Green, Spence, Jeffrey Heer, and Christopher D. Manning (2013). "The Efficacy of Human Post-editing for Language Translation". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, pp. 439–448. URL: http://doi.acm.org/10.1145/2470654.2470718.

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 of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1225–1236. URL: https://www.aclweb.org/anthology/D14-1130.

Grissom II, Alvin, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III (2014). "Don't Until the Final Verb Wait: Reinforcement Learning for Simultaneous Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1342–1352. URL: https://www.aclweb.org/anthology/D14-1140.

Gu, Jiatao, Zhengdong Lu, Hang Li, and Victor O.K. Li (2016). “Incorporating Copying Mechanism in Sequence-to-Sequence Learning”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1631–1640. URL: https://www.aclweb.org/anthology/P16-1154.

Gulcehre, Caglar, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio (2016). “Pointing the Unknown Words”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 140–149. URL: https://www.aclweb.org/anthology/P16-1014.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn (2013). “Scalable Modified Kneser-Ney Language Model Estimation”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 690–696. URL: https://www.aclweb.org/anthology/P13-2121.

Heyn, Matthias (1996). “Integrating machine translation into translation memory systems”. In: *Proceedings of the EAMT Machine Translation Workshop, TKE96*, pp. 113–126.

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post (2017). “Sockeye: A Toolkit for Neural Machine Translation”. In: *arXiv preprint arXiv:1712.05690*. arXiv: 1712.05690 [cs.CL]. URL: https://arxiv.org/abs/1712.05690.

Hokamp, Chris and Qun Liu (2017). “Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, P1535–1546. URL: https://www.aclweb.org/anthology/P17-1141.

Huang, Liang and David Chiang (2007). “Forest Rescoring: Faster Decoding with Integrated Language Models”. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 144–151. URL: https://www.aclweb.org/anthology/P/P07/P07-1019.

Hutchins, John (1998). “The Origins of the Translator’s Workstation”. In: *Machine Translation* 13.4, pp. 287–307. URL: https://doi.org/10.1023/A:1008123410206.

Isabelle, Pierre, George Foster Marc Dymetman, Jean-Marc Jutras, Elliott Macklovitch, François Perrault, XiaoPo Ren, and Michel Simard (1993). “Translation Analysis and Translation Automation”. In: *Proceedings of the Fifth International Conference* *on Theoretical and Methodological Issues in Machine Translation.* Kyoto, Japan. URL: http://www.iro.umontreal.ca/~foster/papers/trans-tmi93.pdf.

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.* Lisbon, Portugal: Association for Computational Linguistics, pp. 134–140. URL: https://www.aclweb.org/anthology/W15-3014.

Junczys-Dowmunt, Marcin (2012). "A Phrase Table without Phrases: Rank Encoding for Better Phrase Table Compression". In: *Proceedings of th 16th International Conference of the European Association for Machine Translation (EAMT).* Ed. by Mauro Cettolo, Marcello Federico, Lucia Specia, and Andy Way. Trento, Italy, pp. 245–252. URL: http://www.mt-archive.info/EAMT-2012-Junczys-Dowmunt.

Junczys-Dowmunt, Marcin (2019). "Microsoft Translator at WMT 2019: Towards Large-Scale Document-Level Neural Machine Translation". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1).* Florence, Italy: Association for Computational Linguistics, pp. 225–233. URL: https://www.aclweb.org/anthology/W19-5321.

Junczys-Dowmunt, Marcin, Tomasz Dwojak, and Hieu Hoang (2016). "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. URL: http://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_4.pdf.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch (2018). "Marian: Fast Neural Machine Translation in C++". In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, pp. 116–121. URL: https://www.aclweb.org/anthology/P18-4020.

Kalchbrenner, Nal and Phil Blunsom (2013). "Recurrent Continuous Translation Models". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1700–1709. URL: https://www.aclweb.org/anthology/D13-1176.

Karimova, Sariya, Patrick Simianer, and Stefan Riezler (2018). "A User-study on Online Adaptation of Neural Machine Translation to Human Post-edits". In: *Machine Translation* 32.4, pp. 309–324. URL: https://doi.org/10.1007/s10590-018-9224-8.

Kay, Martin (1980). *The Proper Place of Men and Machines in Language Translation.* Palo Alto Research Center: Xerox Corporation. URL: http://www.mt-archive.info/Kay-1980.pdf.

