Machine Translation of User-Generated Content

Pintu Lohar

B.E., M.E.

A dissertation submitted in fulfillment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to

DCU

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Dublin City University
School of Computing

Supervisors:
Professor Andy Way, Dr. Haithem Afli, Dr. Maja Popović

September, 2020

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Pintu Lohar

(Candidate) ID No.: 15211412

Date: September, 2020

Contents

List of Figures

List of Tables

Abbreviations

BLEU	Bilingual Evaluation Understudy
BOV	Bag of Vectors
BOW	Bag of Words
chrF	character n-gram F-score
chrTER	character Translation Error Rate
CLIR	Crosslingual Information Retrieval
CLSA	Crosslingual Sentiment Analysis
DA	Dialectal Arabic
EM	Expectation Maximisation
FaDA	Fast Document Aligner
IMDb	Internet Movie Database
IR	Information Retrieval
JSC	Jaccard Similarity Coefficient
LM	Language Modelling
MADA	Morphological Analysis and Disambiguation for Arabic
MERT	Minimum Error Rate Training
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MSA	Modern Standard Arabic
MT	Machine Translation

NE	Named Entity
NLP	Natural Language Processing
NMT	Neural Machine Translation
OOV	Out of Vocabulary
PBMT	Phrase-Based Machine Translation
Q&A	Questions and Answers
RNN	Recurrent Neural Network
RQ	Research Question
SEtimes	South-East European Times
SMT	Statistical Machine Translation
SP	Sentiment Preservation
TER	Translation Error Rate
TWB	Translators without Borders
UGC	User Generated Content
URL	Uniform Resource Locator
WER	Word Error Rate

Machine Translation of User-Generated Content

Pintu Lohar

Abstract

The world of social media has undergone huge evolution during the last few years. With the spread of social media and online forums, individual users actively participate in the generation of online content in different languages from all over the world. Sharing of online content has become much easier than before with the advent of popular websites such as Twitter, Facebook etc. Such content is referred to as ‘User-Generated Content’ (UGC). Some examples of UGC are user reviews, customer feedback, tweets etc. In general, UGC is informal and noisy in terms of linguistic norms. Such noise does not create significant problems for human to understand the content, but it can pose challenges for several natural language processing applications such as parsing, sentiment analysis, machine translation (MT), etc.

An additional challenge for MT is sparseness of bilingual (translated) parallel UGC corpora. In this research, we explore the general issues in MT of UGC and set some research goals from our findings. One of our main goals is to exploit comparable corpora in order to extract parallel or semantically similar sentences. To accomplish this task, we design a document alignment system to extract semantically similar bilingual document pairs using the bilingual comparable corpora. We then apply strategies to extract parallel or semantically similar sentences from comparable corpora by transforming the document alignment system into a sentence alignment system. We seek to improve the quality of parallel data extraction for UGC translation and assemble the extracted data with the existing human translated resources. Another objective of this research is to demonstrate the usefulness of MT-based sentiment analysis. However, when using openly available systems such as Google Translate, the translation process may alter the sentiment in the target language. To cope with this phenomenon, we instead build fine-grained sentiment translation models that focus on sentiment preservation in the target language during translation.

Acknowledgments

First of all, I would like to express my deepest gratitude to my supervisor, Professor Andy Way for his esteemed guidance and immense support. He has consistently motivated me throughout this long journey, which kept me positive even in my difficult times. He presented himself not only as my supervisor but also as a friend so that I can conduct my research in a very friendly and comfortable environment.

I am also grateful to my co-supervisors Dr. Haithem Afli and Dr. Maja Popović for their continuous guidance and support. I would also like to thank my examiners Professor Ondřej Bojar, Dr. Annalina Caputo and chairperson Professor Cathal Gurrin.

My sincere gratitude goes to Alberto Poncelas and Eva Vanmassenhove with whom I started my PhD and shared some beautiful moments during the whole journey. I would also want to specially thank Chao-Hong Liu, Abhishek Kaushik for helping me through difficult times. My special thanks go to Guodong Xie for his help and support. In addition, I am also grateful to all my colleagues, post docs, research fellows, faculties and all staffs in ADAPT Centre. I would also like to thank all my friends who supported and motivated me since my childhood.

Last but not the least, I would like to express my deepest gratitude to my family members, specially my parents Sadho Lohar and Indrani Lohar, my brother Goutam Lohar and my sister Rohini Lohar, for their endless support and love for my entire life.

My whole PhD research has been fully supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106).

Chapter 1

Introduction

Natural Language Processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages. One of the main research areas in the field of NLP is Machine Translation (MT) that investigates the use of software to translate text or speech from one language (source) to another (target).

The main ingredient for MT is a parallel corpus, a collection of texts in the source language aligned with their translations into the target language. In addition, the parallel corpus normally needs to be clean, of high quality and sufficiently large in order to build a robust MT system.

In general, parallel corpora are not available for all domains, language pairs and/or types of texts, which poses challenges in MT. One example of such a challenging type of text is User Generated Content (UGC) which comprises online content generated by users from all over the world on social media.

1.1 Machine translation of user-generated content

Before discussing UGC translation, let us explain the characteristics of UGC.

1.1.1 Characteristics of UGC

UGC is usually informal in nature and often deviates from linguistic norms (Jiang et al., 2012). *Tweets*, *customer feedback*, *online reviews* etc. are good examples of UGC. In Twitter, users must write their messages in 280 characters. Due to this limitation, they tend to make the sentences shorter by omitting prepositions, using short forms, etc. They are informal and often not fully grammatical. As a result, tweets often create problems in NLP tasks involving UGC. In contrast, customer feedback, reviews etc. are generally cleaner and longer than tweets as they usually are not bound by such a character limitation.

1.1.2 Challenges in UGC translation

The topic of UGC translation has drawn special attention among researchers during the last few years. As mentioned earlier, translating UGC is always a challenging task because of the lack of parallel corpora in this field. Moreover, its informal nature and grammatical incorrectness aggravates problems in the translation process. For example, consider the tweet below:

Brazil 5 WorldCup championship Argentina 2 WorldCup championship so Ill go with Brazil

The above tweet contains multiple errors in terms of grammatical correctness and linguistic norms. For example, the verb *won* is missing after the word *Brazil* and *Argentina*. Moreover, *championship* should be plural, i.e. *championships* and *Ill* should be *I'll*. Such errors pose challenges in translation because most MT engines are built for naturally clean texts and so are incapable of properly translating informal texts.

1.2 Related work

A significant amount of research has been done in the area of UGC translation. For example, Jiang et al. (2012) demonstrate that building robust, high-quality MT engines can be problematic, especially when users deliberately decide to violate linguistic norms in the languages they speak. They propose to translate UGC with specific routines to handle shortforms, acronyms, typos, punctuation errors, non-dictionary slang, wordplay, censor avoidance and emoticons. Banerjee et al. (2011) use mixture modelling (Foster and Kuhn, 2007) to adapt their models for translating online user-generated forum data and the results show a more profound effect of language model adaptation over translation model adaptation with respect to translation quality.

Researchers also adopt normalisation and supplementary training data acquisition techniques that are guided by the goal of reducing the number of out-of-vocabulary (OOV) items in the target language with respect to the training data (Banerjee et al., 2012). Sajjad et al. (2013) present a dialectal Egyptian Arabic to English SMT system that leverages dialectal to modern standard Arabic (MSA) adaptation. Although a lot of work focuses on improving the quality of UGC-translation, the lack of parallel data has always been a major challenge in this area. To the best of our knowledge, no large parallel corpora are available for UGC.

1.3 Motivation

With the rapid development of Internet technology, people from different locations and cultural backgrounds now communicate via widely used social networking websites such as Twitter, Facebook etc., in different languages. Nowadays, the vast majority of Internet users are non-English speakers.

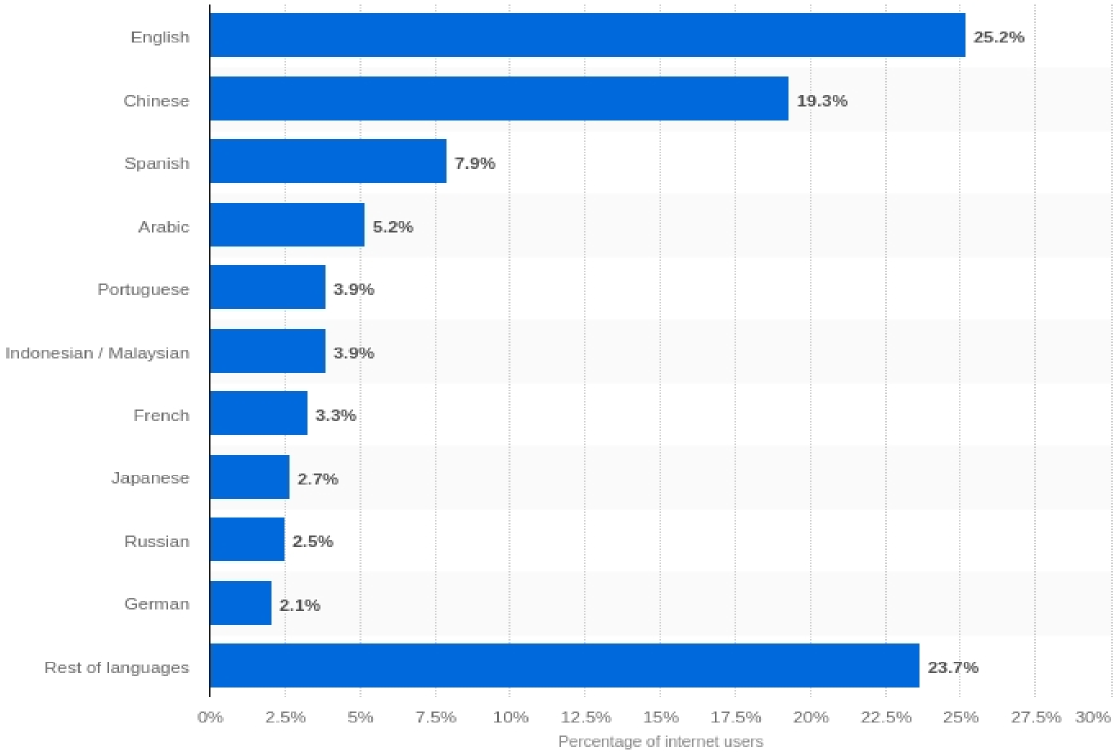


Figure 1.1: Language distribution[1]

Figure 1.1 shows the distribution of languages used on the Internet as of April, 2019. We can observe that only 25.2% of Internet users use English and the remaining 74.8% use other languages. It is clear that a large amount of UGC requires translation into English in order to use specific downstream language-analysis tools that are either available only in English, or whose English-language versions are better in quality than those of the other languages.

Most of the research work described in Section 1.2 exposes the difficulties in UGC translation and thus propose different approaches to resolve such problems. One of the major difficulties is the lack of parallel resources for training MT systems for UGC.

[1] https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/

MT-based sentiment analysis is the process of translating the UGC from the source language to the target language and then performing sentiment analysis in the target language. Such an approach has shown to be useful for sentiment analysis in multilingual platforms (Araujo et al., 2016; Mohammad et al., 2016). However, there is another big challenge in the area of MT-based sentiment analysis. It is known that MT can alter the sentiment during translation (Mohammad et al., 2016). Such a phenomenon is very harmful for MT-based sentiment analysis because the original sentiment polarity is not maintained in the target language. It is extremely important to preserve the sentiment of the original text during the translation process.

Considering the above scenarios in this thesis, we are interested in addressing the following problems in UGC translation.

- **Problem 1:** Scarcity of parallel UGC corpus
- **Problem 2:** Sentiment alteration in UGC translation

One of the ways to solve the first problem is to use back-translation (Poncelas et al., 2018; Sennrich et al., 2016a), which is the process of translating the monolingual data in the target language in order to create synthetic parallel data for MT. However, we are interested in different approaches in this research. We consider the use of a bilingual comparable corpus, a collection of similar texts/documents in two languages that can be considered as a useful resource for parallel corpus extraction. Such a corpus is likely to contain similar information in two languages and so can be exploited to extract similar bilingual sentences or phrases. However, the best way to obtain parallel texts is to initially align bilingual similar documents and then use those aligned documents for parallel sentence/phrase extraction. The extracted parallel data can then be added to the existing training data in order to build an enhanced MT system.

Another motivation in this research comes from the second problem which can pose major challenges in the area of MT-based sentiment analysis. For example, multinational companies like Amazon, eBay etc. always keep track of their customers' behaviour by analysing the underlying sentiment of the customer feedback to their products or services. As the feedback can be in multiple languages, it is often required to translate them into English and then use the sentiment analysis tools in English. If the sentiment is altered during translation, it can have a negative impact on the analysis process and wrong information is transferred in the target language. Let us assume that a feedback in Japanese with negative sentiment is translated into English with neutral or positive sentiment. This will result in wrong information transfer about the feedback and so the company would not know that the customer was unsatisfied with the product or service. This problem can be partially resolved by building a special kind of MT system that not only translates UGC but also focuses on preserving the sentiment of the source-language text during the translation process. As 'sentiment preservation' is the primary concern here, it is crucial to design *sentiment-specific* translation models rather than mainly focusing on translation quality per se. However, it is also important not to allow a significant loss in translation quality while maintaining sentiment polarity in UGC translation.

The purpose of this thesis can be framed in the following three research goals.

- **Research goal 1:** Implementation of a sophisticated bilingual document alignment system for a collection of available bilingual comparable document pairs in order to build comparable resources for UGC;
- **Research goal 2:** Implementation of an efficient parallel data extraction system from a comparable corpus of UGC;
- **Research goal 3:** Building a sentiment translation system for sentiment preservation.

1.4 Research questions

Each of the above research goals is described in terms of its corresponding research question (RQ), which we address in this thesis.

- **RQ-1: Provided with a collection of bilingual comparable documents, can we implement a sophisticated document alignment system that extracts semantically similar document pairs?**

  The similarity of bilingual documents depends upon the quality of the document alignment system, i.e. the better the quality of alignment is, the more similar the documents are. Once the document alignment system is implemented, its utility can be leveraged for parallel data extraction. This gives rise to our second research question.

- **RQ-2: Given the effectiveness of our document alignment system, can we implement an efficient, automatic, good quality parallel data extraction system from a comparable corpus of UGC?**

  The objective here is to utilise the full capability of the document alignment system for parallel resource development for MT engine training. However, as our alignment system is at document level, we transform it into a sentence-level alignment system in order to extract parallel sentences from comparable corpora.

Finally, we explore one of the most recent area of UGC translation, namely: 'sentiment preservation in MT' which is addressed in our third and final RQ as follows:

- **RQ-3: Can we build an MT-based sentiment preservation system using sentiment classification in order to best preserve the sentiment of the source-language texts during translation?**

To to the best of our knowledge, RQ-3 has never been investigated before. As we mentioned earlier, MT can alter the sentiment of the input document during translation. This phenomenon is more harmful for UGC than natural texts because UGC contains a certain degree of sentiment in many cases, whose sentiment polarity change can have a negative impact in MT-based sentiment analysis. For this reason, we aim at building sentiment translation systems based on sentiment classification with the purpose of maintaining sentiment polarity during the translation process.

All the above research questions can be visually described in Figure 1.2 that depicts the data models underlying the research conducted in this thesis.

1.5 Research contributions

Our contributions in this research can be summarised as follows.

- Firstly, we conduct experiments on the translation of different types of UGC and set distinct research goals from our findings.

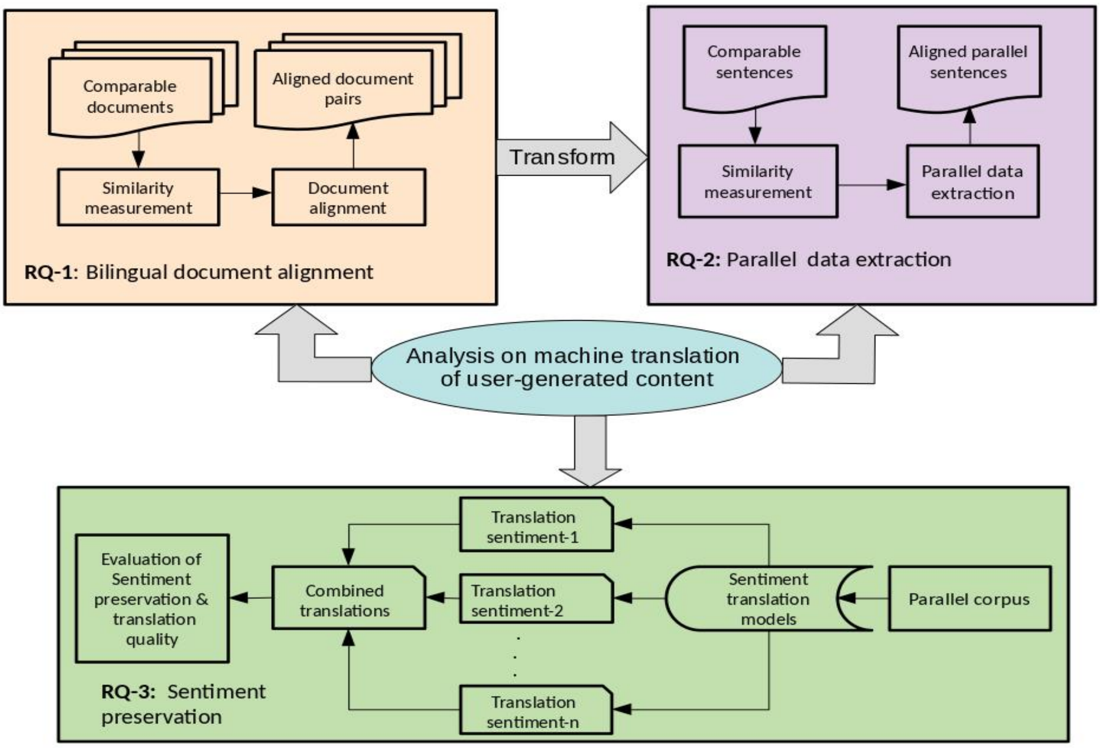


Figure 1.2: Data models underlying the research conducted in this thesis

- In order to address our first research goal, we develop a sophisticated bilingual document alignment system in order to find semantically equivalent documents in two different languages.

- To address our second research goal, we transform our document alignment system into an efficient parallel data extraction system in order to extract similar bilingual sentences. These extracted sentence pairs are treated as parallel corpora which can be considered as an additional training data for improved MT systems.

- We address our third research goal by building the first ever sentiment translation models based on sentiment classification with the aim of improving sentiment preservation during the translation of UGC.

1.6 Thesis outline

The outline of this thesis is organised in the following chapters.

- **Chapter 2: Background**

  In this chapter, we begin with the important concepts used in this thesis. We provide a brief introduction to MT and its most popular variants nowadays. Subsequently, we introduce UGC and its characteristics with several examples. Thereafter, we explain the document alignment system which takes comparable corpora and aligns the most probable source and target documents, some of which are candidates for inclusion in MT system training data. We then provide a detailed discussion on parallel data extraction techniques from comparable corpora. Afterwards, we discuss in detail the area of sentiment analysis and its types. Finally, we describe the concept of sentiment preservation system which is one of most recent applications of UGC translation.

- **Chapter 3: Machine translation of user generated content**

  We begin with our general experiments on UGC translation in this chapter. We use different types of UGC such as online posts, tweets, reviews etc., in addition to using the clean natural texts. Our focus is mainly on the major issues in UGC translation. We describe the experimental results and set three research goals from our findings in this chapter. The research goals are then dealt with in the subsequent chapters.

- **Chapter 4: Bilingual document alignment**

  This chapter describes the implementation of a bilingual document alignment system. We utilise the comparable corpora of a collection of documents in two different languages in order to extract similar document pairs.

- **Chapter 5: Parallel data extraction**

  In this chapter, we explain how we leverage our document alignment system in order to extract additional parallel data. We describe in detail the implementation of a parallel sentence extractor that attempts to find semantically equivalent sentences in two different languages from comparable corpora. Initially, we test the system on a comparable corpus of news texts. Afterwards, we conduct similar experiments on a comparable corpus of UGC.

- **Chapter 6: Sentiment preservation**

  This chapter discusses the implementation of a sentiment translation system. Our objective is to perform sentiment classification to classify the parallel UGC corpus and also the parallel corpus of natural texts into different sentiment categories. We then show how a suite of sentiment translation models can be built from each part of the *sentiment-classified* corpora. Finally, we compare the system performance between the *sentiment translation* system and the baseline translation system in terms of both translation quality and sentiment preservation.

- **Chapter 7: Conclusions and Future work**

  In the final chapter, we summarise our findings in this thesis. We also discuss the advantages and disadvantages of our approaches along with discussions on some unexplored methodologies. Finally, we point out future directions in this research field and some further possibilities of applying new approaches.

1.7 Publications

We have published some of our findings in this research in peer-reviewed journals, conferences and workshops, as follows:

Relevant publications

1. Afli, H., Aransa, W., Lohar, P., and Way, A. (2016a). From Arabic user-generated content to machine translation: integrating automatic error correction. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–14, Konya, Turkey

2. Lohar, P., Popovic, M., Afli, H., and Way, A. (2019a). A systematic comparison between SMT and NMT on translating user-generated content. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–11, La Rochelle, France

3. Lohar, P., Popović, M., and Way, A. (2019b). Building English-to-Serbian machine translation system for IMDb movie reviews. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 105–113, Florence, Italy

4. Lohar, P., Afli, H., Liu, C.-H., and Way, A. (2016a). The ADAPT bilingual document alignment system at WMT16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 717–723, Berlin, Germany

5. Lohar, P., Ganguly, D., Afli, H., Way, A., and Jones, G. J. (2016b). FaDA: Fast Document Aligner using Word Embedding. *The Prague Bulletin of Mathematical Linguistics*, 106(1):169–179

6. Lohar, P. and Way, A. (2020). Parallel data extraction using word embeddings. In *International Conference on NLP Techniques and Applications*, London, United Kingdom (Accepted)

7. Lohar, P., Afli, H., and Way, A. (2017a). Maintaining sentiment polarity in translation of user-generated content. *The Prague Bulletin of Mathematical Linguistics*, 108(1):73 – 84

8. Lohar, P., Afli, H., and Way, A. (2018a). Balancing translation quality and sentiment preservation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pages 81–88, Boston, MA, USA

9. Sluyter-Gäthje, H., Lohar, P., Afli, H., and Way, A. (2018). FooTweets: A bilingual parallel corpus of world cup tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1–5, Miyazaki, Japan

Other publications

1. Poncelas, A., Lohar, P., Way, A., and Hadley, J. (2020). The impact of indirect machine translation on sentiment classification. In *The 14th biennial conference of the Association for Machine Translation in the Americas*, Online

2. Simon, D., Castilho, S., Lohar, P., Afli, H., and Way, A. (2020). Transcasm: A bilingual corpus of sarcastic tweets. In *The 6th edition of "Using Corpora in Contrastive and Translation Studies Conference" (Accepted)*

3. Castilho, S., Moorkens, J., Gaspari, F., Sennrich, R., Sosoni, V., Georgakopoulou, Y., Lohar, P., Way, A., Valerio, A., Miceli Barone, A. V., and Gialama, M. (2017). A comparative quality evaluation of PBSMT and NMT using professional translators. In *Proceedings of the 16th Machine Translation Summit*, pages 116–131, Nagoya, Japan

4. Calixto, I., Stein, D., Matusov, E., Lohar, P., Castilho, S., and Way, A. (2017). Using images to improve machine-translating e-commerce product listings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 637–643, Valencia, Spain

5. Afli, H., Lohar, P., and Way, A. (2017a). MultiNews: A web collection of an aligned multimodal and multilingual corpus. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*, pages 11–15, Taipei, Taiwan

6. Lohar, P., Dutta Chowdhury, K., Afli, H., Hasanuzzaman, M., and Way, A. (2017c). ADAPT at IJCNLP-2017 task 4: A multinomial Naive Bayes Classification approach for customer feedback analysis task. In *Proceedings of The 8th International Joint Conference on Natural Language Processing, Shared Tasks*, pages 161–169, Taipei, Taiwan

Chapter 2

Background

In the previous chapter we provided a brief introduction to UGC translation and our research interests in this area. In this chapter we discuss MT and its most popular variants in Section 2.1. In Section 2.2, we describe in detail the different types and characteristics of UGC along with some examples. Furthermore, we explain the 'document alignment' process in Section 2.3. Subsequently, we discuss 'parallel data extraction' and approaches applied to accomplish this task in Section 2.4. Then, we provide an introduction to sentiment analysis in Section 2.5. Finally, we introduce the concept of 'sentiment preservation' in UGC translation in Section 2.6.

2.1 Machine translation

Machine translation (MT) is the automated process of translating a text from a source language to a target language. Some of the challenging aspects of MT are (i) large variety of languages, alphabets and grammars, (ii) translating a source-language text into the target language is harder for a computer than a human being, and (iii) there is no one correct translation.

Over the years since MT began to flourish, following major approaches in MT have emerged: (i) Statistical machine translation (SMT): 1990s - 2010s, (ii) Neural machine translation (NMT): 2014 onwards.

2.1.1 Statistical Machine Translation

In SMT, the probability for the translation is determined using Bayes Rule which is shown in Equation (2.1)

$$Pr(S|O) = \frac{Pr(S)Pr(O|S)}{Pr(O)} \tag{2.1}$$

In the above equation, (i) $P(S|O)$ is the probability of the state given the observation, (ii) $P(O|S)$ is the probability of the observation given the state, (iii) $P(S)$ is the probability of the state happening in general, and (iv) $P(O)$ is the probability of the observation happening in general. Let us consider that state is the English translation and the observation is the original French sentence. Since $P(O)$, the probability of the French sentence, is the same for every English translation (the state), and we are only interested in comparing the probabilities of different English translations, we need to consider only the $P(O|S)P(S)$.

Now, we can replace the variables of Equation (2.1) with those used in Brown et al. (1993), where S is f, for a French sentence, and O is e, for an English sentence. Accordingly, we arrive at the fundamental theorem of SMT as shown in Equation (2.2)

$$\bar{e} = \operatorname*{argmax}_{e} P(e|f) = \operatorname*{argmax}_{e} P(f|e)P(e) \tag{2.2}$$

This means that the best English translation $\bar{e}$ is the English sentence that maximizes the above equation.

SMT is thus composed of two main components as follows: (i) a language model, and (ii) a translation model

Language model: A statistical language model is a probability distribution over sequences of words. For example, let us consider a sequence of n words. The language model assigns a probability $P(w_1, ...w_n)$ to the whole sequence. In general, the probability is calculated using the chain rule as shown in Equation (2.3)

$$P(x_1, x_2, ...x_n) = P(x_1)P(x_2|x_1)...P(x_n|x_1, ..., x_{n-1}) \tag{2.3}$$

Now, the chain rule applied to compute the joint probability of words in a sequence is shown in Equation (2.4)

$$P(w_1, w_2, ...w_n) = \prod_i P(w_i|w_1 w_2 ...w_{i-1}) \tag{2.4}$$

Let us now consider a sequence of words in the form of a sentence "*Its water is so transparent*". Using Equation (2.4), we can calculate the probability of this sentence as shown in Equation (2.5)

$$\begin{aligned} P(\text{``}Its\ water\ is\ so\ transparent'') &= P(Its) * P(water|Its) \\ &* P(is|Its\ water) * P(so|Its\ water\ is) \\ &* P(transparent|Its\ water\ is\ so) \end{aligned} \tag{2.5}$$

- **N-gram Models:** From the Markov assumption,[1] we can formally define n-gram models where $k = n - 1$ as shown in the following equation.

$$P(w_i|w_1 w_2 ...w_{i-1}) \approx P(w_i|w_{i-n-1} ...w_{i-1}) \tag{2.6}$$

  The simplest versions of the n-gram models are defined as the *Unigram* model ($k = 1$) and the *Bigram* model ($k = 2$). However, Equation (2.6) can be extended to compute higher n-grams such as trigrams, 4-grams, 5-grams and so on. We use trigram models in this thesis.[2]

[1] The idea that a future event (i.e. the next word) can be predicted using a relatively short history (for example, one or two words) is called a Markov assumption.

[2] Note that we have not assessed the performance of word-level exact match for our UGC corpora. However, as this information may be revealing, we will compute it in our future work.

Translation model: The most used model in SMT is the phrase-based translation model (PBMT) that translates the whole sequences of words, where the length of the phrases may differ. The sequences of words are called phrases, but they are not typical linguistic phrases. We do not discuss the working details of a PBMT system in this section, but refer the readers to Koehn et al. (2003) for more details.

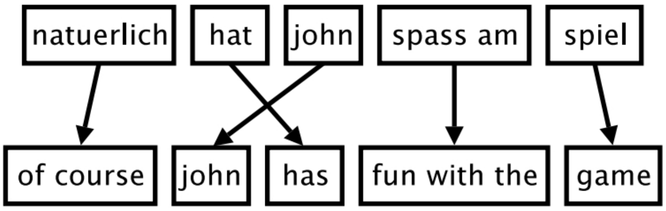


Figure 2.1: A translation example using PBMT system: (Koehn, 2009)

Figure 2.1 shows an example of German to English translation using the PBMT system.

The following are some examples of SMT systems. Note that all of these examples are based on the PBMT architecture.

- Google Translate (from 2006 to 2016, until they announced to change to NMT),
- Microsoft Translator (until 2016, when they changed to NMT),
- Moses: Open source toolkit for statistical machine translation (Koehn et al., 2007).

The following are some of the advantages and disadvantages of an SMT system.

Advantages

- It requires less manual work from linguistic experts,
- The translation probabilities are learned from the parallel and monolingual corpora automatically,
- The translation becomes more fluent with larger parallel corpora

Disadvantages

- It requires large bilingual parallel corpora,
- The specific errors can be hard to fix
- It is less suitable for language pairs with big differences in word order

2.1.2 Neural Machine Translation

NMT is the use of neural network to learn the translation model. It is an end-to-end system as only one model is required for the translation. NMT uses vector representations for words and internal states which means that words are transcribed into a vector defined by a unique magnitude and directionality. This framework is much simpler compared to the phrase-based models. Rather than using the separate components such as the language model and translation model, NMT uses a single sequential model that produces one word at a time.

NMT began to develop using pure sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014) and was improved upon using attention-based variants (Bahdanau et al., 2015; Luong et al., 2015). Since then, many open-source NMT implementations have also been released, including *OpenNMT*[3] (Klein et al., 2017), *Nematus*[4] (Sennrich et al., 2017), *GNMT*[5] (Wu et al., 2016), *fair-seq*[6] (Ott et al., 2019) and *Marian*[7] (Junczys-Dowmunt et al., 2018).

We use *OpenNMT* in this thesis. *OpenNMT* is a community of projects comprised of libraries for training, using, and deploying NMT models. The system was based originally on the attention-based sequence-to-sequence model, which was rewritten

[3] `https://github.com/OpenNMT/OpenNMT-py`

[4] `https://github.com/EdinburghNLP/nematus`

[5] `https://translate.google.com/`

[6] `https://github.com/pytorch/fairseq`

[7] `https://marian-nmt.github.io/`

to increase efficiency, readability, and generalisability. It was designed to be simple to use and easy to extend. Figure 2.2 illustrates the architecture of OpenNMT.

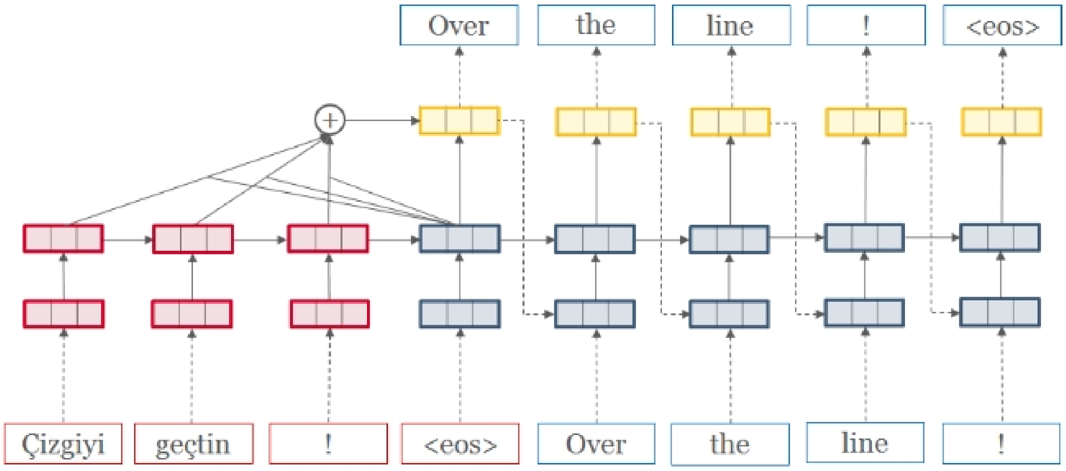


Figure 2.2: Architecture of OpenNMT (Klein et al., 2017):The red source words are first mapped to word vectors and then fed into a recurrent neural network (RNN).[8]Upon seeing the end-of-sentence <eos> symbol, the final time step initializes a target blue RNN. At each target time step, attention is applied over the source RNN and combined with the current hidden state to produce a prediction of the next word. This prediction is then fed back into the target RNN.

The following are some advantages as well as disadvantages of an NMT system.

Advantages

- It is a typical end-to-end model without any need for a pipeline of specific tasks,
- It performs better than SMT when trained on a significantly large data.

Disadvantages

- It usually does not perform well when trained on small amounts of data,
- NMT performs rather poorly for long sentences.

[8] Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states.

2.1.3 In-domain and out-of-domain in MT

In order to translate a text, it is always ideal to use an MT system from the same domain. For example, if we are interested in translating a text from the medical domain, we should preferably build an MT system that is specialised in translating medical texts. In general, an MT system is built from the parallel (training) data and tested on unseen (test) data from the same domain. Such an MT system is called an in-domain MT engine. In contrast, if there is a domain mismatch between the training and the test data set, we refer to it as an out-of-domain MT system. An example of such an MT system is a translation model which is trained on parallel texts from parliamentary proceedings (e.g. 'Europarl' corpus)[9] (Koehn, 2005) and is used to translate the texts from the sports domain.

2.2 User-generated content

UGC can be best described as any form of content, such as images, videos, text, and audio, that are posted by users on online platforms such as social media. Examples of UGC are as follows: (i) social media content, (ii) reviews and testimonials, (iii) blog posts, (iv) online video content, (v) Q&A forums etc.

Some of the most popular types of UGC can be illustrated in Figure 2.3. We briefly discuss some of these in the following paragraphs.

- **Social media content:** People share thousands of messages, posts, photos, videos etc. daily on social platforms, and a huge portion of those interact with some sort of brand, including hotels, destinations, tour operators etc. Any time someone posts a social message, whether it is a Tweet or an Instagram post, that is UGC.

[9] `http://www.statmt.org/europarl/`

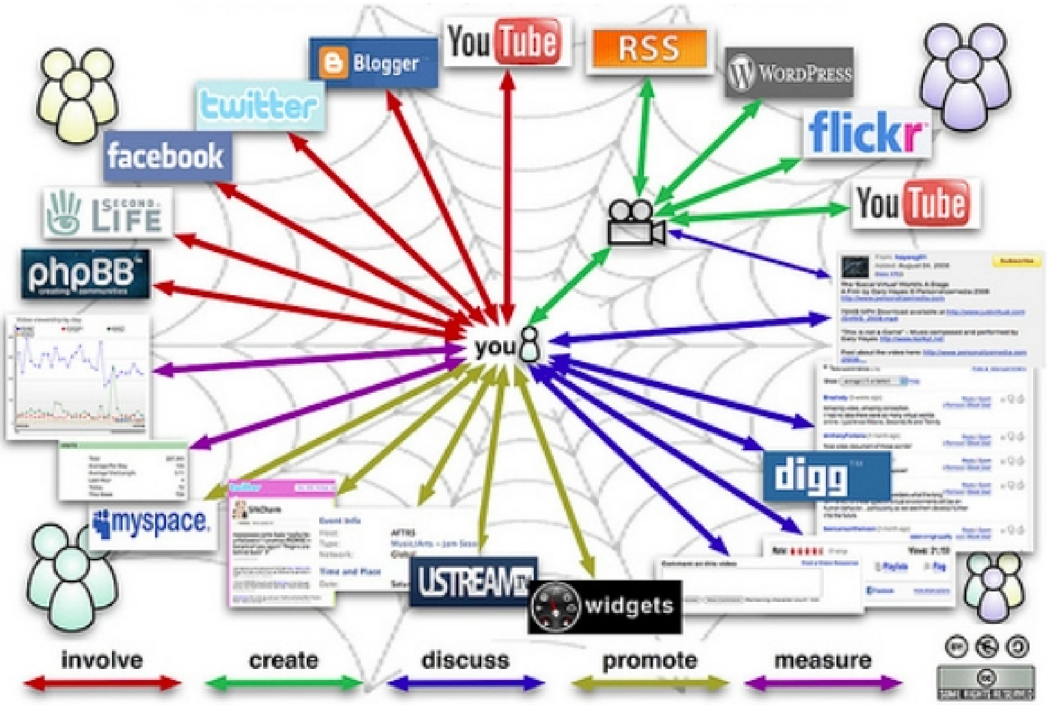


Figure 2.3: Different types of UGC: (Di Gangi and Wasko, 2009)

- **Reviews and testimonials:** Whether the customers write reviews on a section of a website, or whether they use third-party sites (such as Yelp, TripAdvisor, G2Crowd, Google, etc.), this kind of feedback is also UGC.

- **Blog post:** It is an entry (e.g. an article) which is written on a blog. It can include content in the form of text, photos, infographics, or videos.

- **Video content:** GoPro videos, Instagram stories, natively shot videos etc. qualify as UGC. This also includes AR lenses or filters or live video streams on Facebook, Instagram or other platforms.

- **Q&A forum:** Q&A (questions and answers) forums provide an avenue for community members to ask and answer questions. It allows members to (i) create new questions, (ii) add inline images, (iii) view and answer questions, (iv) search for a question, (v) help moderate the content, (vi) identify best answers and (vii) move Q&A questions from one page to another.

Although different modalities of UGC are available on the Internet such as audio, video, images and text, we deal only with textual content in this thesis.

Now, consider the following examples of UGC.

(i) *BT i dnt consider it as a penalty*

(ii) *I avoided watching this film for the longest time.*

(iii) *Fast service, homemade fries and very nice burgers, all reasonably priced.*

The above examples show different types of UGC. The first one is a tweet from the football domain while the second and the third examples are movie reviews and restaurant reviews, respectively. The tweet here is full of both grammar and spelling errors. The user wrote *BT* instead of *But*, the small letter *i* instead of the capital *I*, and *dnt* instead of *don't*.

In the second example of a movie review, the user writes a perfect sentence. This is an example of a clean text but such texts do not occur always in UGC. In many cases, the texts are informal and contain errors. For example, the length of a tweet in bounded by the 280-characters' limit and so the users often write short forms, use abbreviations, remove vowels from long words etc. In addition, there is no restriction that UGC must be grammatically correct and formal in nature. Many users thus deliberately deviate from linguistic norms and write informal messages, tweets, reviews etc.

The third example shows a restaurant review which is partially correct in terms of linguistic norms. It does not contain spelling error but it is missing a few verbs in the text. For example, this review should start with a phrase like '*It has*' and there should be the verb '*are*' between the words *all* and *reasonably.*

Such anomalies in UGC pose challenges in MT. In addition, parallel bilingual texts for UGC are very rare and very small. For these reasons, most MT systems are capable of producing decent translation outputs only for clean texts and fail to exhibit quality performance for noisy UGC such as two of the three examples shown above.

2.3 Document alignment

In NLP, document alignment is a method of obtaining crosslingual document pairs that are either translations or near translations of each other. The most common approaches to implement a document alignment system include edit-distance (Levenshtein, 1966) between linearized documents (Resnik and Smith, 2003), cosine distance of idf-weighted bigram vectors (Uszkoreit et al., 2010), or the probability of a probabilistic DOM-tree alignment model (Shi et al., 2006) etc.

A special variant of document alignment is bilingual document alignment where the documents in two different languages are aligned based on their semantic similarity. It is crucial to consider the following issues in the task of bilingual document alignment.

- **Translation overhead:** An MT-based document alignment system is an approach where the source-language documents are translated into the target language and then the document alignment is performed. Translation overhead can occur in such a system, especially for a large corpus where it takes a huge amount of time to translate all the source-language documents into the target language and then perform the alignment task in the target language.

- **Domain mismatch:** This problem can occur for an MT-based document alignment system. In some cases, comparable documents may belong to a particular domain that is different from the domain of the MT system to be used to translate the documents. Domain-specific MT models should be used to obtain a good quality document alignment system.

- **Comparison space:** The alignment task for a corpus with a small number of source- and target-language documents can be easily done in a reasonable amount of time because the number of possible pairings is not too high. However, for a corpus with a large number of bilingual documents, the number of comparisons may be prohibitively large. For example, if there are $10K$

English and $10K$ French documents in a comparable corpus, the number of comparisons becomes $100M$ which would take a huge amount of time if each comparison involves complex computation. It is important therefore, to avoid unnecessary comparisons in order to obtain the alignments in a reasonable amount of time.

- **Variable length:** The source-language and the target-language documents can differ in length. It is important not to consider the comparison of two documents whose lengths differ significantly.

2.4 Parallel data extraction

Parallel data extraction is the process of obtaining bilingual texts from a collection of bilingual documents. However, it can also be done at paragraph level. For example, Kúdela et al. (2017) apply bilingual word embeddings and locality-sensitive hashing to identify parallel segment pairs in a web domain. A typical parallel data extraction system is usually composed of the following main phases: (i) crawling web sites, (ii) document alignment, (iii) sentence alignment, and (iv) sentence pair filtering.

• **Crawling web sites:** Web crawling is an automated process to browse the World Wide Web in a methodical and automated manner. Many websites use web crawling to provide up-to-date data. For the task of parallel data extraction, there are a number of challenges in web crawling, such as (i) identifying web sites with multilingual content, (ii) avoiding crawling of web pages with identical textual content, and (iii) avoiding crawling of large web sites having content in different languages that is not parallel, etc. Some examples of web crawlers are *Httrack*,[10] *ILSP-FC*[11] (Papavassiliou et al., 2013), *RCrawler* (Khalil and Fakir, 2017), *APACHE NUTCH*,[12]

[10] `https://www.httrack.com/`

[11] `http://nlp.ilsp.gr/redmine/projects/ilsp-fc`

[12] `http://nutch.apache.org/downloads.html`

DEEPCRAWL[13] etc.

• **Document alignment:** As mentioned earlier, document alignment is a matching task that takes a pair of documents and computes a score to determine their likelihood of being translations of each others. Some examples of document alignment system are *YODA system* (Dara and Lin, 2016a), Bitextor[14] (Esplà-Gomis, 2009), *UFAL* (Le et al., 2016), ILSPFC[15] (Papavassiliou et al., 2016), *FaDA*[16] (Lohar et al., 2016b) etc.

• **Sentence alignment:** This process is very similar to document alignment except that it is done at sentence level, i.e. it calculates the likelihood of two sentences being translations of each other. Some of the popular tools for sentence alignment are *Hunalign*[17] (Varga et al., 2007), *Gargantua*[18] (Braune and Fraser, 2010), *Bleualign*[19] (Sennrich and Volk, 2010) etc.

• **Sentence pair filtering:** This process filters out bad sentence pairs that exist for at least one of the following reasons: (i) the web sites do not contain any parallel data, (ii) the failure of earlier processing steps, and (iii) sentence lengths differ hugely between the sentences under consideration.

2.5 Sentiment analysis

Sentiment analysis is the automated process of interpreting and classifying emotions such as positive, negative and neutral in a text using text analysis techniques. Liu (2012) gives an elaborated definition of sentiment analysis as follows.