Khayrallah, Huda and Philipp Koehn (2018). “On the Impact of Various Types of Noise on Neural Machine Translation”. In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, pp. 74–83. URL: `https://www.aclweb.org/anthology/W18-2709`.

Khayrallah, Huda, Rebecca Knowles, Kevin Duh, and Matt Post (2019). “An Interactive Teaching Tool for Introducing Novices to Machine Translation”. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE ’19. Minneapolis, Minnesota, USA: ACM, pp. 1276–1276. URL: `http://doi.acm.org/10.1145/3287324.3293840`.

Kingma, Diederik P. and Jimmy Ba (2014). “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980. arXiv: `1412.6980`. URL: `http://arxiv.org/abs/1412.6980`.

Kirov, Christo, John Sylak-Glassman, Rebecca Knowles, Ryan Cotterell, and Matt Post (2017). “A Rich Morphological Tagger for English: Exploring the Cross-Linguistic Tradeoff Between Morphology and Syntax”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 112–117. URL: `https://www.aclweb.org/anthology/E17-2018`.

Knowles, Rebecca, Josh Carroll, and Mark Dredze (2016). “Demographer: Extremely Simple Name Demographics”. In: *Proceedings of the First Workshop on NLP and*

Computational Social Science. Austin, Texas, USA: Association for Computational Linguistics, pp. 108–113. URL: https://aclweb.org/anthology/W16-5614.

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, Texas, USA. URL: https://amtaweb.org/wp-content/uploads/2016/10/AMTA2016_Research_Proceedings_v7.pdf#page=113.

Knowles, Rebecca and Philipp Koehn (2018a). "Context and Copying in Neural Machine Translation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3034–3041. URL: https://www.aclweb.org/anthology/D18-1339.

Knowles, Rebecca and Philipp Koehn (2018b). "Lightweight Word-Level Confidence Estimation for Neural Interactive Translation Prediction". In: *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*. Boston, Massachusetts, USA: Association for Machine Translation in the Americas, pp. 35–40. URL: https://www.aclweb.org/anthology/W18-2102.

Knowles, Rebecca, John Ortega, and Philipp Koehn (2018). "A Comparison of Machine Translation Paradigms for Use in Black-Box Fuzzy-Match Repair". In: *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*. Boston, Massachusetts, USA: Association for Machine Translation in the Americas, pp. 249–255. URL: https://www.aclweb.org/anthology/W18-2108.

Knowles, Rebecca, Marina Sanchez-Torron, and Philipp Koehn (2019). "A user study of neural interactive translation prediction". In: *Machine Translation*. URL: https://doi.org/10.1007/s10590-019-09235-8.

Knowles, Rebecca, Mark Dredze, Kathleen Evans, Elyse Lasser, Tom Richards, Jonathan Weiner, and Hadi Kharrazi (2014). "High Risk Pregnancy Prediction from Clinical Text". In: *NeurIPS Workshop on Machine Learning for Clinical Data Analysis*. Montreal, Canada.

Knowles, Rebecca, Adithya Renduchintala, Philipp Koehn, and Jason Eisner (2016). "Analyzing Learner Understanding of Novel L2 Vocabulary". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pp. 126–135. URL: https://www.aclweb.org/anthology/K16-1013.

Kobus, Catherine, Josep Crego, and Jean Senellart (2017). "Domain Control for Neural Machine Translation". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., pp. 372–378. URL: https://doi.org/10.26615/978-954-452-049-6_049.

Koehn, Philipp (2005). "Europarl: A parallel corpus for statistical machine translation". In: *MT summit*. Vol. 5, pp. 79–86.

Koehn, Philipp (2009). “A process study of computer-aided translation”. In: *Machine Translation* 23.4, pp. 241–263. URL: http://www.jstor.org/stable/40783527.

Koehn, Philipp (2010). *Statistical Machine Translation*. 1st. New York, NY, USA: Cambridge University Press.

Koehn, Philipp and Ulrich Germann (2014). “The Impact of Machine Translation Quality on Human Post-Editing”. In: *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 38–46. URL: https://www.aclweb.org/anthology/W14-0307.

Koehn, Philipp and Barry Haddow (2009). “Edinburgh’s Submission to all Tracks of the WMT2009 Shared Task with Reordering and Speed Improvements to Moses”. In: *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece: Association for Computational Linguistics, pp. 160–164. URL: https://www.aclweb.org/anthology/W/W09/W09-0429.