[13] `https://www.deepcrawl.com/`

[14] `http://bitextor.sourceforge.net/`

[15] `http://nlp.ilsp.gr/redmine/projects/ilsp-fc`

[16] Available at `https://github.com/loharp/FaDA`

[17] `http://mokk.bme.hu/en/resources/hunalign/`

[18] `https://sourceforge.net/projects/gargantua/`

[19] `https://github.com/rsennrich/Bleualign`

Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. (Page 7: Liu (2012))

Sentiment analysis is one of the most explored research areas in NLP that involves UGC analysis. For example, Turney (2002) presents an unsupervised learning algorithm for rating a review as thumbs up or down. Their algorithm consists of three steps: (i) extracting phrases that contain adjectives or adverbs, (ii) estimating the semantic orientation of each phrase, and (iii) classifying the review based on the average semantic orientation of the phrases. Pang and Lee (2004) examine the relation between subjectivity detection and polarity classification. They show that subjectivity detection can compress reviews into much shorter extracts that still retain polarity information at a level comparable to that of the full review.

A set of techniques for mining and summarising product reviews based on data mining and NLP methods has been proposed in Hu and Liu (2004). They provide a feature-based summary of a large number of customer reviews of a product sold online. Kim and Hovy (2004) present a system that automatically finds people who hold opinions about a given topic and the sentiment of each opinion. Their system contains a module for determining word sentiment and another for combining sentiments within a sentence. He et al. (2019) propose an interactive multi-task learning network for jointly learning aspect and opinion term co-extraction, and aspect-level sentiment classification.

Sentiment analysis allows businesses to identify their customers' sentiment towards products, brands or services through online conversations and feedback. Figure 2.4

[20] Source: `https://monkeylearn.com/sentiment-analysis/`

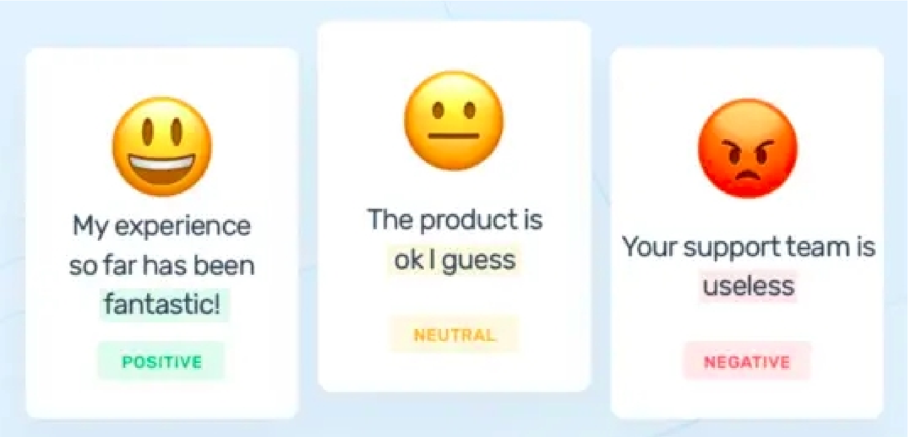

My experience so far has been fantastic!
POSITIVE
The product is ok I guess
NEUTRAL
Your support team is useless
NEGATIVE

Figure 2.4: Customers' sentiment[20]

illustrates the possibilities of a typical response from a customer towards a product, brand or service. We can see that there are three possible sentiment classes associated with the response: positive, neutral and negative. It is crucial for businesses to identify such sentiment in order to keep track of their customers' behaviour. Once these responses are tracked properly, the companies can act accordingly, either to continue their good service or to improve it, depending upon the responses.

Depending on the scale, sentiment analysis can be categorised into two types: (i) coarse-grained, and (ii) fine-grained.

- **Coarse-grained sentiment analysis:** This analysis type is done at the document and sentence levels. It entails the following two tasks.

  (i) **Subjectivity classification:** It is necessary to determine whether a sentence is objective or subjective. An objective sentence consists of facts about an object or a topic. An example is the following sentence.

  Three strangers are reunited by astonishing coincidence after being born identical triplets, separated at birth, and adopted by three different families.

  In contrast, a subjective sentence expresses someone's attitude towards a subject. An example can be the following sentence.

This apartment is wonderful. I enjoy every minute I spend in here.

(ii) **Sentiment detection and classification:** The objective of this task is to define whether a sentence contains a sentiment or not. If it does, this process determines whether the emotion is positive, negative, or neutral. The following examples show these emotions.

Positive: *One of the most surprising and satisfying movies of the year.*

Negative: *The fact that it's also clumsily made and rife with mediocre performances seems almost beside the point in the context of how pointless this thing is in the first place.*

Neutral: *I think everyone deserves a second chance expresses their subjective opinion.*

- **Fine-grained sentiment analysis:** This type of sentiment analysis refers to the detection of sentiment, not on the document level, but rather on the sentence, subsentence, or even aspectual level. Usually a sentence is broken into phrases or clauses and each part is analysed in connection with the others. It is useful for processing comparative expressions (e.g. *Samsung is way better than iPhone*) or short social media posts.

  Fine-grained sentiment analysis not only enable us to understand how people evaluate a product or service; it also identifies which feature or aspect are discussing (Lohar et al., 2017c). For example, consider the review "*A* touchpad *on my laptop stopped working after* 4 *months of use.*" This review does not address everything about the laptop. Instead, it shows that the customer is mainly concerned with its touchpad which she is not satisfied with.

2.6 Sentiment preservation in MT

Sentiment preservation is the technique that enables an MT system to maintain the sentiment polarity of the source-language text during the translation process. It is one of the most recent topics in MT-based sentiment analysis and has been active since 2017 (Lohar et al., 2017b, 2018b).

Ex.	Source	Reference translation	Google Translate output
1	عم انراسلكم على الصفحه ما حدا برد علينا	We messaged you on the webpage, no one has responded to us	I am sending you on the page
2	بشو بتفيد هي المنظمه ولشو تابعه	whats the benefit of the organization and it belongs to whom?	Bisho benefit is the organization and Bisho affiliate
3	ضلوا خبرونا بما هو مفيد.. دمتم بخير	Keep inform us about what is useful... Wish you all the best	Go astray with what is useful .. long as you are fine

Table 2.1: Sentiment alteration: as produced in September, 2019

Table 2.1 gives some examples of Arabic UGC and their translations using Google Translate.[21] The first example conveys negative sentiment but it is translated as a neutrally sentimented text. In the second example, the original text is neutral. The translation is not correct, as it is implied that *Bisho benefit* is the name of the organisation and it is affiliated to *Bisho*, which does not make sense; nonetheless, the whole output is categorised as neutrally sentimented text. In contrast, the translation of the third example is a bit more complicated. The original text is positive but its translation is neither positive nor negative. First of all, the translation is almost meaningless but we can still infer negative sentiment from the first part (*Go astray with what is useful*) because it seems like someone is advised to stray from a useful path, thus receiving wrong or negative advice. However, positive sentiment can be inferred from the second part (*long as you are fine*). These two sentiment classes cancel each other and the whole translation is collectively classified as neutral.

[21] https://translate.google.com/

We can see from the above outputs that Google Translate fails to preserve the sentiment of the original texts in two out of three cases. Such a phenomenon is harmful for MT-based sentiment analysis because sentiment alteration can create negative impact. For example, companies always want to keep track of their customers' behaviour and so they want the translations of their customers' feedback to maintain the sentiment, rather than mainly focusing on translation quality per se. If the sentiment alters, it would send wrong information to the company and they would not be able to keep track of the original sentiment associated with the feedback. Considering this situation, it is extremely important to implement an MT-based sentiment preservation system in such a scenario. In Chapter 6 of this thesis, we will discuss in detail how we leverage the utility of sentiment classification to build an MT-based sentiment preservation system in order to preserve sentiment classes during translation.

In this chapter, we provided background information on MT, UGC translation and other topics discussed in this thesis. In the next chapter, we will discuss our experiments on different types of UGC and set some research goals from our findings.

Chapter 3

Machine Translation of User Generated Content

In the previous chapter, we introduced MT and UGC translation along with their relevant topics. In this chapter, we provide a detailed description of our work in MT of UGC in general and set some research goals from our findings.

UGC is usually informal and noisy in terms of linguistic norms. In general, the noise in UGC does not create big problems for humans to understand the content, but it can pose challenges for several NLP applications such as parsing, sentiment analysis, MT, etc. An additional challenge for MT is the sparseness of bilingual (translated) parallel UGC corpora. Although PBMT can largely manage the training of a translation model using a small parallel corpus, NMT usually finds it difficult as it generally requires a lot more data. However, Sennrich and Zhang (2019) show that an optimized NMT system can outperform PBMT with far less data than previously claimed.

In this chapter, we explore general issues in UGC translation. We build translation systems using the parallel resources available for UGC and then evaluate translation quality. Finally, we set three research goals from some of our experimental findings.

3.1 Related Work

A considerable amount of work has been done in the area of UGC translation. For example, Jehl et al. (2012) translate microblog messages from Twitter by using a translation-based CLIR system. Some researchers have attempted to build parallel resources for UGC, since the lack of large parallel corpora represents one of the major challenges for UGC translation. For example, Ling et al. (2013) crawl a considerable amount of parallel sentence pairs from micro-blogs. They extract more than 1 million Chinese–English parallel segments from 'Sina Weibo' (the Chinese counterpart of Twitter) using only their public APIs.

Automatic collection and crowd-sourcing approaches to build a parallel corpus of Tweets such as *TweetMT* is described in Vicente et al. (2016). IR-based methods can also be employed in translating hashtags in Twitter (Carter et al., 2011). Banerjee et al. (2012) investigate domain adaptation and reduction of OOV words for English-to-German and English-to-French translation of web forum content. The estimation of comprehensibility and fidelity of machine-translated UGC from English to French is investigated in Rubino et al. (2013).

However, the scarcity of parallel training data has always become a major challenge in MT, especially for the NMT-based approach, as well as discrepancies between the training and test domains (Koehn and Knowles, 2017). Unlike PBMT, NMT is even more sensitive to low-resource settings and domain mismatch (Koehn and Knowles, 2017) than PBMT.

Our work in this research area includes (i) integrating spelling error correction for Arabic UGC translation, (ii) translating Twitter data and incorporating back-translation to extend the translation models, (iii) investigating the performance of PBMT and NMT systems for translating movie reviews, (iv) translating restaurant reviews, and (v) analysing sentiment preservation in UGC translation.

3.2 Data sets

Our experiments are comprised of five different data combinations. We describe each of them in detail in the following sections.

3.2.1 Arabic UGC

Experiment-1 (discussed in Section 3.4.1) involves Arabic UGC translation. The data set for this experiment consists of 1.3M Arabic UGC words that are obtained from the 'QALB' corpus (Zaghouani et al., 2014). The segments are manually corrected by the annotators, based on which the corrected modern standard Arabic (MSA) versions of the training, development and the test data are generated. The UGC texts and the MSA versions are then aligned at sentence level and are tokenised using the 'MADA' tokeniser (Habash and Rambow, 2005). Statistics of the corpus used for automatic error correction in this experiment is shown in Table 3.1.

Bitexts	**# UGC tokens**	**# Ref MSA tokens**
Training_raw	1.22M	1.31M
Training_MADA_tok	1.48M	1.56M
Dev_raw	64.6K	69.5K
Dev_MADA_tok	78.0K	83.4K
Test_raw	61.2K	65.9K
Test_MADA_tok	74.1K	79.4K

Table 3.1: Statistics of the training, development (dev) and the test data

3.2.2 Twitter data and short news texts

These data sets are used for experiment-2 and experiment-5 (discussed in Section 3.4.2 and Section 3.4.5, respectively). The in-domain data consists of the *FIFA*-2014 Twitter data[1] containing 4,000 English tweets from the football World Cup

[1] Available at: `https://github.com/HAfli/FooTweets_Corpus`

2014 and their manual translations into German. The out of domain data for this experiment is comprised of the following: (i) *Twitter Harvard data*,[2] containing the English tweets collected by crawling Twitter's REST API using the Python library *tweepy 3*.[3] This data is collected by extracting tweets from the 20 most popular Twitter users (with the most followers) such as *Katy Perry*, *Barack Obama*, etc; and (ii) short news texts (*News*), containing around 216K short segments taken from the 'News-Commentary' parallel corpus.[4] These short text segments consist of up to 32 words (the reason is explained in Section 3.4.2.2). The statistics of these data sets is shown in Table 3.2.

Dataset	**#Total segments**	**#Training**	**#Dev**	**#Test**
FIFA-2014	4000	$3,000$	500	500
Harvard	$52,542$	$52,542$	/	/
News	$216,742$	$216,742$	/	/

Table 3.2: Statistics of the in-domain and out-of-domain data for experiment-2

Note that we held out the development and test data from *FIFA-2014* data set only, not from the *Harvard* and *News* data sets. For this reason those two column entries are empty and replaced by the '/' character which implies 'not applicable' in this case.

3.2.3 IMDB reviews and SEtimes news data

These data sets are used in experiment-3 (discussed in Section 3.4.3). The in-domain data in this experiment consists of the publicly available 'Large Movie Review Dataset'[5](Maas et al., 2011) containing $50,000$ IMDb user movie reviews in English which is mainly created for sentiment analysis research, so each review is associated with its binary sentiment polarity label 'positive' or 'negative'. Negative reviews have a score $\leq$4, positive reviews have a score $\geq$7 out of 10, and reviews

[2] Available at: `https://dataverse.harvard.edu/file.xhtml?persistentId=doi:10.7910/DVN/JBXKFD/F4FULO&version=2.2`

[3] `https://github.com/felHR85/Tweepy-3`

[4] `http://data.statmt.org/wmt16/translation-task/training-parallel-nc-v11.tgz`

[5] `http://ai.stanford.edu/~amaas/data/sentiment/`

with neutral sentiment are not included. The overall distribution of labels is balanced, i.e. $25K$ positive and $25K$ negative reviews. In the entire collection, up to 30 reviews are allowed for any particular movie. We keep 200 reviews (100 positive and 100 negative) consisting of about $2,500$ sentences for testing purposes, and use the remaining $49,800$ reviews (about $500K$ sentences) for training. Serbian reference translations are available for 33 test reviews (17 negative and 16 positive) containing 485 sentences (208 negative and 277 positive).

The out-of-domain data set consists of the South-East European Times (*SEtimes*) news corpus (Alperen et al., 2010) containing about $200K$ parallel sentences from news articles. In order to be able to compare the results with the in-domain scenario, the development set is taken from the *SEtimes* corpus, too. Table 3.3 shows the data statistics of IMDb and *SEtimes* data. Note that the *SEtimes* data do not contain development and test segments and so we place the '/' character in those columns.

Data set	**# Training segments**	**# Dev segments**	**# Test segments**
IMDb	$536,433$	500	485
SEtimes	$224,167$	/	/

Table 3.3: Statistics of the IMDb and *SEtimes* data

3.2.4 FourSquare and hotel review corpus

These data sets are used in Experiment-4 (discussed in Section 3.4.4). The French–English parallel corpus of Foursquare restaurant reviews[6] (Berard et al., 2019) contains over $11K$ reviews (or $18K$ sentences). The reviews were originally written in French, and then translated into English by professional translators. The authors also provide the official training, development and test splits for this data set.

[6] https://europe.naverlabs.com/research/natural-language-processing/machine-translation-of-restaurant-reviews/

The *Hotel_ Review* corpus[7] consists of $878K$ reviews from $4,333$ hotels crawled from *TripAdvisor*. Although most of the reviews are in English, some are also written in French. The statistics of the *FourSquare* and the *Hotel_ Review* data sets are shown in Table 3.4.

Data set	# **Reviews**	# **Parallel sentences**	# **training**	# **Dev**	# **Test**
FourSquare	$11,551$	$17,945$	$14,864$	$1,243$	$1,838$
Hotel_Review	$878,561$	/	/	/	/

Table 3.4: Statistics of the FourSquare parallel and the Hotel review data sets

In Table 3.4 the number of total parallel sentences, training, development and test data distributions for the *Hotel_ Review* corpus are not shown because it is not a parallel corpus and thus replaced by the '/' characters.

3.2.5 Flickr, News commentary and Arabic social media data

These data sets are used in experiment-5 (discussed in Section 3.4.5). The original Flickr data was composed of $\sim 30K$ pictures from Flickr, one description in English and one human translation of the English description into German. We use the textual part of the original data in our experiments. The 'New commentary' data[8] (in short 'news' data) consists of news texts from different domains. This statistics of the Flickr and News data sets is shown in Table 3.5. The Flickr and the *News* data are much larger than the Twitter data and so it would be a much more time-consuming process to annotate the sentiment labels in these data sets. We therefore apply an automatic sentiment analysis tool developed by Afli et al. (2017b) to assign sentiment scores to these data sets.

Data	**#Negative**	**#Neutral**	**#Positive**
Flickr	$9,677$	$11,065$	$8,258$
News	$111,337$	$14,306$	$113,200$

Table 3.5: Statistics of the Flickr and News commentary data set

Note that the size of the *News* data in Table 3.5 differs from that in Table 3.2

[7] https://www.cs.cmu.edu/~jiweil/html/hotel-review.html

[8] http://data.statmt.org/wmt16/translation-task/training-parallel-nc-v11.tgz

Sentiment	**#Training**	**#Dev**	**#Test**
Negative	770	50	50
Positive	514	50	50

Table 3.6: Data statistics of Arabic Social Media Posts

because the *News* data in Table 3.2 shows the total number of segments in the whole corpus (after filtering long sentences), whereas Table 3.5 shows the partition of the whole *News* data into three sentiment classes: negative, neutral and positive.

The Arabic social media data contains *Levantine* (one of the most popular dialects of Arabic) Arabic social media posts or comments manually translated into English. The data is collected by 'Translators without Borders'[9] (TWB) from their *Mercy Corps Khabrona* programme in Jordan connecting Syrian refugees to critical information and services.

Each of the posts and comments carries a certain degree of sentiment such as (i) expressing gratitude, (ii) criticising and (iii) being merely neutral questions. They also manually assigned sentiment scores to all the Levantine Arabic posts and their English translations in a similar way as the Twitter sentiment annotation discussed earlier. However, for this data set, we consider only two categories of sentiment classes– negative and positive– because TWB were interested only in posts that show whether the users are satisfied or not. We, therefore, divided the corpus into negative pairs (conveying negative sentiment) and positive pairs (conveying positive sentiment). The data distribution is shown in Table 3.6.

3.3 MT systems and evaluation

We used only PBMT systems in the initial stages of our experiments and included NMT systems afterwards. The later experiments are based on only the NMT approach because it has significantly improved recently and has replaced PBMT as the state-of-the-art in most cases.

[9] `https://translatorswithoutborders.org/`

- **PBMT Configuration:** We use the standard PBMT system trained by using the Moses toolkit (Koehn et al., 2007). The language models used are SRILM (Stolcke, 2002) and KenLM (Heafield, 2011). The word and phrase alignments are obtained using GIZA++ (Och and Ney, 2003). The maximum phrase length for training is set to 7. Finally, the models are tuned using minimum error rate training (MERT) (Och, 2003).

- **NMT Configuration:** We train sequence-to-sequence NMT models (Sutskever et al., 2014) based on recurrent neural networks with an attention mechanism (Luong et al., 2015). The NMT framework we use is the freely available open source NMT toolkit 'OpenNMT'[10] (Klein et al., 2017). We use the default parameter settings: RNN as the default type of encoder and decoder, $word\_vec\_size = 500$, $rnn\_size = 500$, $rnn\_type = LSTM$, $global\_attention\_function = softmax$, $save\_checkpoint\_steps = 5000$, $training\_steps = 100,000$ etc.

- **MT evaluation metrics:** The translations are evaluated by using BLEU-4 score (Papineni et al., 2002), smoothed BLEU (Lin and Och, 2004), TER (Snover et al., 2006) and chrF (Popović, 2015). Note that the BLEU and METEOR are precision based metrices, the higher the score the better the system. In contrast, TER is an error-based metric so the lower the score the better the system. The calculation of chrF score is based on character n-gram precision and recall enhanced with word n-grams. It measures the F-score averaged on all character and word n-grams, where the default character n-gram order is 6 and word n-gram order is 2.

[10] `https://github.com/OpenNMT/OpenNMT-py`

3.4 Experiments

3.4.1 Experiment-1: Automatic error correction for improving Arabic UGC translation

In this experiment, we investigate the utility of automatic error correction to improve the quality of Arabic UGC and its translation. Arabic UGC has different challenges in informal Arabic language processing compared to MSA texts because the majority of NLP tools for Arabic language are designed for MSA, while most of the online Arabic users use dialectal Arabic (DA) and the informal style. We use a DA-to-MSA normalisation and an MSA correction system based on the statistical approach. We investigate the usefulness of Arabic tokenisation for improving error detection.

3.4.1.1 Challenges

One of the challenges in Arabic UGC translation is shown in Table 3.7.[11] We can see that some of the OOV words in translation are spelling errors of MSA words such as قنات instead of قناة (which means *channel*) in the second example, or a dialectal word such as the Levantine (Syrian/Lebanese) word هيك converted to هكذا (which means *that*, *like*, *that kind* or *such*) in MSA. These challenges come from the fact that the online users linguistically switch between MSA and DA, either in the course of a single sentence or across the different sentences (i.e. code-switching).

The types of challenges in Arabic UGC can be summarised as follows:

- **OOV problem:** The words do not exist in the MSA dictionary, e.g. قنات instead of قناة (which means *channel*).

[11] Note: Although the romanisation of the Arabic examples would be very useful in Table 3.7, the content is taken from an already published work of (Afli et al., 2016a) and so it cannot be modified.

Arabic: وهيك معارضة بدها هيك تعامل امني .
MT output: and هيك opposition needs such behaviour
Ref Arabic: وهكذا معارضة يلزمها هكذا تعامل امني .
Ref Translation: and that opposition needs that kind of security dealings.
Arabic: اللي العاملين في قنات الجزيرة .
MT output: and اللي working for dقنات Al-Jazeera.
Ref Arabic: إلى العاملين في قناة الجزيرة .
Ref Translation: for those who are working in Al-Jazeera channel.

Table 3.7: Some examples of wrong translations. The first example is from Levantine dialect (without a spelling error) mixed with MSA words. The source side of “security dealings” is present in the Arabic text but the MT system did not produce its translation. The second example shows an MSA sentence with spelling errors.

It happens because of spelling mistakes or due to the fact that the word is a dialectal word.

- **Segmentation:** Extra space errors divide the word and generate a segmentation problem, e.g. قن اة instead of قناة.
- **Punctuation:** Punctuation like *commas* or *fullstops* are in the wrong place or missing altogether.
- **Character format:** Some Farsi/Urdu characters are used such as ڨ instead of ق, or ڢ instead of ف in Tunisian, Algerian and Moroccan dialects.
- **Word sense ambiguity:** Some words are common between DA and MSA but convey different meanings. For example, بقي in the Egyptian dialect means *become* whereas it means *remain* in MSA, which is أصبح.

3.4.1.2 Statistical conversion system

We use a conversion system trained on UGC texts that have been post-edited, manually corrected and normalised to MSA. This approach is considered as a UGC-to-MSA MT system with spelling error correction. The MT system handles the conversion process as the transformation of a sequence of words in a specific language into another sequence of words in the same language.

Now, let us consider a sequence of Arabic words in informal form $s^M = s_1 ... s_M$ of size M which is to be translated into a corrected MSA sentence $t^N = t_1 ... t_N$ of size N in the same language. The statistical approach determines the translation t^* which maximizes the posterior probability given the source sentence. Using Bayes' rule, we can show the formula in Equation (3.1).

$$t^* = \arg\max_t P(t|s) = \arg\max_t P(s|t)P(t) \tag{3.1}$$

The above equation shows that the whole conversion system is decomposed into a language model probability $P(t)$ and a translation model probability $P(s|t)$. The language model is trained on a large quantity of MSA corrected data and the translation model is built from bilingual texts aligned at sentence (segment) level, e.g. a UGC for a segment and its ground-truth in MSA obtained by manual annotation.

3.4.1.3 Arabic tokenisation

Arabic words are often ambiguous from a morphological perspective, which is due to its rich system of affixation and clitics and the omission of disambiguating short vowels and other orthographic diacritics in standard orthography (Habash and Rambow, 2005).

An example and its MSA correction with and without tokenisation is shown in Table 3.8.

UGC text without tokenisation
بعد إللي شفته وسمعته أنا جمال باشا السفاح أصدرت فرماني بنصب ٦١ مشنقة في ساحة المرجة ليشنق عليها أعمدة النظام السوري وذلك بعد العيد مباشرة لا محاكم ولا لاهاي ولا تضييع وقت .
Reference MSA text without tokenisation
بعد الذي رأيته وسمعته ، أنا جمال باشا السفاح ، أصدرت فرماني بنصب 16 مشنقة في ساحة المرجة ليشنق عليها أعمدة النظام السوري ، وذلك بعد العيد مباشرة ، لا محاكم ، ولا لاهاي ، ولا تضييع وقت .
UGC text with MADA tokenisation
بعد اللي شفت +ه و+ سمعت +ه انا جمال باشا السفاح اصدرت ف+ رما +ني ب+ نصب 16 مشنقة في ساحة المرجة ل+ يشنق علي +ها اعمدة النظام السوري و+ ذلك بعد العيد مباشرة لا محاكم و+ لا لاهاي و+ لا تضييع وقت .
Reference MSA text with MADA tokenisation
بعد الذي رايت +ه و+ سمعت +ه ، انا جمال باشا السفاح ، اصدرت ف+ رما +ني ب+ نصب 16 مشنقة في ساحة المرجة ل+ يشنق علي +ها اعمدة النظام السوري ، و+ ذلك بعد العيد مباشرة ، لا محاكم ، و+ لا لاهاي ، و+ لا تضييع وقت .
UGC in Buckwalter spelling
bEd <lly $fth wsmEth >nA jmAl bA$A AlsfAH >Sdrt frmAny bnSb ٦١ m$nqp fy sAHp Almrjp ly$nq ElyhA >Emdp AlnZAm Alswry w*lk bEd AlEyd mbA$rp lA mHA km wlA lAhAy wlA tDyyE wqt .
Reference MSA text in Buckwalter spelling
bEd Al*y r>yth wsmEth , >nA jmAl bA$A AlsfAH , >Sdrt frmAny bnSb 16 m$nqp fy sAHp Almrjp ly$nq ElyhA >Emdp AlnZAm Alswry , w*lk bEd AlEyd mbA$rp , lA mHAkm , wlA lAhAy , wlA tDyyE wqt .

Table 3.8: An example and its MSA correction with and without tokenisation

The source and the references are also presented in Latin letters using the 'Buckwalter code' (Buckwalter, 2002).[12]

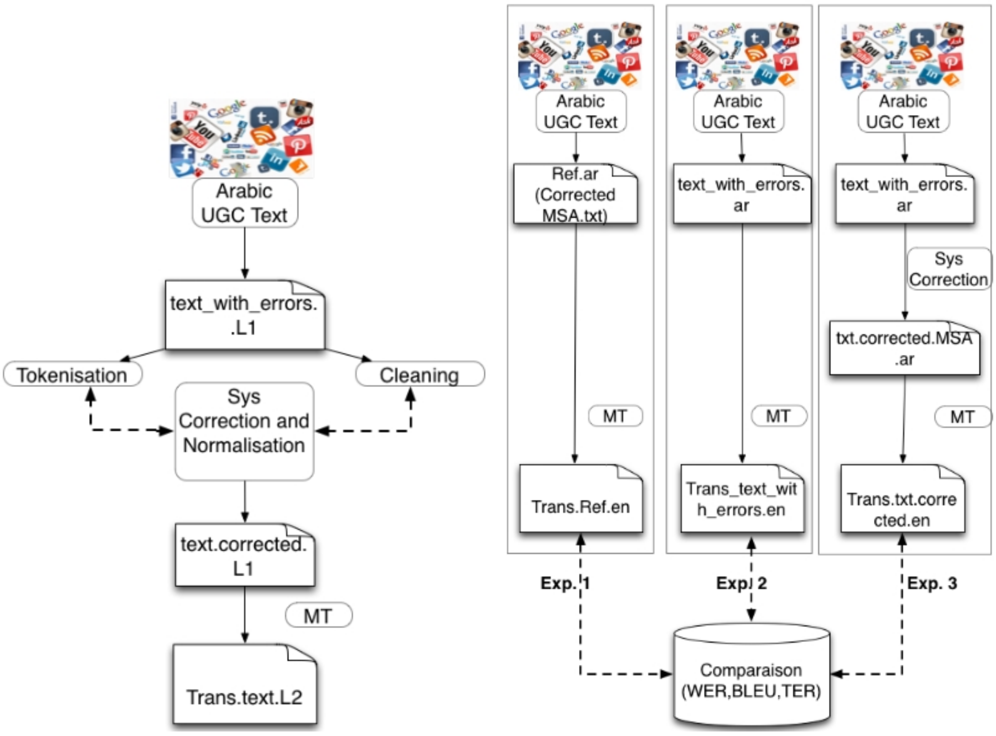
[12] While the experienced reader may view this output as erroneous, it is in fact the actual output from the Buckwalter code (Buckwalter, 2002) at the time of this work (Afli et al., 2016a) in 2016. If we were instead to use more recently developed Arabic tokenisers, these issues may be overcome, but we have not had a chance to verify this.

3.4.1.4 System description

It is assumed that the morphological complexities in Arabic affect the detection of spelling errors. Considering this situation, we propose to use two different systems in order to verify this assumption:

(i) *System 1* (Sys1): trained using the cleaned data without any tokenisation.

(ii) *System 2*(Sys1): trained using the MADA tokenisation (Habash and Rambow, 2005). The architecture of our proposed system is shown in Figure 3.1.



(a) UGC-to-MT framework

(b) Overall experimental architecture

Figure 3.1: Our proposed system

The whole system is divided into two parts; (i) the UGC-to-MT framework, and (ii) the overall experimental architecture.

- **UGC-to-MT framework:** The UGC-to-MT framework works in three steps as follows: (i) automatic tokenisation and cleaning, (ii) error correction (Sys Correction), and (iii) MT. Firstly, the original documents in language L1 (Arabic UGC in our case) are cleaned and an automatic tokenisation is generated. Secondly, the texts are corrected by the statistical conversion method.

- **Overall experimental architecture:** We conduct three different types of experiments which are referred to as *Exp-1*, *Exp-2* and *Exp-3*, respectively. Firstly, in *Exp-1*, we use the MSA reference (*Ref.ar*) as input to the MT system. This is the most favourable condition because it simulates the case where the error correction systems do not commit any error. We consider this as the reference during the automatic evaluation process. Secondly, in *Exp-2* we use the UGC text (*text_with_errors.ar*) directly as input to the MT system without any correction. We consider *Exp-2* as the baseline because the translation of UGC texts with errors is done by the standard PBMT system. Finally, *Exp-3* represents the complete proposed translation framework.

3.4.1.5 Results

- **Automatic error correction:** We evaluate the effectiveness of error correction by using word error rate (WER) which is derived from the Levenshtein distance (Levenshtein, 1966). We compare results produced by different systems against the baseline results which represent the scores between the original Arabic UGC text and the corrected MSA reference (called UGC-Baseline). We report the results in Table 3.9 in terms of the percentage of correctness, accuracy and WER of different system outputs. The correctness is the precision and the accuracy is simply the percentage of accuracy in this case. The 'UGC-Baseline' in the upper half of the table does not involve any tokenisation whereas the lower half does.

Systems	**Correctness ↑**	**Accuracy ↑**	**WER ↓**
UGC-Baseline (untokenised)	71.79	70.38	29.62
Sys1	86.44	82.48	17.52
UGC-Baseline (tokenised)	70.61	55.99	44.01
Sys2	84.05	77.92	22.08

Table 3.9: Results with Sys1 and Sys2 compared to the UGC-Baseline

We can observe in Table 3.9 that the two models trained with the same method with or without tokenisation (*Sys1* and *Sys2*) are capable of increasing the correctness, accuracy and reducing the WER.

• **Translation evaluation:** The translations are evaluated by using BLEU-4 score, smoothed BLEU and TER between the output of *Exp-1* (the reference) and *Exp-2* (the baseline) or *Exp-3* (our proposed framework). Table 3.10 illustrates the results of the two translation outputs from *Exp-2* and *Exp-3* compared to the outputs of *Exp-1*. The results show that our proposed framework is able to correct the final translation of the Arabic UGC text to some extent. Our best system (*Sys2*) increases by around 4 points in BLEU-4 and 3 points in smoothed BLEU scores. Furthermore, the TER score decreases by 1.72 points.

Systems	**BLEU-4 ↑**	**Smoothed BLEU ↑**	**TER ↓**
Exp-2	64.41	64.42	23.17
Exp-3 Sys1	67.30	66.53	21.94
Exp-3 Sys2	**68.31**	**67.42**	**21.45**

Table 3.10: BLEU-4, smoothed BLEU and TER scores on the test translated UGC-data corrected by Sys1 and Sys2

• **System comparison:** In order to analyse the degree of the agreement between the different systems, we transform the Sys1 outputs to the same tokenisation of Sys2 and score all of them comparing to the MSA correction reference transformed in MADA tokenisation (of Sys2), and using the WER metric.

Systems	**Correctness ↑**	**Accuracy ↑**	**WER ↓**
UGC-Baseline	67.24	59.54	40.46
Sys1	70.31	63.18	36.82
Sys2	**74.81**	**68.68**	**31.32**

Table 3.11: WER, accuracy and correctness on the test UGC-corrected data

The results are shown in Table 3.11. We notice that the best scores are obtained with *Sys2* using the MADA tokenisation. It is able to decrease 9.14% of the UGC word errors, which is a 22.59% relative improvement.

3.4.2 Experiment-2: PBMT vs NMT for translating tweets

In this experiment, we perform a systematic comparison between the PBMT and NMT on translating tweets.

3.4.2.1 System description

We use the *FIFA-2014*, *News* and the *Harvard* data sets and our translation direction is from German-to-English in this experiment. However, the *Harvard* tweets are available only in English, so we translate them into German by an English-to-German MT system (trained on the combination of the parallel Twitter and the *News* data sets) in order to create a synthetic parallel data. This process is called 'back-translation', because the monolingual data is normally written in the target language and then translated into the source language. We obtain about $50K$ additional parallel segments using this back-translation process.

3.4.2.2 Data combination

As we mentioned earlier in Section 3.2.2, $4,000$ tweet pairs is usually very small for MT training, so we held out only small amounts of data for development and testing purposes and keep the rest for training. We use $3,000$ segments for training, 500 for development and 500 for testing, respectively. The additional data consists of the following.

***Harvard* data:** This data set consists of $52K$ English tweets from the 20 most popular Twitter users (as explained earlier in Section 3.2.2).

***News*:** The out-of-domain *News* data consists of about $216K$ short segments. These short text segments contain up to 32 words. To find this length, we examine all the tweets in *FIFA-2014* data set and found that the lengthiest tweet consists of 32 words. Accordingly, we consider only those news segments that contain up to 32 words.

Table 3.12 shows the statistics of each data set, whereas the data combinations used for training the PBMT and NMT systems are shown in Table 3.13.

Data set	**#Segments**
Twitter football (in-domain)	$3,000$
Development	500
Test	500
Twitter Harvard (out of domain)	$52,542$
News (out of domain)	$216,742$

Table 3.12: Data statistics

Training Data	**#Segments**
Twitter_WC	$3,000$
Twitter_WC+Harvard	$55,542$
Twitter_WC+Harvard + News	$272,284$

Table 3.13: Data combinations used for PBMT and NMT training

3.4.2.3 System architecture

We built two MT models (one PBMT and one NMT) for each training data combination.

As a result, a total of 6 different MT models are built using all the data sets in this experiment. The test data is translated by each of them and so 6 different translation outputs are generated. Figure 3.2 illustrates the architectural diagram of the whole system. Note that 'T-PBMT' refers to the PBMT system trained on the Twitter corpus, 'TH-PBMT' refers to the PBMT system trained on the concatenation of the Twitter and the 'Harvard' corpus and so on.

3.4.2.4 Results

We evaluate the translation quality using three widely spread automatic measures: BLEU, METEOR and TER. The results are shown in Table 3.14.

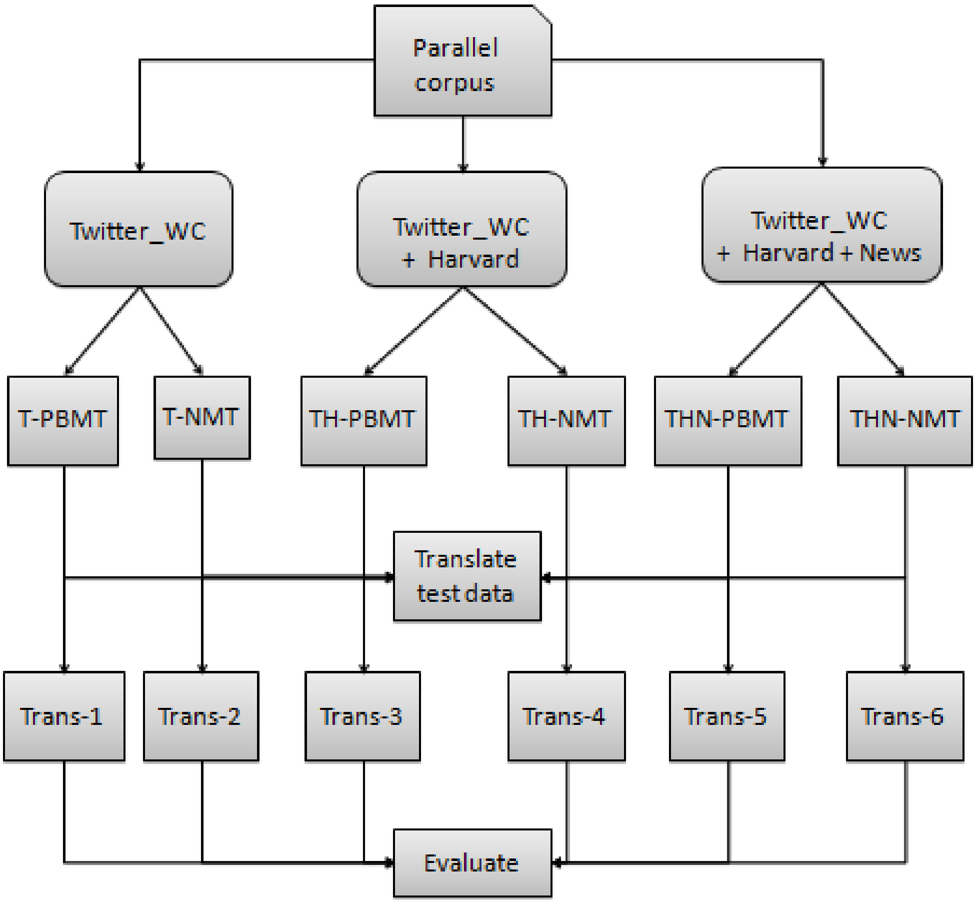


Figure 3.2: Illustration of the training of three PBMT and three NMT systems using three different parallel training corpora

Data set	MT system	BLEU ↑	METEOR ↑	TER ↓
Twitter_WC	PBMT	46.6	39.1	33.5
	NMT	0.8	6.8	88.4
Twitter_WC+Harvard	PBMT	48.6	41.4	30.9
	NMT	45.0	38.8	34.7
Twitter_WC+Harvard + News	PBMT	**50.0**	**42.2**	29.9
	NMT	**50.0**	41.9	**29.6**

Table 3.14: BLEU, METEOR and TER scores for each of the 6 MT outputs generated by two MT approaches and three training sets

As PBMT and NMT are the standard approaches used for translation, it would be normal to consider the models built from the 'Twitter_WC' data set as the baselines. However, as the Twitter data is too small to be used for training an NMT model, we consider the PBMT model as the baseline. Note that, the *Harvard* and *News* data sets are used as additional data to build the extended models in order to see if any improvement can be seen in the system performance.

We note the following observations from Table 3.14:

- **PBMT with different training data:** (i) training on the extremely scarce in-domain corpus already causes decent scores to be obtained, and (ii) all scores moderately improve with an increase in the size of the training corpus.

- **NMT with different training data:** (i) training of the NMT system with a scarce in-domain corpus is practically useless, (ii) adding back-translated tweets results in a huge improvement in the performance of the system so that it almost reaches the performance of the PBMT system training on the scarce corpus only, adding *News* data shows further improvement and the BLEU score becomes the same as for the PBMT system trained on the same full corpus, whereas the METEOR scores shows the PBMT system to be a little better, while the opposite is true for TER.

- **PBMT vs. NMT in different settings:** (i) the performance of the PBMT system is better for very scarce training data, (ii) the more training data is used (even though being back-translated, out-of-domain), the closer is the performance of the two MT approaches.

3.4.3 Experiment-3: Translating movie reviews

In this experiment, we deal with the challenges of building an MT system for UGC involving Serbian, a morpho-syntactically complex South Slavic language. We translate English IMDb user movie reviews into Serbian in a low-resource scenario. We attempt to answer the following research questions in this experiment:

- *What performance can be expected of an English-to-Serbian MT system trained on news articles and applied to movie reviews?*

- *Can the performance be improved by translating the monolingual English movie reviews into Serbian thus creating additional synthetic in-domain bilingual data?*

- *What are the main issues and the most important directions for the next experiments?*

In order to answer the above research questions, we built an NMT system on the publicly available clean out-of-domain 'News-Commentary' corpus (discussed in Section 3.2.3; unlike in experiment-2, we use all the news segments), and a PBMT system trained on the same data in order to compare the two approaches in this specific scenario. Subsequently, we use these two systems to generate synthetic Serbian movie reviews in order to create an additional in-domain bilingual dataset. Afterwards, we compare five different set-ups in terms of corpus statistics, overall automatic scores, and error analysis.

3.4.3.1 Data description and synthetic data creation

In terms of NLP resources, Serbian is generally not very well supported. The English–Serbian publicly available parallel OPUS data[13] contains mostly subtitles, which are rather noisy. In contrast, the only really clean parallel corpus is *SEtimes*, which is the reason why we use it for the baseline system in our experiments.

To the best of our knowledge, there are no publicly available parallel corpora for UGC in Serbian. Accordingly, we create a synthetic IMDb parallel corpus by translating English IMDb movie reviews into Serbian using our baseline system. This technique of synthetic data creation (Burlot and Yvon, 2018) has become very helpful for NMT systems (Sennrich et al., 2016a; Poncelas et al., 2018). As the synthetic data is created using back-translation, it consists of noisy source and clean target-language texts. However, in our case, we are interested in translating the data

[13] http://opus.nlpl.eu/

into Serbian, but we do not have any movie reviews in Serbian, only in English, which is the source language in our experiment. For this reason, we actually applied the 'forward-translation' technique. Unlike back-translation, forward-translation is the process of translating the monolingual data in the source language in order to create synthetic data in the target language for MT. For example, Bertoldi and Federico (2009) propose an approach of synthesising a bilingual corpus by translating the monolingual adaptation data into the counterpart language. However, forward-translation is less useful than back-translation (Park et al., 2017; Burlot and Yvon, 2018).

Serbian is a morphologically rich language and has a free word order. Moreover, it is bi-alphabetical (with both Latin and Cyrillic scripts) so attention should be paid in order not to mix the two scripts in one corpus. Another possible inconsistency in the corpus is the different ways of handling person names. In Cyrillic, only transcription is possible, whereas in Latin, both transcription as well as leaving the original are allowed. In addition, all person names are declined, as in other Slavic languages.

Recently, a comparison between the NMT and PBMT back-translation approach (Burlot and Yvon, 2018) has shown that using a PBMT system for synthetic data can lead to comparable improvement of the baseline NMT system with a lower training cost. Therefore, we are interested in using and comparing both approaches in order to improve our baseline NMT system.

3.4.3.2 Experiments

In Serbian, two alphabets are used: Cyrillic and Latin. All the data sets used in our experiments are in Latin only. Our English-to-Serbian MT systems are built in the following way:

- An out-of-domain PBMT system is trained on the *SEtimes* corpus.