Koehn, Philipp and Barry Haddow (2012). “Interpolated backoff for factored translation models”. In: *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

Koehn, Philipp and Rebecca Knowles (2017). “Six Challenges for Neural Machine Translation”. In: *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver, Canada: Association for Computational Linguistics, pp. 28–39. URL: https://www.aclweb.org/anthology/W17-3204.

Koehn, Philipp and Jean Senellart (2010). “Convergence of Translation Memory and Statistical Machine Translation”. In: *AMTA Workshop on MT Research and the Translation Industry*. URL: http://mt-archive.info/JEC-2010-Koehn.pdf.

Koehn, Philipp, Chara Tsoukala, and Herve Saint-Amand (2014). “Refinements to Interactive Translation Prediction Based on Search Graphs”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 574–578. URL: https://www.aclweb.org/anthology/P14-2094.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst (2007). “Moses: Open Source Toolkit for Statistical Machine Translation”. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, pp. 177–180. URL: https://www.aclweb.org/anthology/P/P07/P07-2045.

Koponen, Maarit (2012). “Comparing human perceptions of post-editing effort with post-editing operations”. In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, pp. 181–190. URL: https://www.aclweb.org/anthology/W12-3123.

Koponen, Maarit (2016). “Is Machine Translation Post-editing Worth the Effort? A Survey of Research into Post-editing and Effort”. In: *The Journal of Specialised Translation*, pp. 131–148.

Kothur, Sachith Sri Ram, Rebecca Knowles, and Philipp Koehn (2018). “Document-Level Adaptation for Neural Machine Translation”. In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, pp. 64–73. URL: https://www.aclweb.org/anthology/W18-2708.

Kumar, Shankar and William Byrne (2004). “Minimum Bayes-Risk Decoding for Statistical Machine Translation”. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Boston, Massachusetts, USA: Association for Computational Linguistics, pp. 169–176. URL: https://www.aclweb.org/anthology/N04-1022.

Lam, Tsz Kin, Julia Kreutzer, and Stefan Riezler (2018). “A Reinforcement Learning Approach to Interactive-Predictive Neural Machine Translation”. In: URL: http://www.cl.uni-heidelberg.de/~riezler/publications/papers/EAMT2018.pdf.

Langlais, Philippe, George Foster, and Guy Lapalme (2000). “TransType: a Computer-Aided Translation Typing System”. In: *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*. URL: https://www.aclweb.org/anthology/W00-0507.

Langlois, Lucie, Michel Simard, and Elliott Macklovitch (2016). “Machine Translation of Canadian Court Decisions”. In: *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*. URL: https://amtaweb.org/wp-content/uploads/2017/02/AMTA2016_User_Track_Proceedings_v9updated_gov_papers.pdf.

Läubli, Samuel, Rico Sennrich, and Martin Volk (2018). “Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4791–4796. URL: https://www.aclweb.org/anthology/D18-1512.

Läubli, Samuel, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk (2013). “Assessing post-editing efficiency in a realistic translation environment”. In: *Proceedings of Workshop on Post-editing Technology and Practice*, pp. 83–91. URL: http://www.mt-archive.info/10/MTS-2013-W2-Laubli.pdf.

Li, Xiaoqing, Jiajun Zhang, and Chengqing Zong (2018). “One Sentence One Model for Neural Machine Translation”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. Miyazaki, Japan: European Languages Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1146.

Lison, Pierre, Jörg Tiedemann, and Milen Kouylekov (2018). “OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora”. In:

Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Ed. by Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hlne Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA).

Luong, Minh-Thang and Christopher D Manning (2015). “Stanford neural machine translation systems for spoken language domains”. In: *Proceedings of the International Workshop on Spoken Language Translation*.

Luong, Minh-Thang and Christopher D. Manning (2016). “Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1054–1063. URL: `https://www.aclweb.org/anthology/P16-1100`.

Luong, Thang, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba (2015). “Addressing the Rare Word Problem in Neural Machine Translation”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 11–19. URL: `https://www.aclweb.org/anthology/P15-1002`.

Ma, Qingsong, Johnny Wei, Ondřej Bojar, and Yvette Graham (2019). "Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, pp. 62–90. URL: `https://www.aclweb.org/anthology/W19-5302`.

Macklovitch, Elliott, Guy Lapalme, and Fabrizio Gotti (2008). "TransSearch: What are translators looking for?" In: *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA)*. Waikiki, Hawaii, pp. 412–419. URL: `http://www.mt-archive.info/AMTA-2008-Macklovitch.pdf`.