- A baseline out-of-domain NMT system is trained on the *SEtimes* corpus.

- The English IMDb training corpus is translated into Serbian using the PBMT system, thus generating a synthetic parallel corpus (referred to as IMDb$_{pbmt}$).

- The English IMDb training corpus is translated into Serbian using the baseline NMT system, thus generating another synthetic parallel corpus (referred to as IMDb$_{nmt}$).

- A new NMT system is trained on the *SEtimes* corpus enriched with the IMDb$_{pbmt}$ corpus.

- Another NMT system is trained using the *SEtimes* corpus enriched with the IMDb$_{nmt}$ corpus.

- One more NMT system is trained using the *SEtimes* corpus enriched with both IMDb$_{pbmt}$ and IMDb$_{nmt}$ (referred to as IMDb$_{joint}$).

training	**reviews**	**segments**	**words (en)**	**voc (en)**	**words (sr)**	**voc (sr)**
SEtimes (natural)	/	224,167	4,675,549	81,064	4,439,280	155,447
IMDb (natural)	49,800	536,433	11,313,315	223,972	/	/
IMDb$_{pbmt}$	49,800	536,433	/	/	12,012,734	236,272
IMDb$_{nmt}$	49,800	536,433	/	/	11,077,566	195,912

dev (SEtimes)	/	1,000	20,338	4,757	19,244	6,806
OOV rate [%]		SEtimes	0.25	5.6	0.48	7.9
		IMDb	1.29	19.9	/	/
		IMDb$_{pbmt}$	/	/	2.21	29.0
		IMDb$_{nmt}$	/	/	2.18	29.0

test (IMDb)	33	485	8,530	2,548	7,630	3,220
OOV rate [%]		SEtimes	1.16	17.5	1.83	22.2
		IMDb	0.24	4.2	/	/
		IMDb$_{pbmt}$	/	/	2.39	27.4
		IMDb$_{nmt}$	/	/	2.76	32.3

Table 3.15: Corpus statistics: voc (en) and voc (sr) refer to the vocabulary size of English and Serbian, respectively.

Table 3.15 shows the statistics of the three training corpora (*SEtimes*, IMDb$_{pbmt}$ and IMDb$_{nmt}$), the development and the test set. Although the *SEtimes* corpus is not very large, it contains a decent amount of parallel segments ($224K$) which is not too small for training an NMT model capable of reasonable performance. For this reason, we consider the NMT model built from this data set as the baseline model as the NMT approach has recently become the state of the art in many cases. It can be seen in the table that the IMDb training data contains more than twice as many segments and running words than the English part of the *SEtimes* corpus, and it has a much larger vocabulary. Due to its rich morphology, the Serbian *SEtimes* vocabulary is almost twice as large as the English data. However, the Serbian vocabulary for the synthetic IMDb data is only barely larger or even comparable to the English one.

Machine-translated data generally exhibits less lexical and syntactic variety than natural data (Burlot and Yvon, 2018; Vanmassenhove et al., 2019), and here we are additionally dealing with a low-resource out-of-domain MT system translating into a more complex language. As expected for the development set, OOV rates are smaller for the in-domain *SEtimes* corpus and for the less morphologically complex English language. The English part of the test set behaves in the same way, i.e. the OOV rates are smaller when compared to the in-domain IMDb training data. However, the OOV rates for the synthetic Serbian data are comparable with those of the out-of-domain development data, and much higher than for the development data when compared to its in-domain *SEtimes* data.

3.4.3.3 Results

• **MT evaluation:** The results for both the development and the test sets are shown in Table 3.16. The results for the development set are as expected, i.e. the best option is to use an NMT system trained on the in-domain data (*SEtimes*), and using any kind of additional out-of-domain data causes all scores to deteriorate. In contrast, for the test set, it is expected that the scores can be worse than for the development set. However, the following interesting results can be observed:

(i) The baseline NMT system (NMT trained on *SEtimes* data) outperforms the baseline PBMT system (PBMT trained on *SEtimes* data) despite the scarcity of training data and domain mismatch.

(a) Overall automatic evaluation scores for the development set (SEtimes)

development set (SEtimes)						
system	**training data**	**BLEU↑**	**METEOR↑**	**TER↓**	**chrF↑**	**chrTER↓**
PBMT	SEtimes	33.1	29.4	48.9	61.2	41.5
NMT	SEtimes	**39.2**	**32.2**	**42.6**	**62.7**	**39.1**
	SEtimes+IMDb$_{pbmt}$	36.2	30.8	44.7	61.1	41.0
	SEtimes+IMDb$_{nmt}$	38.1	31.7	43.0	61.6	40.1
	SEtimes+IMDb$_{joint}$	35.1	30.2	45.5	59.8	41.9

(b) Overall automatic evaluation scores for the test set (IMDb)

test set (IMDb)						
system	**training data**	**BLEU↑**	**METEOR↑**	**TER↓**	**chrF↑**	**chrTER↓**
PBMT	SEtimes	10.8	18.6	69.1	40.5	56.3
NMT	SEtimes	13.7	19.2	65.8	37.4	61.4
	SEtimes+IMDb$_{pbmt}$	11.6	19.0	66.9	**40.7**	**55.3**
	SEtimes+IMDb$_{nmt}$	**14.7**	**20.4**	**63.2**	38.8	60.2
	SEtimes+IMDb$_{joint}$	13.3	19.7	64.8	40.6	55.5

Table 3.16: Overall word-level and character-level automatic evaluation scores for the development (*SEtimes*) and the test (IMDb) data

However, this happens only in terms of word-level scores whereas the observations for character-level scores are different.

The chrF score for the baseline NMT system decreases by 3.1 points compared to the baseline PBMT system, and the chrTER increases by 5.1 points, which shows that the PBMT system performs better at character level.

(ii) Adding the IMDb$_{pbmt}$ data causes the word-level scores to deteriorate but improves both character-level scores.

(iii) Adding the IMDb$_{nmt}$ data improves all the baseline scores, but the improvements in terms of the character-based scores are smaller than those produced by adding the IMDb$_{pbmt}$ data.

(iv) Using all of the synthetic data (IMDb$_{joint}$) improves all the scores (except BLEU) over the baseline. However, these improvements are smaller than the improvements of each individual synthetic data set (IMDb$_{nmt}$ for word-level scores and IMDb$_{pbmt}$ for character-level scores).

• **Automatic error analysis:** In order to better understand the character-metrics preference for the PBMT-based systems, we perform a more detailed evaluation in the form of automatic error classification of all translation outputs using the open source tool 'Hjerson' (Popović, 2011) which is based on the combination of edit distance, precision and recall, and distinguishes five error categories as follows: (i) inflectional error, (ii) word order, (iii) omission, (iv) addition and (v) mistranslation. Following the set-up used for a large evaluation involving many language pairs and translation outputs to compare the performance between the PBMT and NMT systems in Toral and Sánchez-Cartagena (2017), we group omissions, additions and mistranslations into a unique category called *lexical errors*. The results for both the development and the test sets are shown in Table 3.17 in the form of error rates (raw error count normalised over the total number of words in the translation output). The development set is used for minimum error rate training (Och, 2003) to tune the translation systems.

(a) Error rates (%) for the development set (SEtimes)

development set (SEtimes)				
system	**training corpus**	**inflection**	**word order**	**lexical**
PBMT	SEtimes	15.4	5.3	**36.1**
NMT	SEtimes	**11.8**	**4.0**	**36.1**
	SEtimes+IMDb$_{pbmt}$	12.5	4.4	37.2
	SEtimes+IMDb$_{nmt}$	11.8	4.1	36.6
	SEtimes+IMDb$_{joint}$	12.6	4.4	38.0

(b) Error rates (%) for the test set (IMDb)

test set (IMDb)				
system	**training corpus**	**inflection**	**word order**	**lexical**
PBMT	SEtimes	14.2	5.1	54.1
NMT	SEtimes	**10.0**	4.9	60.1
	SEtimes+IMDb$_{pbmt}$	14.4	**4.6**	**53.7**
	SEtimes+IMDb$_{nmt}$	10.4	5.0	57.3
	SEtimes+IMDb$_{joint}$	13.4	4.7	53.8

Table 3.17: Results of automatic error analysis including three error categories for the development (*SEtimes*) and test (IMDb) corpus

The findings for the in-domain development set are as expected and in line with the findings of Toral and Sánchez-Cartagena (2017).

The NMT system handles grammatical features (morphology and word order) better than the PBMT system (Bentivogli et al., 2016) whereas no difference is noticed regarding the lexical aspect.

We observe similar inflectional errors for the test set. The lowest inflectional error rate is seen for the baseline NMT system, which slightly increases when the IMDb$_{nmt}$ corpus is added. The other three systems involving the PBMT approach produce many more inflectional errors. In contrast, the situation is slightly different for the other two error categories. Word order also becomes better for the baseline NMT system than for the PBMT system, but adding the IMDb$_{nmt}$ corpus does not improve it, whereas the IMDb$_{pbmt}$ corpus does. One possible reason for this is the free word order in Serbian, so that the system trained on the IMDb$_{pbmt}$ data simply generates the word order closest to the one in the reference translation. From the lexical error perspective, it is seen that the lexical error rate is much higher for the baseline NMT system than for the baseline PBMT system, which corresponds to the domain-mismatch for NMT (Koehn and Knowles, 2017). In addition, the highest reduction of this error type is noticed when the IMDb$_{pbmt}$ corpus is added.

- **Lexical error analysis:** We perform a qualitative manual inspection of three translation outputs: (i) from the baseline NMT system, (ii) from the NMT system with additional IMDb$_{pbmt}$ corpus, and (iii) from the NMT system with additional IMDb$_{nmt}$ corpus. We observe the presence of many person names (actors, directors, etc., as well as characters) in the IMDb corpus. As mentioned earlier in Section 3.4.3.1, Serbian (Latin) allows both transcription as well as leaving the original names, but it should be consistent in a text. In contrast, the names in the test reference translation are left in the original form, and we see that neither of the MT systems handles the names in a consistent manner. Both the PBMT and NMT systems generates the original forms, transcriptions and sometimes unnecessary translations of the names in a random way. In addition, the NMT systems often omit or repeat (the parts of) the names.

	IMDb$_{pbmt}$ is better	**IMDb$_{nmt}$ is better**
source	best Clark Kent	to watch Patrick Duffy
reference	najbolji **Clark Kent**	gledati **Patricka Duffyja**
SEtimes	best Kent	pratiti **Patrick Duffy**
SEtimes+IMDb$_{pbmt}$	najbolji **Klark Kentu**	da gledaju Patrik Dafi
SEtimes+IMDb$_{nmt}$	best Kent Kent	pratiti **Patrick Duffy**
source	the Richard Donner Cut	Kate Winslet (as Rose)
reference	verziju **Richarda Donnera**	Kate Winslet (kao **Rose**)
SEtimes	odlaska Richard Cut	Winslet (kao Jack)
SEtimes+IMDb$_{pbmt}$	**Ričard Donner** smanji	Kate Winslet (kao ruža)
SEtimes+IMDb$_{nmt}$	Richard Cut Cut	Kate Winslet (kao **Rose**)

Table 3.18: Examples of different name entities (person names)

This finding explains both the increase in the lexical error rates as well as the decrease in the character-level overall scores for the NMT-based systems. Table 3.18 shows several examples where for each example, the best version of the given name is shown in bold. The names on the left are problematic for the baseline NMT system, which is then improved (albeit not always in the perfect way) by adding the IMDb$_{pbmt}$ corpus, but not improved (or even worsened) by adding the IMDb$_{nmt}$ corpus.

The names on the right are treated properly by both the baseline NMT system and the IMDb$_{nmt}$ system, but the IMDb$_{pbmt}$ system transcribes the first name thus making it more distant from the reference, and unnecessarily translates the second name as though it were a common noun.

It is worth noting that the MT-evaluation scores typically underreport the actual quality, as presumably many 'good' translations are produced which differ from the reference, and are thus penalised.

The 'forward-translation' technique we used in this experiment improved the baseline results, although 'backtranslation' (translating natural Serbian texts into English) would probably be more helpful. Further analysis showed that morphology and syntax are better handled by the NMT system compared to the PBMT system, whereas the situation is different for the lexical aspect, especially for person names.

In general, named entities in other Slavic languages have gender and case and it might be problematic. In addition, the described issues with named entities can appear in any other language which allows both a transcribed as well as the original form.

The findings in this experiment, together with the facts described in Section 3.4.3.1, indicate that Serbian, as well as other Slavic person names and other named entities should be further investigated in the context of MT, not only for movie reviews or other types of UGC, but also in general.

3.4.4 Experiment-4: Translating hotel reviews

This is a small experiment to investigate how a baseline MT system performs while translating restaurant reviews. Our data set consists of the parallel sentences from the *FourSquare* corpus (discussed in Section 3.2.4). The translation model is built from the 14,864 parallel training sentences, tuned on 1,243 parallel development sentences and tested on 1,838 test sentences. We refer to this model as our Baseline as it is built using OpenNMT which is one of the most popular NMT toolkits available. In Chapter 6 we will discuss our approach of building extended translation models and compare their performance with this baseline model.

3.4.4.1 Results

Table 3.19 shows the BLEU score obtained in this experiment.

Translation Model	**BLEU**
Baseline model	22.1

Table 3.19: BLEU score on translating test sentences from restaurant reviews

Although there is nothing interesting to see in this result, we have a special purpose for running this experiment. We are interested in investigating how a baseline MT system performs while translating UGC and then plan to improve the performance with the help of parallel data extraction for UGC. We will see in Chapter

5 that the BLEU score can be further improved by accompanying the *FourSquare* data set with additional UGC training data extracted by our proposed approach of automatic parallel data extraction for UGC.

3.4.5 Experiment-5: Measuring sentiment preservation in translation of UGC

In this experiment, we investigate how the MT systems perform in terms of sentiment preservation in UGC translation. We use the Twitter, Flickr and the News data (shown in Tables 3.2 and 3.5) in this experiment.

3.4.5.1 Sentiment classification and translation models

For the Twitter dataset, the sentiment scores are manually annotated. As the annotation was done by only one annotator, it was not possible to calculate the inter-annotator agreement. As expected, the tweets are informal in nature and hence their translations are also informal. For example, the English tweet 'GOAAAAL ♡ ♥ ♡ ♥' is manually translated as 'TOOOOR ♡ ♥ ♡ ♥' in German (the actual translation of 'goal' is 'Tor' in German) so that the positive emotion is emphasised in the manual translation. We consider the tweets with manually annotated sentiment scores as our 'gold standard' data.

However, it was impractical to manually assign sentiment scores to the Flickr and the News data because they are much larger, so instead we used a lexicon-based sentiment analysis tool (Afli et al., 2017b) for these data sets. We also evaluate the performance of this system by comparing its outputs with the manual sentiment annotations and found that it achieved an accuracy of 74.7% with a Pearson correlation coefficient of 0.603. These results show that this tool has a good correlation with manual sentiment annotation and so has the capability of assigning correct sentiment levels to the Flickr and the News data in most of the times.

Once we obtain the sentiment scores of all the text pairs, they are divided into the following three classes: (i) negative texts with sentiment score < 0.4, (ii) neutral texts with sentiment score $>= 0.4$ and $<= 0.6$ and (iii) positive texts with sentiment score > 0.6.

Data	Training	Development	Test
Twitter	$3,700$	150	150
Flickr	$29,000$	/	/
News_comm	$238,843$	/	/

Table 3.20: Data distribution after sentiment classification

Once the sentiment classification is performed, we held out 150 tweet pairs for tuning and another 150 tweet pairs (50 per sentiment class) for testing purposes. Table 3.20 shows the distribution of training, development and test data for both the in-domain (Twitter) and out-of-domain (Flickr and News) data sets.

For Arabic social media posts, we consider only two sentiment classes: (i) negative ($score <= 0.3$) and (ii) positive ($score >= 0.7$) because TWB were interested only in considering the posts that show either the users are satisfied with the service or not.

For our experiment on Tweets, we build three translation models as follows: (i) Model-1: Baseline; using the Twitter data only, (ii) Model-2; using the concatenation of Twitter and Flicker data, and (iii) Model-3; using the concatenation of Twitter, Flicker and the News_comm data.

For the experiment on Arabic social media posts, we build a baseline PBMT system from the whole corpus using the data shown in Table 3.6. Note that, we consider the PBMT system as the baseline because the data size in this experiment is too small (only $3.9K$ pairs of social media posts) to train a good NMT system.

3.4.5.2 Results

We evaluate the outputs in terms of both translation quality and sentiment polarity preservation. Table 3.21 shows the results. As the Twitter data set is too small (only $3.7K$ parallel segments) for NMT training, we apply the standard PBMT approach to build the translation model from this data set and so we consider it as the baseline model.

Translation model	**BLEU ↑**	**METEOR ↑**	**TER ↓**	**Sent_Pres.**
Model-1	50.3	60.9	31.9	66.66%
Model-2	50.7	62.0	31.3	62.66%
Model-3	**52.0** ∗	**63.4** ∗	**30.1** ∗	73.33%

Table 3.21: Experimental evaluation: With data concatenation

Note that, BLEU and METEOR are precision based metrics, that is, the higher the score the better the system. In contrast, TER is an error-based metric, so the lower the score the better the system. The last column represents the sentiment preservation score which shows the percentage of times the original sentiment class of the source-language text is retained after translation.

In terms of MT evaluation, we notice that the Baseline is outperformed by Model-2, which is trained from the concatenation of Twitter and Flickr data. However, the sentiment preservation score dips to 62.66%. The best performance is obtained by Model-3 which is built from the concatenation of all the three data sets. Model-3 obtains up to 73.33% of sentiment preservation which is around 10% relative improvement over the Baseline. We use the '*' symbol to show that the results produced by Model-3 are statistically significant, as determined by using MultEval (Clark et al., 2011).

We summarise the results of the experiments with Arabic social media posts in Table 3.22. The outputs generated by our translation system (Baseline) are compared with the outputs produced by Google Translate[14] in terms of both translation quality and sentiment preservation.

MT system	BLEU	METEOR	TER	Sent_Pres.
Google Translator	**18.1**	**22.1**	**71.7**	59.33%
Baseline	11.6	18.2	84.0	67.33%

Table 3.22: Results on Arabic UGC translation

Google Translate produces higher BLEU scores than the Baseline, which is expected because Google's MT system clearly has been built on much larger datasets than ours. However, it is interesting to notice that our translation system surprisingly obtains higher sentiment preservation than Google Translate (13% relative improvement).

In the above experiments with Twitter data and Arabic social media posts, our main concern is to investigate the system performance from sentiment preservation perspective. Here, we are particularly interested in finding techniques to improve this sentiment preservation score because maintaining sentiment polarity is more important than translation quality for MT-based sentiment analysis. With this research goal in mind, we implement a sentiment translation system with the aim of improving sentiment preservation which is discussed in detail in Chapter 6.

3.5 Conclusions

In this chapter, we investigated general issues in MT of UGC. We conducted several experiments translating different types of UGC for different languages and domains.

In the first experiment, we proposed a framework for Arabic UGC translation by integrating a error correction system prior to the translation phase. We conducted a set of experiments to analyse the impact of our proposed framework on the final translation. The experimental evaluation revealed that the integration of an

[14] `https://translate.google.com/`

error correction module as a pre-processing step is very helpful in improving the translation of Arabic UGC. We observed that all the systems built with or without tokenisation can decrease the word errors of the Arabic UGC. Our best model outperformed the Baseline by up to 4 BLEU points and 1.72 TER points. The morphological complexity of languages like Arabic, which contains billions of surface forms (e.g. 60 billion for Arabic), complicates correction methods such as dictionary-based methods because listing all the possible words is an extremely difficult task. We believe that our proposed method can be a good way to resolve such a problem.

In our second experiment, we presented a comparative study of the PBMT and NMT systems for translating a specific type of UGC, in this case, the *FIFA-2014* Twitter data. We summarised our findings in this work as follows: (i) when trained on about $270K$ segments, both the NMT and PBMT performance were on the same level in terms of automatic MT evaluation metrics, (ii) using smaller amounts of training data significantly deteriorated the performance of the NMT system, whereas the performance of the PBMT system was only moderately deteriorated. The NMT system trained on the tiny Twitter corpus exhibited useless performance even though all the data were in-domain compared to the test set. However, the back-translated data improved the system performance significantly, and adding more out-of-domain data resulted in even further improvements. These results revealed that the NMT systems are very data-hungry and can perform much better when significantly larger data is supplied for training, even though these data came from different domain, or the source part was synthetic. Considering the fact that parallel resources for Twitter are scarcely available, our experiments showed the potential for creating additional parallel Twitter data by incorporating back-translation and including out-of-domain parallel resources. Our best performing NMT system was built on the combination of only $3K$ in-domain tweets, $50K$ back-translated Harvard tweets and $200K$ short text segments from the 'News-Commentary' corpus, which is still considered to be a small amount of data for training NMT models.

In our third experiment, we focused on building English-to-Serbian MT systems for IMDb movie reviews. Firstly, we trained a PBMT and an NMT system on an out-of-domain clean parallel corpus and used them as the baselines. We then created additional synthetic in-domain parallel data by translating the English IMDb reviews into Serbian using the two baseline MT systems. The concatenation of this synthetic data as an additional resource then improved the baseline results. Further analysis showed that the morphology and syntax are better handled by the NMT system than by the PBMT system, whereas the situation is different from the lexical point of view, especially for person names. Our findings also revealed that in general, the translation of person names into Slavic languages (especially those which require/allow transcription) should be investigated more systematically.

In our fourth experiment, we performed an investigation on translating restaurant reviews. Although the experimental result did not seem to be interesting, we ran the experiment to analyse how a baseline MT system performs while translating such kind of UGC texts. We plan to improve the system performance by automatically extracting additional parallel data for UGC.

In our fifth and final experiment, we measured the sentiment preservation score in UGC translation. The data sets include tweets and social media posts. Although our baseline system is outperformed by the Google Translate in terms of translation quality, our system obtains higher sentiment preservation score. This is very interesting result as preserving the sentiment class is more important than the translation quality in MT-based sentiment analysis. It is crucial to further improve the sentiment preservation score so that the sentiment analysis in multilingual platform can be benefited from it.

In some of our experiments we used either out-of-domain or synthetic data to build enhanced MT models in order to improve the translation quality. However, there are other alternatives to deal with this issue. For example, implementing an efficient and good quality automatic parallel data extraction system to extract parallel segments/sentences from a comparable corpus. An appropriate way to do this is to implement a bilingual document alignment system that aligns bilingual similar document pairs so that the aligned documents can then be used for parallel sentence extraction as they contain similar information.

Furthermore, our findings in experiment-5 suggest that it is important to explore a new research area of 'sentiment preservation' in UGC translation. To this aim, we perform an in-depth analysis on the new concept of 'sentiment translation' models which will be discussed in Chapter 6.

In the next chapter, we will discuss our first research question that deals with 'bilingual document alignment' using comparable corpora.

Chapter 4

Bilingual Document Alignment

In the previous chapter we discussed general issues relating to MT of UGC. Our experimental evaluation revealed that developing an MT system for UGC is quite challenging due to the lack of parallel resources. In this chapter, we discuss our first research question that deals with 'bilingual document alignment' and describe our proposed approach to address this research question.

A parallel corpus is the main ingredient for building an MT system. A huge amount of parallel corpora has been developed so far, such as OPUS,[1] Europarl,[2] News Commentary[3] etc. However, parallel resources are not sufficiently available for many domains and for many language pairs. For example, medical science or space technology still lacks sufficient parallel resources for many language pairs. Furthermore, a parallel resource for UGC is rarely available on the Internet. Therefore, building robust MT systems in such scenarios is a challenging task.

In these circumstances, exploitation of comparable corpora becomes a useful alternative. A comparable corpus is a resource consisting of texts from the same domain in more than one language. Comparable corpora are useful in several multilingual

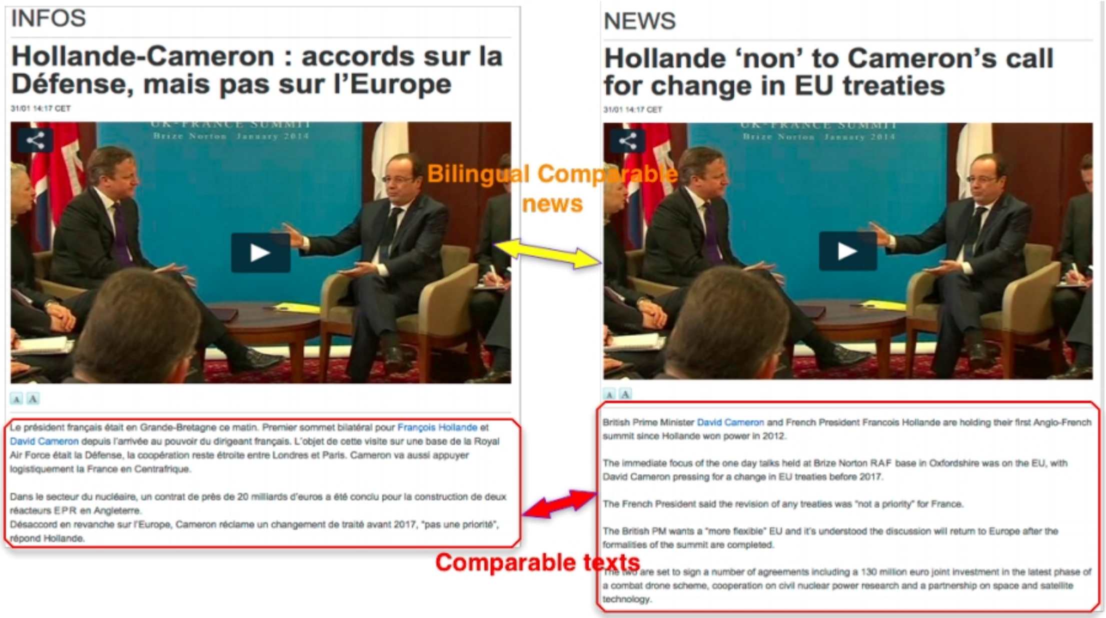
[1] http://opus.nlpl.eu/

[2] https://www.statmt.org/europarl/

[3] http://opus.nlpl.eu/News-Commentary-v11.php

NLP tasks. As they belong to the same domain, they may contain semantically similar information in different languages. For example, on 11^{th} September in 2001, it is safe to assume that all newspapers led with the attacks on the World Trade Centre in New York city. While their contents were not direct translations, the reports were comparable, and could be put to good use as possible training data in a range of NLP applications for which large amounts of relevant, truly parallel data are unavailable.

A bilingual comparable corpus is a collection of texts in the same domain in two different languages.



INFOS

Hollande-Cameron : accords sur la Défense, mais pas sur l'Europe

31/01 14:17 CET

Le président français était en Grande-Bretagne ce matin. Premier sommet bilatéral pour François Hollande et David Cameron depuis l'arrivée au pouvoir du dirigeant français. L'objet de cette visite sur une base de la Royal Air Force était la Défense, la coopération reste étroite entre Londres et Paris. Cameron va aussi appuyer logistiquement la France en Centrafrique.

Dans le secteur du nucléaire, un contrat de près de 20 milliards d'euros a été conclu pour la construction de deux réacteurs EPR en Angleterre.
Désaccord en revanche sur l'Europe, Cameron réclame un changement de traité avant 2017, "pas une priorité", répond Hollande.

NEWS

Hollande 'non' to Cameron's call for change in EU treaties

31/01 14:17 CET

British Prime Minister David Cameron and French President Francois Hollande are holding their first Anglo-French summit since Hollande won power in 2012.

The immediate focus of the one day talks held at Brize Norton RAF base in Oxfordshire was on the EU, with David Cameron pressing for a change in EU treaties before 2017.

The French President said the revision of any treaties was "not a priority" for France.

The British PM wants a "more flexible" EU and it's understood the discussion will return to Europe after the formalities of the summit are completed.

The two are set to sign a number of agreements including a 130 million euro joint investment in the latest phase of a combat drone scheme, cooperation on civil nuclear power research and a partnership on space and satellite technology.

Figure 4.1: Example of comparable documents from the Euronews Web site (Afli et al., 2017a)

Figure 4.1 shows an example of the same news published in French and English. The texts are not exact translations of each other because they are likely to be reported by different reporters. However, these news are comparable as they contain similar information and thus can be exploited to extract similar bilingual sentence/phrase pairs. It is, therefore, very useful to extract such semantically similar bilingual document pairs, which can be then treated as a useful bilingual resource. This task can be accomplished by the application of 'bilingual document alignment' which is a method to identify semantically similar documents in two different languages.

One of the easiest ways of developing a bilingual document alignment system is to employ an MT system that translates the source-language documents into the target language and then measures text similarity. Such an approach is best suited for a small bilingual comparable corpus because the translation of the source-language documents can be done in a reasonable amount of time. However, for a comparatively much larger corpus, it is impractical to translate all the source-language documents into the target language as this would require a huge amount of time. Moreover, building an MT system itself for translating the documents consumes a significant amount of time. Furthermore, a translation model for a specific domain may not be available to help with alignment task. Considering this scenario, we propose to design a bilingual document alignment system without using any MT system. We conduct two different experiments as follows:

- **Experiment 1**: measuring the (a) sentence-based, (b) word-based and (c) named entity (NE)-based scores to extract the similar document pairs.
- **Experiment 2**: combining (a) CLIR, (b) word embedding-based similarity and (c) text similarity to perform the alignment task.

We discuss these methods in detail in Section 4.3.

4.1 Related Work

Most work on bilingual document alignment use the Web as a comparable corpus (Zhao and Vogel (2002);Resnik and Smith (2003)). The work in Dara and Lin (2016a) combine (i) URL-matching, (ii) n-gram-matching and (iii) IR-based methods for document alignment. Le et al. (2016) use the following methods: (i) measuring the term position similarity between the candidate document pairs, (ii) matching the automatically translated versions of the target-language text with the candidates, and (iii) considering the string similarity of URLs of the document pairs. Medved et al. (2016) align English–French web pages based on statistical extraction of keywords and compare them using a translation dictionary. Yang and Li (2003) identify one-to-one title pairs in an English–Chinese corpus collected from the Web. Their approach is based on applying the longest common sub-sequence to find the most reliable Chinese translation of an English word. Utiyama and Isahara (2003) extract similar article pairs, and then align sentences using a sentence-pair similarity score and use a dynamic programming method to find the least-cost alignment over the document pair.

Munteanu and Marcu (2005) use a bilingual lexicon to translate some of the words of the source-language sentence and then use the translations as a query to find matching translations using IR. Li and Gaussier (2010) develop a comparability measure based on the expectation of finding translation word pairs in the corpus. Bitextor[4] (Esplà-Gomis, 2009) and ILSPFC[5] (Papavassiliou et al., 2016) employ web-based methods to extract monolingual/multilingual comparable documents from multilingual websites. Afli et al. (2016b) show that it is possible to extract only 20% of the true parallel data from a collection of sentences with $1.9M$ tokens by applying an automated approach. However, Kúdela et al. (2017) efficiently identify parallel segments at paragraph level from the pages of a web domain regardless of their structure with much higher accuracy. Klempová et al. (2009) create 'document descriptor vectors', binary vectors of varying length reflecting the linear structure of the HTML source, in addition to pre-filtering by sentence or word lengths for document alignment. Balikas et al. (2018) propose Wasserstein distance and its regularised version in the task of crosslingual document retrieval. El-Kishky et al. (2019) employ URL-matching rules to curate crosslingual documents from the com-

[4] `http://bitextor.sourceforge.net/`

[5] `http://nlp.ilsp.gr/redmine/projects/ilsp-fc`

moncrawl corpus.[6] They also focus on aligning documents based on content rather than meta-information. El-Kishky and Guzmán (2020) present a crosslingual sentence mover's distance metric to measure the semantic similarity of two documents in different languages.

Although most works employ MT to ease the task of bilingual document alignment, it is in fact not the best solution as the translation process itself requires a huge amount of time. We propose a different approach that removes the translation process instead, we apply word-embeddings in combination with text similarity accompanied by a bilingual dictionary. We will see in Section 4.3.4 that better results can be obtained using our proposed method compared to when we use an MT system for the document alignment task.

4.2 Architecture of the Document Alignment System

The architecture of a generic bilingual document alignment system can be depicted in Figure 4.2. Note that, this is the simplified version without showing the low-level details. The whole system is usually based on the combination of a series of measurement methods. In our case, we use different similarity measurement methods in experiment-1 (Section 4.3.3) and experiment-2 (Section 4.3.4). Firstly, we measure the individual similarity scores between the source- and the target-language documents using each of these methods. Secondly, we combine all the scores to obtain the total similarity score. Finally, we select the candidate target-language document with the highest score as the target alignment. The final result is a one-to-one mapping of the source- and target-language documents.

[6] https://commoncrawl.org/

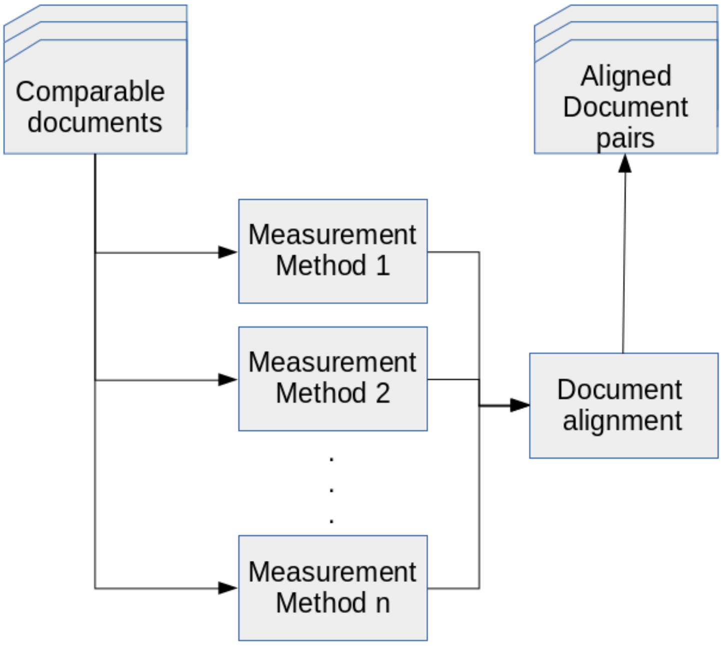


Figure 4.2: Bilingual document alignment system

4.3 Experiments

4.3.1 Dataset

We use two different datasets for the experiments; (i) Euronews data and (ii) WMT-2016 test data.

- **Euronews data:** The Euronews data (Afli et al., 2017a) consists of a multimodal corpus of comparable documents and their images in 9 different languages containing news articles from the Euronews website.[7] The 9 languages are English, Arabic, German, Spanish, French, Italian, Portuguese, Turkish and Ukranian.

  We consider only the textual part and their multilingual alignments. In our experiments, we use English, French, German and Arabic documents. Table 4.1 shows the data statistics when English is considered as the source language.

[7] https://www.euronews.com/

Language Pair	# Source	# Target	# Aligned
En-Ar	$40,421$	$36,836$	$35,761$
En-De	$40,421$	$37,293$	$36,114$
En-Fr	$40,421$	$37,293$	$36,762$

Table 4.1: Statistics of the Euronews data set

Language	# Documents
English	$681,611$
French	$522,631$

Table 4.2: Statistics of the WMT-2016 test data set

- **WMT-2016 test data:** This data set is provided by the organisers of the shared task on 'bilingual document alignment' at the 'First Conference on Machine Translation (WMT16)'.[8] It consists of texts from 203 web domains with more than 1 million documents in total, with over $600K$ English and over $500K$ French documents as shown in Table 4.2. Each document contains at least one line of text.

4.3.2 MT System Architecture

To build the translation models, we use the Moses phrase-based MT system (Koehn et al., 2007). The language models are trained using SRILM (Stolcke, 2002) and the word and phrase alignments are obtained using the GIZA++ alignment tool (Och and Ney, 2003). We set the maximum phrase length for training to 7. Finally, the models are tuned using MERT (Och, 2003).

4.3.3 Experiment 1: Using the length and NE-based similarities

In this section, we discuss our initial experiments on 'bilingual document alignment' using mainly the text and NE-based similarity measurements. The following scoring methods are used to obtain the similarity score in this experiment: (i) sentence-based scoring, (ii) word-based scoring, and (iii) NE-based scoring.

[8] `http://www.statmt.org/wmt16/bilingual-task.html`

The aim of the WMT-2016 'bilingual document alignment' task was to identify English–French document pairs from a given collection of comparable documents such that one document is the translation of the other. However, the alignment was done at URL level; that is, the task was to align the URL of a source-language document with the URL of the target-language document whose texts are comparable. We align the URLs solely based on their contents without performing any URL matching because our main focus was on measuring the content similarity rather than finding URL similarity. We discuss each of the text similarity measurement methods later in Section 4.3.3.2.

4.3.3.1 System Architecture

The architecture of the bilingual document alignment system is depicted in Figure 4.3.

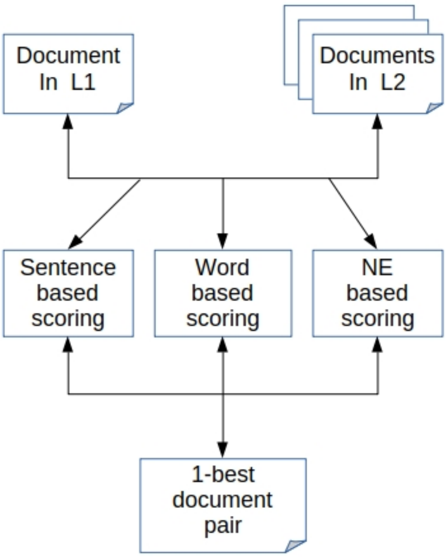


Figure 4.3: Architecture of our document alignment system

The whole system works in the following simple steps:

(i) each source-language (L1) document is compared with all the target-language (L2) documents one by one,

(ii) each comparison is done by using the sentence-based, word-based and NE-based scoring,

(iii) a specific weight is assigned to each of the above components and then they are simply combined to obtain the final score, and

(iv) the document in L2 that has the highest similarity score is aligned with the document in L1.

4.3.3.2 System Description

Now we discuss in detail all the similarity measurements we use in this experiment.

• **Sentence-based scoring:** A further constraint in the bilingual document alignment task of WMT-2016 was that it was required to align the documents from the same web-domain. Comparing each source-language document with all the target-language documents can take a huge amount of time even for a single web domain that contains thousands of documents. To avoid this situation, we restrict the comparisons only to those document pairs that have a close sentence-length ratio. Let S_s and S_t be the total number of sentences in the source- and the target-language documents, respectively. Assuming that, the sentence-length ratio (R_{SL}) is calculated using the formula shown in Equation (4.1)

$$R_{SL} = \frac{min(S_s, S_t)}{max(S_s, S_t)} \tag{4.1}$$

This equation confines the ratio between 0 and 1. If either the source- or the target-language document contains no sentences, then $min(S_s, S_t) = 0$, hence $R_{SL} = 0$, and $R_{SL} = 1$ if they contain the equal number of sentences. Therefore, a value of 1 or close to it gives a slightly positive indication of being comparable. However, this is not the only requirement, as there are many documents that contain an equal or very similar sentences. We, therefore, also consider the word-based and NE-based scores.

- **Word-based scoring:** Word-based scores are calculated using the word-length comparisons. The word-length ratio (R_{WL}) is calculated using the Equation (4.2)

$$R_{WL} = \frac{min(W_s, W_t)}{max(W_s, W_t)} \tag{4.2}$$

In the above equation, W_s and W_t are the number of words in the source-language and the target-language documents, respectively.

- **NE-based scoring:** We extract NEs from all the documents using the Stanford Named Entity Recognizer[9] (Finkel et al., 2005) that can extract NEs in multiple languages. Let us assume that NE_S and NE_T are the total number of NEs present in the source-language and the target-language document, respectively. We then calculate the NE-length ratio (R_{NL}) using Equation (4.3)

$$R_{NL} = \frac{min(NE_s, NE_t)}{max(NE_s, NE_t)} \tag{4.3}$$

Now, let us assume that the total number of matched NEs is M_{NE}. Assuming that, we now calculate the ratio between M_{NE} and NE_s. We refer to this ratio as R_{SNM} which is calculated as shown in Equation (4.4)

$$R_{SNM} = \frac{M_{NE}}{NE_s} \tag{4.4}$$

However, in many cases, the source-language and the target-language document can differ hugely in the total number of NEs they contain, and it may happen that most or all the source-language NEs may match with those in the target-language document. For example, if NE_s and NE_t are 2 and 10, respectively, $R_{SNM} = 1$. It is not practical to consider R_{SNM} as the NE-based score because the document pair need not necessarily be a good alignment in this case.

[9] `http://nlp.stanford.edu/software/CRF-NER.shtml`

Feature	**Weight combinations**				
	C_1	C_2	C_3	C_4	C_5
R_{SL}	0.33	0.25	0.15	0.1	0
R_{WL}	0.33	0.25	0.15	0.1	0
SC_{NE}	0.33	0.5	0.7	0.8	1

Table 4.3: Different weight combinations for different features

We therefore calculate the final NE-based score(SC_{NE}) by multiplying R_{SNM} by R_{NL} as shown in Equation (4.5)

$$SC_{NE} = R_{SNM} * R_{NL} \tag{4.5}$$

- **Combining the Scores:** We calculate the final similarity score by summing up the sentence-, word- and NE-based scores. However, it is required to find proper weights for each of these similarity measurement components. In order to do this, we perform a random selection of documents from 10 web-domains from the WMT-2016 training data and use them to tune the weights. We assign weights in 5 different combinations (C_n, where $n = 1, 2...5$) so that their sum is always 1. Note that more combinations can be applied to tune the system but our focus was on using equal weights for all the features and then increasing the weight of SC_{NE} while decreasing others, as NE-based score is the key component of our similarity measurement technique. Table 4.3 shows the different combinations of weights we explore in this experiment. It can be seen that C_1 represents the combination where all the features are assigned equal weights. Afterwards, the weights of R_{SL} and R_{WL} are decreased whereas that of SC_{NE} is increased gradually. In C_5, SC_{NE} is assigned full weight whereas R_{SL} and R_{WL} are not considered at all. Let us assume that the weights assigned to R_{SL}, R_{WL} and SC_{NE} are λ_1 , λ_2 and λ_3, respectively. Having said that, the final similarity score of a document-pair is calculated using Equation (4.6)

$$SC_A = \lambda_1 R_{SL} + \lambda_2 R_{WL} + \lambda_3 SC_{NE} \tag{4.6}$$

Note that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ in the above equation.

4.3.3.3 Results

We show our experimental results on the development data with different scoring combinations in Table 4.4.

Weight Combination	# References	# Correct Outputs	Recall
C_1	247	147	0.5951
C_2	247	152	0.6153
C_3	247	153	0.6194
C_4	247	153	0.6194
C_5	247	147	0.5951

Table 4.4: Result on Development data

Both C_3 and C_4 produce the highest recall value of 0.6194. We choose C_3 as the optimal combination and apply it to the test data set. Although the WMT-2016 test data set contains several hundred thousand documents for each language, the organisers provided only $2,402$ reference aligned document pairs after we submitted our system.

Therefore, we were unaware of the fact that it was not required to find alignments for all the source-language documents. The reason was that the organisers encouraged the participants to find alignments for all the documents and thus recall was the only evaluation metric. For this reason, we do not calculate the precision value. We calculate only the recall value which is shown in Table 4.5. It is important to observe that the results on the development data are much better than on the test data (compare Table 4.4 and Table 4.5). The reason is that we tuned our system on a very small development data, resulting in a lower number of comparisons which makes the alignment task easier. In contrast, the test data is much larger and results in a very large number of comparisons, so the error propagates and we obtain a lower recall value.