Marcu, Daniel (2001). "Towards a Unified Approach to Memory- and Statistical-based Machine Translation". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. ACL '01. Toulouse, France: Association for Computational Linguistics, pp. 386–393. URL: `https://doi.org/10.3115/1073012.1073062`.

Martindale, Marianna and Marine Carpuat (2018). "Fluency Over Adequacy: A Pilot Study in Measuring User Trust in Imperfect MT". In: *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*. Boston, MA: Association for Machine Translation in the Americas, pp. 13–25. URL: `https://www.aclweb.org/anthology/W18-1803`.

Maruf, Sameen and Gholamreza Haffari (2018). "Document Context Neural Machine Translation with Memory Networks". In: *Proceedings of the 56th Annual Meeting of* *the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1275–1284. URL: https://www.aclweb.org/anthology/P18-1118.

Moorkens, Joss and Sharon O'Brien (2015). "Post-Editing Evaluations: Trade-offs between Novice and Professional Participants". In: *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*. Antalya, Turkey, pp. 75–81. URL: https://www.aclweb.org/anthology/W15-4910.

Moorkens, Joss and Sharon O'Brien (2017). "Assessing User Interface Needs of Post-Editors of Machine Translation". In: *Human Issues in Translation Technology*. Ed. by Dorothy Kenny, pp. 109–130.

Nguyen, Toan and David Chiang (2018). "Improving Lexical Choice in Neural Machine Translation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 334–343. URL: https://www.aclweb.org/anthology/N18-1031.

O'Brien, Sharon (2002). "Teaching Post-editing: A Proposal for Course Content". In: *6th EAMT Workshop Teaching Machine Translation*. Manchester, pp. 99–106.

Och, Franz Josef (1999). "An Efficient Method for Determining Bilingual Word Classes". In: *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 71–76.

Och, Franz Josef, Richard Zens, and Hermann Ney (2003). “Efficient Search for Interactive Statistical Machine Translation”. In: *10th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary: Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/E03-1032`.

Ott, Myle, Michael Auli, David Grangier, and Marc’Aurelio Ranzato (2018). “Analyzing Uncertainty in Neural Machine Translation”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmssan, Stockholm Sweden: PMLR, pp. 3956–3965. URL: `http://proceedings.mlr.press/v80/ott18a.html`.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. URL: `https://www.aclweb.org/anthology/P02-1040`.

Peris, Álvaro and Francisco Casacuberta (2019). “Online learning for effort reduction in interactive neural machine translation”. In: *Computer Speech & Language* 58, pp. 98 –126. URL: `http://www.sciencedirect.com/science/article/pii/S0885230818300536`.

Peris, Álvaro, Luis Cebrián, and Francisco Casacuberta (2017). "Online Learning for Neural Machine Translation Post-editing". In: *CoRR* abs/1706.03196. arXiv: 1706.03196. URL: http://arxiv.org/abs/1706.03196.

Peris, Álvaro, Miguel Domingo, and Francisco Casacuberta (2017). "Interactive neural machine translation". In: *Computer Speech & Language* 45, pp. 201 –220. URL: http://www.sciencedirect.com/science/article/pii/S0885230816301000.

Plitt, Mirko and François Masselot (2010). "A productivity test of statistical machine translation post-editing in a typical localisation context". In: *The Prague bulletin of mathematical linguistics* 93, pp. 7–16.

Post, Matt and David Vilar (2018). "Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1314–1324. URL: https://www.aclweb.org/anthology/N18-1119.

Renduchintala, Adithya, Rebecca Knowles, Philipp Koehn, and Jason Eisner (2016a). "Creating Interactive Macaronic Interfaces for Language Learning". In: *Proceedings of ACL-2016 System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, pp. 133–138. URL: https://www.aclweb.org/anthology/P16-4023.

Renduchintala, Adithya, Rebecca Knowles, Philipp Koehn, and Jason Eisner (2016b). “User Modeling in Language Learning with Macaronic Texts”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1859–1869. URL: `https://www.aclweb.org/anthology/P16-1175`.

Rico Pérez, Celia and Enrique Torrejón (2012). “Skills and Profile of the New Role of the Translator as MT Post-editor”. In: *Revista Tradumtica: tecnologies de la traducci* 10, pp. 166–178.