Weight combination	# Test references	# Correct outputs	Recall
C_3	$2,402$	699	0.291

Table 4.5: Result on test data

In addition, our manual analysis on the source of misalignment problems reveals that there exist many articles that deal with similar topics in different documents without any significant amount of text and NE matches. It is therefore not always helpful to rely mostly on length similarity and NE-count matching for obtaining good quality alignments.

4.3.4 Experiment 2: Combining CLIR and Word Embedding

In the previous experiment, we found that the naive approach based on only length and NE-based similarity is not sufficient enough to achieve a good quality document alignment system. The low recall value suggests that a more sophisticated method should be applied so that the system performance increases significantly. An ideal way to address this issue would be to apply an MT system where all the source-language documents are translated into the target language. This process eases the computation of text similarity because simply calculating word matches is then sufficient to retrieve the alignments. However, it is impractical to apply an MT system to a large comparable corpus because of the computational overhead of translating all the source-language documents into the target language. Moreover, it takes a significant amount of time to build the MT system itself. Considering this situation, we attempt to reduce such complexity and at the same time introduce an effective method of document alignment. To this aim, we propose to apply an inverted index-based CLIR method in combination with the word embedding-based method without using any MT system. For this reason, this method requires much less computation time compared to the MT-based method. We develop a tool using this strategy and due to its faster nature, we call it *Fast Document Aligner (FaDA)* (Lohar et al., 2016b).

The FaDA tool[10] measures the distances between the embedded word vectors in combination with the text similarity between the source-language and the target-

[10] Available at `https://github.com/loharp/FaDA`

language documents. We initially construct a pseudo-query[11] from a source-language document and then translate the query terms into the target language. We then represent the target-language documents and the translated pseudo-query as word embeddings to find the average similarity between them. Finally, the word embedding-based similarity is combined with the text similarity in order to obtain the total similarity score.

4.3.4.1 Architecture of FaDA

The architecture of FaDA is divided into two main components; (i) the CLIR-based system, and (ii) the word embedding-based system.

• **CLIR-based system:** We use the Lucene[12] framework to index the documents in our CLIR system. The architecture of our CLIR-based system is shown in Figure 4.4. It works in the following steps:

(i) the source-language and the target-language documents are indexed,

(ii) each source-language document is used to construct a pseudo-query from only a fraction of the constituent terms of that document. It is not practical to use all the terms of a document to construct the pseudo-query because using too long a query results in a very slow retrieval process. Furthermore, it is more likely that a long query will contain many outlier terms which are not related to the core topic of the document which will reduce retrieval effectiveness,

(iii) the pseudo-query is then translated into the target language by a bilingual dictionary which is built using the GIZA++ word alignment tool (Och and Ney, 2003) from the English–French parallel sentences from the 'Europarl' corpus,

[11] A *pseudo-query* is the modified form a user's original query in order to improve the ranking of retrieval results compared to the original.

[12] `https://lucene.apache.org/`

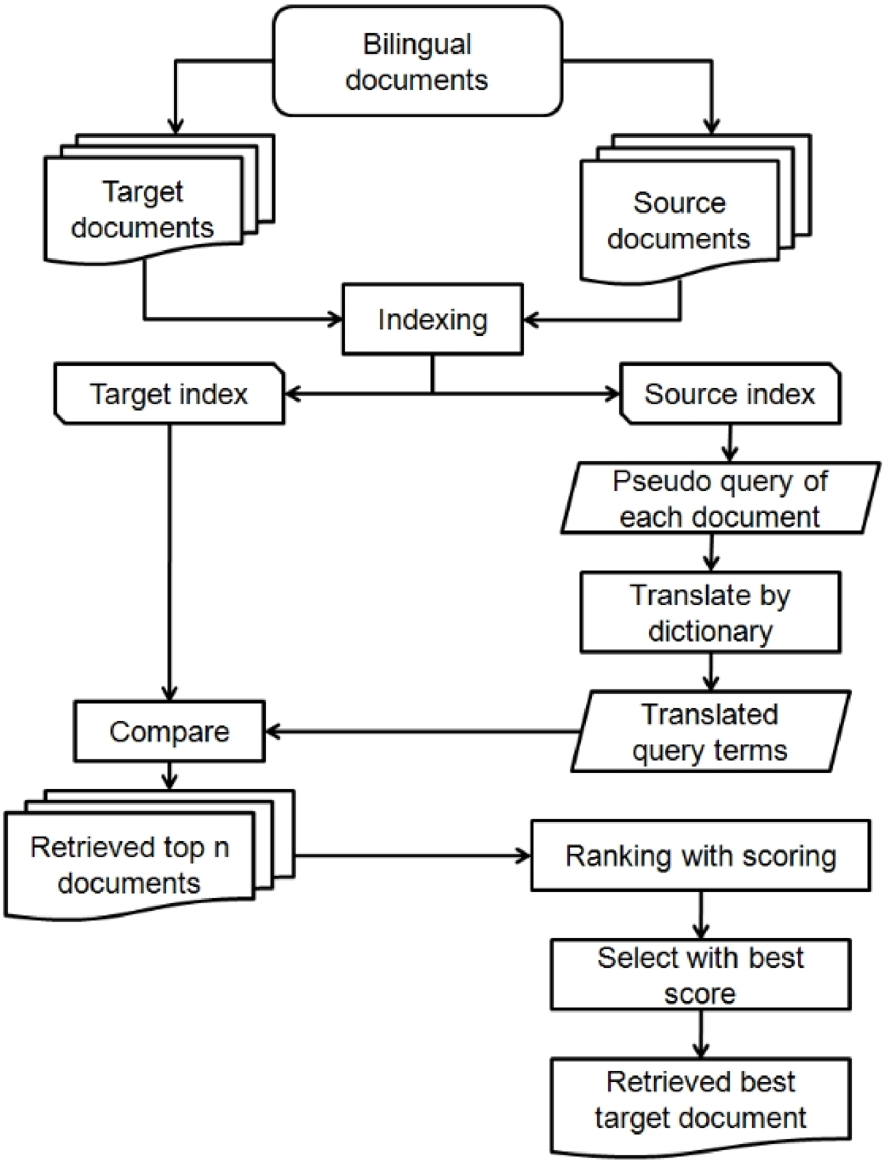


Figure 4.4: Architecture of CLIR-based system

(iv) the translated query terms are then searched in the target-language index to retrieve the top-n[13] candidates using the CLIR approach, and

(v) the retrieved top-n documents are ranked based on the descending order of scores.

[13] In this experiment we consider the top 10 candidates, that is $n = 10$.

- **Word embedding-based system:** The architecture of this system is shown in Figure 4.5.

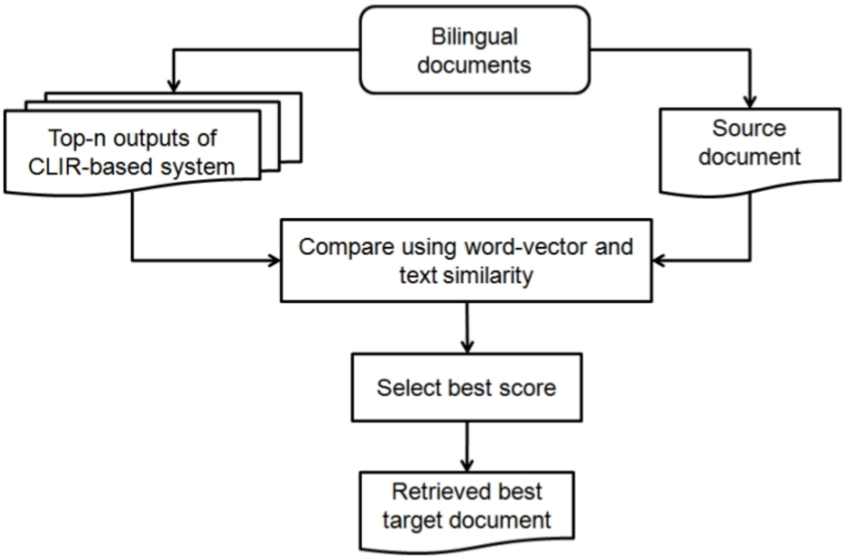

Bilingual documents
Top-n outputs of CLIR-based system
Source document
Compare using word-vector and text similarity
Select best score
Retrieved best target document

Figure 4.5: Architecture of word embedding-based system

It works in the following steps:

(i) the top-n outputs of the CLIR-based system are taken as the input to the word embedding-based system,

(ii) each source-language document is compared with each of the n candidate target-language documents using the word-embeddings in combination with the text similarity measurements,

(iii) finally, we select the candidate document with the highest similarity score and consider it as the target alignment.

4.3.4.2 System Description

The previous section outlined the architecture of *FaDA* at a high level. In this section we explain the finer working details of each component of *FaDA*.

- **CLIR-based system:** In this approach, firstly the source-language and the target-language documents are indexed and then each indexed source-language document is used to construct a pseudo-query which is considered to be the suitable representative of the document. We use Equation (4.7) in order to select the terms for pseudo-query formation.

$$\tau(t,d) = \lambda \frac{tf(t,d)}{len(d)} + (1-\lambda)\log(\frac{N}{df(t)}) \tag{4.7}$$

In the above equation, $tf(t,d)$ denotes the term frequency of a term t in a document d, $len(d)$ refers to the length of d, and N and $df(t)$ are the total number of documents and the number of documents in which t occurs, respectively. In addition, $\tau(t,d)$ represents the term-selection score and is a linear combination of the normalised term frequency of a term t in d, and the inverse document frequency (idf) of the term.

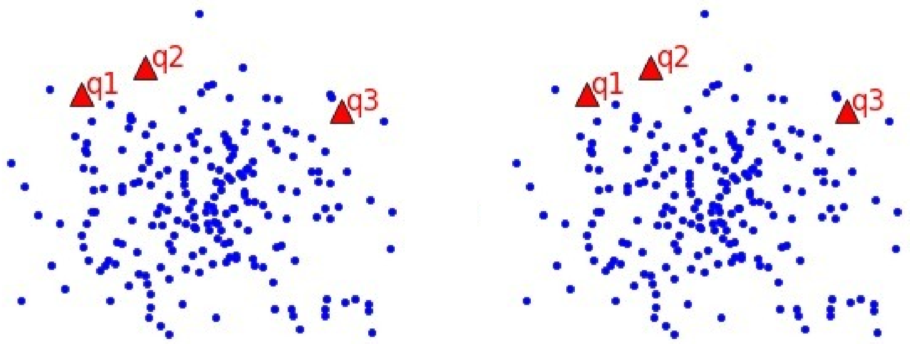
Equation (4.7) shows that the frequent terms of the document d and the relatively less frequent terms in the whole collection are given a higher priority due to the following reasons: (i) the frequent terms of d are the key contents and act as the important part of d, (ii) the less frequent terms (sometimes, rare terms) in the whole collection exist only in a few documents and so they are good indicators of finding those documents easily. The parameter λ controls the relative importance of tf and idf. Each term in d is associated with a score using this function and the terms are sorted in decreasing order of score. Afterwards, a fraction σ (between 0 and 1) is selected from this sorted list to construct the pseudo-query from d. The query terms are then translated by a bilingual dictionary. As a word can be translated in different ways depending upon the context, we use multiple translations of a query term (the term 'M' shown in Table 4.6 and Table 4.7 in Section 4.3.4.4). The translated query terms are then compared with the indexed target-language documents. Once the comparison is done, the top-n documents are extracted and ranked using the scoring method shown in Equation (4.7).

- **Word embedding-based system:** This approach considers the vector embedding of words and incorporates them with the CLIR-based approach in order to estimate the semantic similarity between the source-language and the target-language documents. It facilitates the formation of 'bag-of-vectors' (BoV) which expresses a document as a set of words with one or more clusters of words where each cluster represents a topic of the document.

We now demonstrate the key idea of the usefulness of the set-based similarity of the constituent word vectors of documents and queries with illustrative examples. Consider the documents of Figure 4.6, where for illustrative purposes, we assume that each word is embedded in a two-dimensional space. The individual word vectors of a document are shown with dots, whereas the translated query word vectors are shown with triangles. Note that the document in Figure 4.6a has one cluster and all three query points are relatively close to the centroid of this cluster. In contrast, for the document in Figure 4.6b, the query terms are relatively far away from the central theme of the document, i.e. the position of the centroid vector of the predominant topic of the document. This indicates that the posterior query likelihood for the document in Figure 4.6b is lower than that of Figure 4.6a, which means that the document of Figure 4.6a will be predicted as a more probable alignment with the query document than the document in Figure 4.6b.

Intuitively speaking, the closer the translated pseudo-query terms are to the clusters of the constituent word vectors of a target-language document, the higher the likelihood of the alignment.

It is observed that a standard CLIR-based system is only capable of using the query term matches, and cannot calculate the semantic distance between the terms. In contrast, the word embedding-based similarity shown is able to use the semantic distance and so can be considered as a more sophisticated approach for developing a document alignment system.



(a) (b)

Figure 4.6: Example cases of word vector based query likelihoods

- **Text similarity:** Our text similarity is based on term overlap between the source and the target language documents. The probability-based language modeling approach is applied to calculate this similarity.

- **Combining word-embedding with text similarity:** Following the calculation of word embedding and text similarity, we introduce an indicator binary random variable to combine the individual contributions of the text-based and word vector-based similarity. Let us denote this indicator by α. We now construct a mixture model of the two similarity measurement methods to calculate the final similarity score as shown in Equation (4.8)

$$Sim_{FINAL} = \alpha Sim_{TEXT} + (1 - \alpha) Sim_{WVEC} \tag{4.8}$$

- **Index Construction:** We run the K-means clustering algorithm for the whole vocabulary of the words in the English documents of the Euronews corpus. This process clusters the words into distinct semantic classes. Each of these semantic classes is different and discusses a global topic (i.e. the cluster id of a term) of the whole collection. This results in embedding of the semantically related words in close proximity. The cluster-id of each constituent term is retrieved using a table look-up while indexing each document in order to obtain the per-document topics from the global topic classes. The words of a document form different groups based on their cluster-id values. Subsequently, the cluster centroid of each cluster-id is computed by calculating the average of the word vectors in that group.

Finally, the information about the cluster centroids is stored in the index. This facilitates computing the average similarity between the query points and the centroid vectors during the retrieval process.

We refer the reader to Lohar et al. (2016b) for more technical details on our word embedding-based system, text similarity and index construction.

4.3.4.3 Experimental Setup

In this experiment, we consider French as the source-language and English as the target language. We use two different sets of data; namely (i) *Euronews* data extracted from the *Euronews* website[14] and (ii) the WMT-16 test dataset,[15] which is already shown in Table 4.1 and Table 4.2. We use the Jaccard similarity coefficient[16] (JSC) as the baseline system. This method is based on the measurement of the term overlap between the document pair. It solves two purposes as follows: (i) NE matches are considered, and (ii) the common words between English and French are also matched.

We build the translation model using the French–English parallel sentences from the 'Europarl' corpus (Koehn, 2005) in order to translate the French documents into English. The system is tuned on the Euronews data and the optimal parameters are applied on the WMT-2016 test data.

[14] http://www.euronews.com

[15] http://www.statmt.org/wmt16/

[16] https://en.wikipedia.org/wiki/Jaccard_index

Method	**Parameters**		**Evaluation Metrics**			**Run-time**
	τ	M	**Precision**	**Recall**	**F-score**	**(hh:mm)**
JaccardSim	N/A	N/A	0.0433	0.0466	0.0448	08:30
JaccardSim-MT	N/A	N/A	0.4677	0.5034	0.4848	36:20
CLIR ($\lambda = 0.9$)	0.6	7	0.5379	0.5789	0.5576	**00:05**
CLIR-WVEC ($\lambda = 0.9$)	0.6	7	**0.6252**	**0.6728**	**0.6481**	00:13

Table 4.6: Results on the development set (EuroNews dataset).

4.3.4.4 Results

We tune our document alignment system by exploring different parameter values and found that the optimal performance is obtained using the following parameter settings; (i) $\lambda = 0.9$, (ii) $M = 7$, that is, using 7 translation terms, and (iii) $\tau = 0.6$, which is the 60% of the terms from a document to construct the pseudo-query. The results on the Euronews data with the tuned parameters are shown in Table 4.6. We can see from the table that the baseline system (JaccardSim) takes more than 8 hours to complete the alignment task. This approach has a quadratic time complexity as all possible comparisons are taken into account. Moreover, the run time exceeds 36 hours when combined with the MT system. In contrast, the application of our CLIR-based approach reduces the runtime dramatically, taking only 5 minutes to produce the results. In addition, the 'JaccardSim' method has a very low effectiveness and can only lead to a considerable improvement when accompanied by the MT system.[17] The CLIR-based approach in combination with word embedding-based system (CLIR-WVEC) gives the highest scores both in terms of precision and recall.

The results on the WMT-2016 test dataset is shown in Table 4.7. The official evaluation metric in WMT-2016 was only the recall measure to estimate the effectiveness of the document-alignment system. We achieve a recall value up to 0.66 which is a massive improvement over the recall value of 0.29 (127% relative improvement) obtained by our length-based and NE match count-based methods discussed in Section 4.3.3.3.

[17] A French-to-English PBMT built from Europarl corpus is used to translate all the French documents into English and then Jaccard similarity is performed.

Method	**Parameters**				**Recall**	**Run-time**
	λ	τ	M	α		(hhh:mm)
JaccardSim	N/A	N/A	N/A	N/A	0.4950	130:00
CLIR	0.9	0.6	7	N/A	0.6586	007:35
CLIR-WVEC (FaDA)	0.9	0.6	7	0.9	**0.6619**	024:18

Table 4.7: Results on the WMT-2016 test dataset

The results produced by our system are not comparable with the results produced in the 'bilingual document alignment' task in WMT-2016 because the original task was to align the urls instead of the actual documents. In contrast, our system is solely based on content similarity and does not consider the urls at all. Although our system is not comparable with the submissions in WMT-2016, we attempt to perform the comparison in terms of runtime complexity. Let us consider the best submitted systems. Dara and Lin (2016b) did not report the runtime but it can be estimated. They use translation matches, url matches and IR approaches and can be expected to have similar runtime as ours. Gomes and Pereira Lopes (2016) use the knowledge encoded in PBMT phrase tables which means that they generated phrase tables by training the MT models and so it certainly took a longer time than our system. However, the alignment system of Buck and Koehn (2016) runs in less than 4 hours which is faster than our system.

The dataset is huge and contains a few hundred thousand French documents, each of which consists of at least one line of text. It is impractical to translate all of them as it takes an unrealistically large amount of time. Therefore, we do not use the 'JacardSim-MT' system for the WMT-2016 dataset in order to reduce the time complexity of the whole system. In this experiment, we consider the CLIR-based approach as our baseline system. IR-based methods are the standard approach in a bilingual document alignment system using the translated query terms (Dara and Lin, 2016a).

We make the following observations from Table 4.7:

(i) the JaccardSim method has a high runtime of 130 hours, showing a quadratic time complexity. In contrast, the CLIR-based system is much faster and completes the whole alignment process in only 7 hours. Moreover, it produces much higher recall scores than the JaccardSim method,

the FaDA system, which is the combination of CLIR and word embedding-based similarity measurements further increases the recall value.

It is expected that using a small number of clusters can have a negative impact on the alignment efficiency for sparse data. We explored different cluster values and found that the best result is obtained when the number of clusters is set to 50.

As the WMT-2016 data set contains only English and French documents, it was out of scope in Lohar et al. (2016b) to conduct similar experiments with any other language pair. However, it is useful to investigate the performance of our alignment system on other languages. To this aim, we run the FaDA tool on the 'Euronews' data because it contains comparable corpora for other languages apart from English and French. We select English–German and English–Arabic for our additional experiments. We apply the same optimal parameter settings and calculate the scores for these language pairs. The new results are shown in Table 4.8.

Language pair	**Precision**	**Recall**	**F-Score**
English–German	0.5241	0.5399	0.5319
English–Arabic	0.4921	0.5056	0.4987

Table 4.8: Results for English–German and English–Arabic

Let us compare these scores with those in Table 4.6. The scores for the English–German language pair are slightly lower than those for English–French. A probable reason is that German is more complicated than French in terms of morphology. Nonetheless, *FaDA* manages to achieve similar scores as in English–French. However, we can notice a further decrease in the scores for English–Arabic, which is possibly because Arabic comes from a completely different family of languages and its structure is completely different from English. Moreover, it has richer morphology than English. It is more difficult for our alignment system to perform as well for Arabic as it does for French and German.

4.4 Conclusions

Many research works have investigated the use of comparable corpora either to generate initial training data for MT engines, or to supplement the available data. In this chapter, we discussed the development of a bilingual document alignment system to exploit comparable corpora. We applied our system on a data set of bilingual comparable documents provided by the organisers of the shared task on 'bilingual document alignment' at WMT-2016.

In our first experiment, we proposed to combine the sentence-based, word-based and NE-based scores for the alignment task. We achieved a recall value of 0.291 for the WMT-2016 test data. Such a low recall value implied that a significant amount of possible valid alignments were discarded in our experiment. Therefore, the naive approach used in the first experiment was not sufficient to achieve good quality alignments. We needed to employ a more sophisticated method that is capable of aligning the documents much more accurately.

In our second experiment, we developed an open-source bilingual document alignment tool called *FaDA* based on a CLIR-based method in combination with word embedding-based similarity and the text similarity measurements. The CLIR approach uses an inverted index-based method and constructs a pseudo query from the source-language document in order to find the top-n candidates from the target-language index. Then we measured the distances between the embedded word vectors in addition to using the text similarity between the source and each of the retrieved n candidate target-language documents. Our approach produced improvements over the Jaccard similarity-based baseline system for both the Euronews and the WMT-2016 data. Furthermore, using the inverted index-based approach in CLIR results in a linear time complexity, which is much less than the Jaccard similarity-based approach that runs with quadratic time complexity. Finally, the performance is further enhanced by the application of the word embedding-based similarity measurements. *FaDA* achieved a recall value of 0.6619 which is a massive 127.45% relative improvement over our previous alignment system that obtained a recall value of 0.291. We found that our alignment system can be efficiently applied to a large collections of documents as it does not require any MT system.