Rodriguez Vazquez, Silvia, Sharon O’Brien, and Dónal Fitzpatrick (2017). *Usability of web-based MT post-editing environments for screen reader users*. eng. ID: unige:97893. URL: `https://archive-ouverte.unige.ch/unige:97893`.

Sanchez Torron, Marina (2017). “Productivity in Post-Editing and in Neural Interactive Translation Prediction: a Study of English-to-Spanish Professional Translators”. PhD thesis. University of Auckland. URL: `http://hdl.handle.net/2292/37205`.

Sanchez-Torron, Marina and Philipp Koehn (2016). “Machine Translation Quality and Post-Editor Productivity”. In: *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*. URL: `https://amtaweb.org/wp-content/uploads/2016/10/AMTA2016_Research_Proceedings_v7.pdf#page=22`.

Sanchis-Trilles, Germán, Daniel Ortiz-Martínez, Jorge Civera, Francisco Casacuberta, Enrique Vidal, and Hieu Hoang (2008). “Improving Interactive Machine Translation via Mouse Actions". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 485–494. URL: https://www.aclweb.org/anthology/D08-1051.

Sanchis-Trilles, Germán, Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L. Hill, Philipp Koehn, Luis A. Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Chara Tsoukala, and Enrique Vidal (2014). "Interactive translation prediction versus conventional post-editing in practice: a study with the CasMaCat workbench". In: *Machine Translation* 28.3, pp. 217–235. URL: https://doi.org/10.1007/s10590-014-9157-9.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016a). "Edinburgh Neural Machine Translation Systems for WMT 16". In: *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, pp. 371–376. URL: https://www.aclweb.org/anthology/W16-2323.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016b). "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)*. Berlin, Germany: Association for Computational Linguistics, pp. 86–96. URL: https://www.aclweb.org/anthology/P16-1009.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016c). "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)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: `https://www.aclweb.org/anthology/P16-1162`.

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 the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, pp. 65–68. URL: `http://aclweb.org/anthology/E17-3017`.

Servan, Christophe, Josep Maria Crego, and Jean Senellart (2016). "Domain specialization: a post-training domain adaptation for Neural Machine Translation". In: *CoRR* abs/1612.06141. arXiv: `1612.06141`. URL: `http://arxiv.org/abs/1612.06141`.

Shi, Xing, Kevin Knight, and Deniz Yuret (2016). "Why Neural Translations are the Right Length". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2278–2282. URL: `https://www.aclweb.org/anthology/D16-1248`.

Simard, Michel and Elliott Macklovitch (2005). "Studying the Human Translation Process through the TransSearch Log-Files". In: *Knowledge Collection from Volunteer Contributors: Papers from the 2005 Spring Symposium.* Ed. by Timothy Chklovski, Pedro Domingos, Henry Lieberman, Rada Mihalcea, and Push Singh. Menlo Park, California: American Association for Artificial Intelligence, pp. 70–77. URL: `http://rali.iro.umontreal.ca/rali/sites/default/files/publis/Simard-Macklovitch-KCVC05.pdf`.

Simianer, Patrick, Joern Wuebker, and John DeNero (2019). "Measuring Immediate Adaptation Performance for Neural Machine Translation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2038–2046. URL: `https://www.aclweb.org/anthology/N19-1206`.

Smith, Jason R., Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez (2013). "Dirt Cheap Web-Scale Parallel Text from the Common Crawl". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Sofia, Bulgaria: Association for Computational Linguistics, pp. 1374–1383. URL: `https://www.aclweb.org/anthology/P13-1135`.

Snover, Matthew, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul (2006). "A Study of Translation Edit Rate with Targeted Human Annotation". In: *5th Conference of the Association for Machine Translation in the Americas (AMTA)*. Boston, Massachusetts. URL: http://mt-archive.info/AMTA-2006-Snover.pdf.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. V Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger. Curran Associates, Inc., pp. 3104–3112. URL: http://papers.neurips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Teixeira, Carlos S.C. and Sharon O'Brien (2017). "Investigating the cognitive ergonomic aspects of translation tools in a workplace setting". In: *Translation Spaces* 6.1, pp. 79–103. URL: https://www.jbe-platform.com/content/journals/10.1075/ts.6.1.05tei.

Ter-Sarkisov, Alex, Holger Schwenk, Fethi Bougares, and Loïc Barrault (2015). "Incremental Adaptation Strategies for Neural Network Language Models". In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Beijing, China: Association for Computational Linguistics, pp. 48–56. URL: https://www.aclweb.org/anthology/W15-4006.

Tezcan, Arda and Vincent Vandeghinste (2011). "SMT-CAT integration in a Technical Domain: Handling XML Markup Using Pre & Post-processing Methods". In: *Proceedings of the 15th International Conference of the European Association for*

Machine Translation (EAMT). Ed. by Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste. Leuven, Belgium, pp. 55–62.

Tiedemann, Jörg (2009). “News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces”. In: *Recent Advances in Natural Language Processing*. Ed. by N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov. Vol. V. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, pp. 237–248.

Tiedemann, Jrg (2012). “Parallel Data, Tools and Interfaces in OPUS”. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis. Istanbul, Turkey: European Language Resources Association (ELRA).

Torregrosa, Daniel, Juan Antonio Pérez-Ortiz, and Mikel L. Forcada (2017). “Comparative Human and Automatic Evaluation of Glass-Box and Black-Box Approaches to Interactive Translation Prediction”. In: *The Prague Bulletin of Mathematical Linguistics* 108, pp. 97–108. URL: `https://ufal.mff.cuni.cz/pbml/108/art-torregrosa-perez-ortiz-forcada.pdf`.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer (2003). “Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network”. In: *Proceedings of the 2003 Human Language Technology Conference of the North*

American Chapter of the Association for Computational Linguistics, pp. 252–259. URL: https://www.aclweb.org/anthology/N03-1033.

Turchi, Marco, Matteo Negri, M Amin Farajian, and Marcello Federicoa (2017). "Continuous Learning from Human Post-Edits for Neural Machine Translation". In: *The Prague Bulletin of Mathematical Linguistics* 108, pp. 233–244.

Ueffing, Nicola, Franz Josef Och, and Hermann Ney (2002). "Generation of Word Graphs in Statistical Machine Translation". In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, pp. 156–163. URL: https://www.aclweb.org/anthology/W02-1021.

Vasconcellos, Muriel and Marjorie León (1985). "SPANAM and ENGSPAN: machine translation at the Pan American Health Organization". In: *Computational Linguistics* 11.2-3, pp. 122–136. URL: https://www.aclweb.org/anthology/J85-2003.

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 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, Rui, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita (2017). "Instance Weighting for Neural Machine Translation Domain Adaptation". In:

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1482–1488. URL: `https://www.aclweb.org/anthology/D17-1155`.

Weaver, Warren (1949). *Translation.* URL: `http://www.mt-archive.info/Weaver-1949.pdf`.

Wu, Jian-Cheng, Kevin C. Yeh, Thomas C. Chuang, Wen-Chi Shei, and Jason S. Chang (2003). "TotalRecall: A Bilingual Concordance for Computer Assisted Translation and Language Learning". In: *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 201–204. URL: `https://www.aclweb.org/anthology/P03-2038`.

Wu, Zhibiao and Martha Palmer (1994). "Verb Semantics and Lexical Selection". In: *32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico, USA: Association for Computational Linguistics, pp. 133–138. URL: `https://www.aclweb.org/anthology/P94-1019`.

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 (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 66–75. URL: `https://www.aclweb.org/anthology/P16-1007`.

Zeiler, Matthew D. (2012). “ADADELTA: An Adaptive Learning Rate Method”. In: *CoRR* abs/1212.5701. arXiv: `1212.5701`. URL: `http://arxiv.org/abs/1212.5701`.

Vita

Rebecca Knowles received her B.S. degree in Linguistics and Mathematics with a concentration in Computer Science from Haverford College in 2012. She entered the Computer Science Ph.D. program at Johns Hopkins University in the fall of 2013, after a year working as a research assistant and substitute teacher. During her Ph.D., she was supported by a National Science Foundation Graduate Research Fellowship, awarded in 2013. She was also awarded the Jun Wu and Yan Zhang Endowed Graduate Student Fellowship in 2013. She received her M.S.E. in Computer Science in the spring of 2015. She served as a teaching assistant for Data Structures, taught two courses on computer aided translation aimed at first year undergraduates and a computational linguistics course at a local middle school, received a Teaching Academy Certificate after participating in the 2017-2018 cohort, and was the local organizer of Johns Hopkins University's chapter of the North American Computational Linguistics Olympiad. Her research has focused on machine translation and computer aided translation.