In this chapter, we addressed our first research question (RQ-1) that involves bilingual document alignment using comparable corpora. We discovered the problems with our naive approach in the first experiment and then applied a much more sophisticated method in the second experiment. Our document alignment system is now capable of achieving good quality alignments and can effectively be applied to a large bilingual comparable corpus. It is very useful to implement such a bilingual document alignment system that can exploit comparable corpora in order to extract semantically similar sentences, phrases or segments in two different languages. We believe that our document alignment system will benefit research in parallel/similar data extraction and draw the attention of researchers in this field of study.

In the next chapter, we will discuss 'parallel data extraction' from comparable corpora. We will show how *FaDA* can be effectively transformed into a sentiment aligner using our proposed approach and can be used to extract parallel or semantically similar sentences from comparable corpora.

Chapter 5

Parallel Data Extraction

In the previous chapter, we discussed the implementation of a bilingual document alignment system based on word embeddings and text similarity. In this chapter, we discuss our second research question that deals with 'parallel data extraction' from comparable corpora.

Building a robust MT system requires a sufficiently large parallel corpus to be available as training data. Usually, there are two ways of parallel corpus acquisition, namely: (i) manual development, and (ii) automatic extraction. Although manual development is ideal and is produced in most cases by human translators, this process requires a huge amount of time and effort which is considered to be less practical than automatic extraction of parallel data for MT. One of the easiest ways to accomplish this task is to employ an MT system that translates all the source-language texts into the target language and then performs text similarity in the target language. However, using an MT system is not always the best solution mainly due to the following reasons: (i) it requires a significant amount of time to build the MT system itself, especially if this is an NMT system, (ii) it also takes a long time to translate all the source-language documents into the target language especially for a large document, and (iii) MT systems for all domains and language pairs are not available. These problems demonstrate that finding a suitable alternative to using an MT system for parallel data extraction is an important aim.

In this chapter, we propose to automatically extract parallel sentences from a comparable corpus without using any MT system or even any parallel corpus at all. Instead, we use CLIR, word embeddings, text similarity and a bilingual dictionary, thus saving a significant amount of time and effort as no MT system is involved in this process. The automatically extracted sentence pairs are then added to the already available parallel training data and an extended translation model is built from this concatenated data. Finally, we compare the performance of our new extended model against a baseline model built from the available data.

5.1 Related Work

The extraction of parallel sentences/segments plays an important role in improving MT (Wolk et al., 2016; Hangya and Fraser, 2019). Many works address the issue of parallel data extraction in different ways. For example, Ling et al. (2014) propose a crowdsourcing approach to extract parallel data from tweets. Instead of translating the texts, they attempt to find the translations in tweets. Šubert and Bojar (2014) propose a "Twitter Crowd Translation" infrastructure for translating tweets. Their approach is based on (i) following certain tweet sources, (ii) managing registrations of volunteer translators, (iii) delivering requests and collecting translations from them, (iv) operating a manual evaluation of the translations and (v) publishing the best translation back to Twitter. Chu et al. (2015) extract both parallel sentences and fragments from comparable corpora to improve PBMT by applying parallel sentence extraction to identify parallel sentences from comparable sentences and then extract parallel fragments from the comparable sentences. They select Chinese–Japanese Wikipedia for these experiments in order to verify the effectiveness of their approach. Gupta et al. (2014) apply a domain-biased parallel data collection and a structured methodology to obtain parallel data for the

English–Hindi language pair. Recently, deep learning has gained popularity in this task (Bouamor and Sajjad, 2018; Grégoire and Langlais, 2018). Some work exploits MT for parallel data extraction (Chu, 2015; Ruiter, 2019).

As an alternative to parallel data, a comparable corpus is considered as a valuable resource for MT. For example, Afli et al. (2015) use a multimodal comparable corpus of audio and texts built from 'Euronews'[1] and 'TED'[2] web sites in order to extract parallel data. Karimi et al. (2018) present a bidirectional method to extract parallel sentences from English and Persian document-aligned Wikipedia. They use two MT systems to translate from Persian to English and the reverse after which an IR system is used to measure the similarity of the translated sentences. Guo et al. (2019) propose multilingual document embeddings for nearest neighbour mining of parallel data. Bañón et al. (2020) release the largest publicly available parallel corpora for many language pairs by crawling a large number of web sites using open source tools. Zhang et al. (2020) use pre-trained language models to filter out noisy sentence pairs from web-crawled corpora. They also release a large Japanese-Chinese web-crawled parallel corpus.

Although many parallel data extraction systems are based on using MT systems, it is not always a good idea as we already mentioned earlier and so we simply discard the requirement of any MT system and any parallel data at all.

5.2 Data sets

We use different data sets for our two experiments. The experiment-1 (details in Section 5.4.1) which is based on average word vector and text similarities uses two data sets, namely: *Euronews* and *News commentary* corpus. In contrast, we use the *FourSquare* and the *Hotel review* corpus for experiment-2 (details in Section 5.4.2)

[1] https://www.euronews.com/

[2] https://www.ted.com/

which adopts similar approaches as experiment-1 with few additional pre-processing steps.

5.2.1 Euronews and News commentary corpus

- **Euronews corpus:** The *Euronews* corpus[3] (Afli et al., 2017a) is a multimodal corpus of comparable documents and their images. In our experiments, we consider only the documents and not the images as this is beyond the scope of this research. Each document in *Euronews* corpus consists of at least one line of text, while many of them contain multiple-line texts with multiple sentences. As our main goal is to find parallel data at sentence level, we split these documents into multiple sentences and consider each sentence separately. The *Euronews* corpus is used as a comparable corpus in experiment-1.

- **News commentary corpus:** Another data set we use in experiment-1 is the English–French parallel sentence pairs from the 'News Commentary' corpus.[4] We refer to this data as *News* and use it as the baseline parallel corpus in experiment-1.

Table 5.1 shows the statistics of the *Euronews* and the *News* data. Note that in the *Euronews* data, $644K$ English and $614K$ French sentences are obtained from $40K$ English and $37K$ French documents, respectively.

Data set	**Language**	# **Documents**	# **Sentences**
Euronews	English	$40,421$	$644,226$
	French	$37,293$	$614,928$
NewsComm	English	/	$246,946$
	French	/	$246,946$

Table 5.1: Data statistics of Euronews and News commentary corpus

[3] https://github.com/loharp/FaDA/tree/master/euronews-data

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

5.2.2 FourSquare and hotel review corpus

- **FourSquare parallel corpus:** As we already described in Section 3.2.4, the *FourSquare* corpus contains over $11K$ restaurant reviews (or $18K$ sentences) in English and their manual translations into French.

- **Hotel review corpus:** The *Hotel_Review* corpus[5] consists of $878K$ reviews from $4,333$ hotels crawled from *TripAdvisor*. Although most of the reviews are in English, some are also written in French. The statistics of the *FourSquare* and the *Hotel_Review* data sets are shown in Table 5.2.

Data set	# **Reviews**	# **Total sentences**	# **training**	# **Dev**	# **Test**
FourSquare	$11,551$	$17,945$	$14,864$	$1,243$	$1,838$
Hotel_Review	$878,561$	/	/	/	/

Table 5.2: Statistics of the FourSquare parallel and the Hotel review data sets

5.3 MT system

The MT models are built using the freely available open source NMT toolkit 'OpenNMT'[6] (Klein et al., 2017). In our experiments, we use the default parameter settings: *RNN* as the default type of encoder and decoder, $word\_vec\_size = 500$, $rnn\_size = 500$, $rnn\_type = LSTM$, $global\_attention\_function = softmax$, $save\_checkpoint\_steps = 5000$, $training\_steps = 100,000$ etc. We evaluate the translation quality using BLEU (Papineni et al., 2002).

[5] `https://www.cs.cmu.edu/~jiweil/html/hotel-review.html`

[6] `https://github.com/OpenNMT/OpenNMT-py`

5.4 Experiments

5.4.1 Experiment-1: Parallel data extraction from Euronews corpus

In this experiment, we use the English–French comparable *Euronews* and parallel *News* data with French as the source and English as the target language. We combine the (i) CLIR, (ii) text similarity and (iii) word embedding-based systems in this task. The CLIR component used in this experiment is a part of the open source bilingual document alignment tool *FaDA* (Lohar et al., 2016b) which is explained in detail in Section 4.3.4.1. As *FaDA* is a document-level alignment tool, we represent each sentence as a document and so it works at sentence level in this experiment. Having said that, initially *FaDA* retrieves a set of suitable candidate English documents for each French document. Afterwards, we proceed with our proposed approach in the following steps. Firstly, all the content words (i.e. after removing the stopwords) of a French document are translated[7] using a French–English bilingual dictionary.[8] We use multiple translations of a word because its translation may be different depending upon the context. Secondly, each of the extracted candidate English documents is then compared with the English (word-level) translation of the French document using the text similarity and average word vectors. The English document (i.e. the sentence) with the highest similarity score is selected as the parallel counterpart of the French document (sentence). The extracted sentence pairs are then treated as the additional parallel training data for building MT systems. Note that all the experiments for 'Para data extraction' were conducted during the later phase of this research. For this reason, we use only NMT to build the translation models as it has already become the state-of-the-art and performs better than PBMT in many scenarios.

[7] This is only a word-to-word translation using a bilingual dictionary, not a generic MT system.

[8] The dictionary is available at: `www.seas.upenn.edu/~nlp/resources/TACL-data-release/dictionaries.tar.gz`

5.4.1.1 System description:

Our proposed system is composed of the following components: (i) CLIR-based system, (ii) sentence length-based pruning, (iii) average word embeddings, (iv) text similarity, and (v) score combination.

• **CLIR-based system:** The workflow of the CLIR-based system (see the architecture in Section 4.4) can be described in the following steps: (i) the source-language and the target-language documents are indexed, (ii) each source-language document is used to construct a pseudo-query[9] which is considered as suitably representative of the document, (iii) all pseudo-query terms are translated into the target language by a bilingual dictionary and the translated query terms are then searched in the target-language index, and finally (iv) the *top-n*[10] target-language documents are retrieved.

• **Sentence length-based pruning:** Prior to performing the text- and the word embedding-based similarities between the source- and the target-language sentences, we exclude some of the comparisons depending upon the sentence-length ratio.

This ratio is calculated in terms of the total number of words in the word translations of the source-language document (sentence) and the total number of words in the target-language document (sentence). We set the threshold for this ratio to 0.5, which means that the shorter of the document pair must be at least half the of the longer document in terms of the total number of words present in them. For example, if a French document contains 5 words and an English document contains 20 words, the ratio is 0.25 which is less than the threshold of 0.5. This document pair, therefore, according to our criteria is less likely to be parallel and hence should not be considered for comparison. The French document must contain at least 10 words to pass this threshold in order to be considered for further similarity mea-

[9] A *pseudo-query* is the modified form of a user's original query in order to improve the ranking of retrieval results compared to the original query.

[10] We use the default value of n used in *FaDA*, where $n = 10$.

surements. However, 0.5 is not an empirically determined threshold; we choose this value so that very unlikely candidates can be removed from the comparison, albeit some of the invalid pairs still pass the threshold.

In general, the average length ratio of English texts over the French translations is near 1.0 (Chen, 2003) but there are many examples that violate this. For example, consider the English sentence 'I like to propose a toast.' that contains 6 words and its equivalent French translation 'J'aime proposer un toast' that contains 4 words. The sentence-length ratio in this case is below 0.7, so setting a high threshold very close to 1.0 may result in discarding many valid sentence pairs like this one.

- **Text similarity:** We calculate text similarity using the following steps: (i) we remove all the stopwords from both the French and English documents; etc. (ii) we translate the remaining content words of the French document into English using a French–English bilingual dictionary. (iii) some of the word translations still contain stopwords such as *to*, *of* etc. We remove these stopwords. (iv) finally, we calculate the total number of word matches between the words in the English document and the word-level English translation of the French document.

- **Average word vector similarity:** We use the widely popular open source tool 'word2vec' (Mikolov et al., 2013) to find the similarity between the average word vectors of the English document and the word-level English translations of the French document. This process helps to identify semantically related words in close proximity to one another in multi-dimensional space. To illustrate this with an example, consider Figure 5.1 that shows a collection of words represented in a two-dimensional space. We can observe that semantically equivalent words are placed in close proximity to one another. For example, the words *electrical*, *electricity*, *electric* etc. are closely grouped together in the same region. However, this figure shows the simplest representation of how the related words are treated. In reality,

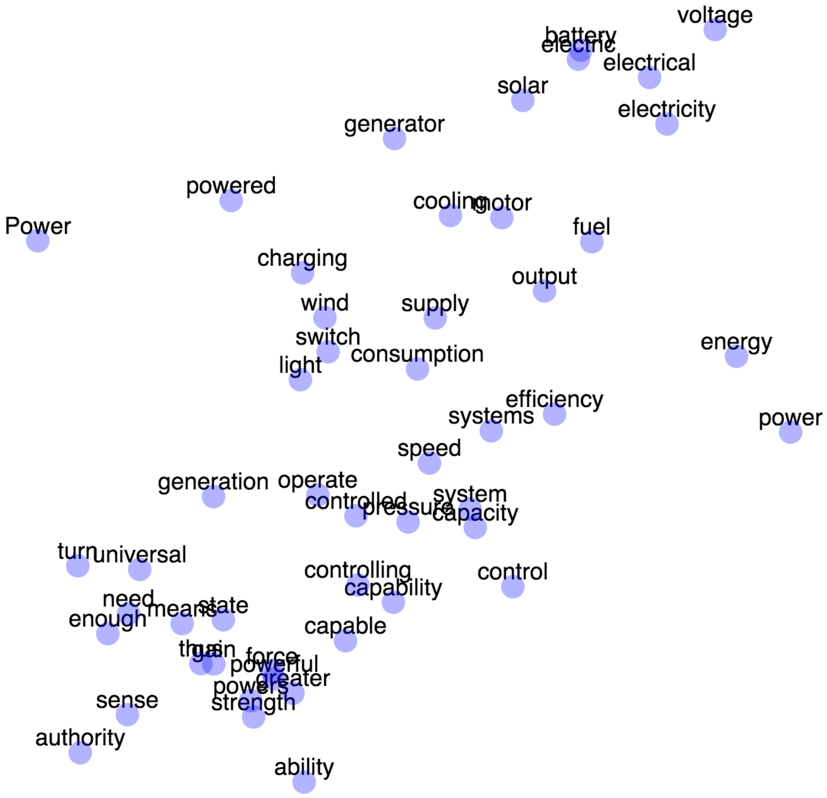


Figure 5.1: Example of semantically related words in two-dimensional space

the words are represented as a vector of real values in much higher dimensions. In pre-trained word embeddings, the semantically related words usually contain similar vector values. Note that, our approach of word vector similarity measurement in this experiment is different from the word-embedding approach of *FaDA* which was described in Section 4.3.4.2. *FaDA* uses the concept of cluster of words for a document where the distance between the query word vector and the word vector of cluster centroid is calculated. In contrast, we calculate the average word vector value to measure the similarity in this experiment.

We now discuss how the average word vectors are actually calculated. Let us consider a sentence S with a sequence of n words: $w_1, w_2, w_3.....w_n$. Let the vector embeddings of the words be $u_{w_1}, u_{w_2}, u_{w_3}.....u_{w_n}$.

The average word embedding of S is calculated using Equation (5.1)

$$U_s = \frac{1}{n}\sum_{i=1}^{n} u_{w_n} \tag{5.1}$$

In order to obtain the word embeddings for our experiment, we use the whole collection of English texts in the *Euronews* corpus because it also belongs to the news domain and is a relevant resource for this experiment. The *Euronews* corpus consists of around $40K$ English documents, each of which contains at least one line of text. We combine the contents of all the documents into a single text file which is then fed as input to the word-vector module to obtain the word embeddings. We use the default parameter settings for word vector training provided in the original training script.[11]

The average word embedding is calculated using the following steps: (i) all the stopwords in both the word translations of the French document and the English document are removed, (ii) the real word vector values of all the remaining words in the word translations of the French document are retrieved and then the average of all these vector values is calculated, (iii) the average word vector values for the English document are calculated in a similar manner, and (iv) the two averages are compared in order to calculate the average word vector similarity.

- **Score combination:**

Once we calculate the text and the average word vector similarities, these scores are then combined to obtain the overall similarity score. Up to now, the combination has been done by assigning equal weights to each of these similarity scores. However, exploring other combinations is planned to tune our proposed system. The overall similarity score S_{sim} is calculated using Equation (5.2)

$$S_{sim} = w_1 WV_{sim} + w_2 Text_{sim} \tag{5.2}$$

[11] `https://github.com/dav/word2vec/tree/master/scripts`

Here, WV_{sim} and $Text_{sim}$ are the average word vector and the text similarity scores with w_1 and w_2 weight values, respectively. Note that $w_1 = w_2 = 0.5$, which means that both of the similarity measurements are given equal importance, as mentioned above.

5.4.1.2 Experimental setup

- **Sentence-level document alignment:**

We store each sentence of the *Euronews* corpus in a single document which results in creating more than $600K$ documents per language. These documents are then fed as input to the CLIR component of *FaDA*. Once the top-n English documents are obtained for a French document, we aim to find its closest semantically equivalent English document. It is, therefore, expected that the total number of extracted sentence pairs is over $600K$. However, it is impractical to consider all these sentence pairs as parallel data because many of them are not semantically equivalent. Accordingly, we extract only those pairs that have average word vector similarity scores greater than a particular threshold. We explored different values to determine this threshold and found that 0.55 is the optimal value (see detailed explanation in Section 5.4.1.3), i.e. selecting the sentence pairs with a similarity score greater than 0.55 and adding them with the *News* data gives the best improvement in BLEU score over the Baseline (discussed in detail in Section 5.4.1.3). Table 5.3 shows the data size of the *News* data and the extracted sentence pairs from the *Euronews* corpus.

Data set	# **Sentences**
NewsComm	$246,946$
Sentence pairs (Euronews)	$31,860$

Table 5.3: News commentary data vs extracted sentence pairs from Euronews

- **MT systems:**

The extracted parallel sentences from the *Euronews* corpus are used as additional data for MT training. We build two translation models in our experiments, the

Baseline and the *Extended* model. The *Baseline* is built using only the *News* data whereas the *Extended* model is built using the concatenated data. We held out $1,000$ sentence pairs for development and another $1,000$ sentence pairs for testing purposes from the *News* data. Table 5.4 shows the data distribution. Note that the number of training sentences in the *News* data in this table is far less than that in Table 5.3 because of filtering out the long sentences containing more than 80 words. The second row in Table 5.4 shows the sum of filtered *News* data and filtered sentence pairs extracted from the *Euronews* data. Each translation model is tuned and tested on the development and test sets (held out from the *News* data) shown in this table.

Model	**Data set**	**# training**	**# Dev**	**# Test**
Baseline	News	$226,946$	$1,000$	$1,000$
Extended	News + Euronews	$253,592$	/	/

Table 5.4: Data distribution for two different MT models

5.4.1.3 Results

Translation model	**BLEU**
Baseline model	27.1
Extended model	**27.5**

Table 5.5: Performance comparison: Baseline vs Extended model

The results are shown in Table 5.5. We can observe that the addition of parallel sentences extracted from the *Euronews* corpus using our proposed system improves the BLEU score, i.e. the Baseline is outperformed by the *Extended* model by 0.4 BLEU points. We also perform the statistical significance testing using MultEval (Clark et al., 2011). However, we found that this improvement is not statistically significant as $p = 0.16$, which is slightly greater than 0.1.

It is to be noted that the above result is obtained when we concatenate the additional sentence pairs that have higher similarity score than the threshold of 0.55

(discussed in Section 5.4.1.2). However, we explored different threshold values starting from 0.45 to 0.65.

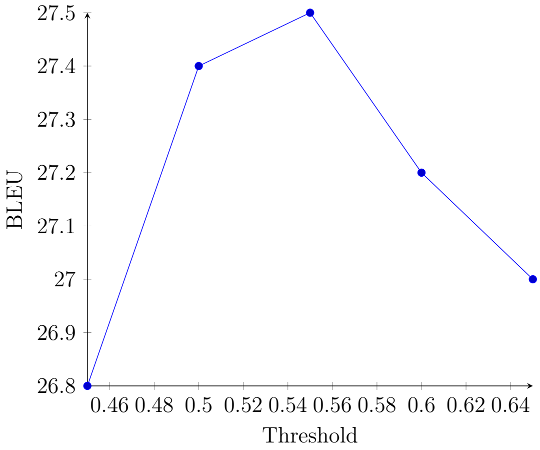


Figure 5.2: BLEU scores for different threshold values

The reason why we do not apply any threshold outside this range is because we found that many sentence pairs pass if we allow a lower threshold, many of which are not actually parallel or semantically equivalent to each other and hence become noise if we concatenate them with the *News* parallel training data.

Furthermore, if we consider a higher threshold (say 0.7), very few sentence pairs pass this threshold. Therefore, only a few extracted sentence pairs would be too small to be added to the existing training data in order to lead to an improvement in the BLEU score. Figure 5.2 shows the BLEU score comparison when 5 different threshold values (0.45, 0.5, 0.55, 0.6 and 0.65) are used. It is clear from the figure that the BLEU score decreases as the threshold is reduced or increased from 0.55. The size of the extracted corpora at all of the thresholds are shown in Table 5.6.

The improvement in BLEU score shows that our automatic parallel data extraction system helps improve MT quality by supplying additional training data. However, this is the beginning phase of our experiment and further plans are made to extend this work.

Threshold value	# Parallel sentences
0.45	73, 211
0.5	42, 235
0.55	26, 646
0.6	18, 108
0.65	17, 923

Table 5.6: Amounts of data obtained from *Euronews* corpus using 5 different threshold values

As of now, we illustrate some example outputs where the *Baseline* model is outperformed by our *Extended* model in Table 5.7.

Example	Reference	Baseline model	Extended model
1	But, equally important, workers organized themselves to defend their interests.	But, as important, workers have been organized to defend their interests.	But, equally important, workers have been organized to defend their interests.
2	Overall, however, the inequality gaps are large and, in many cases, growing.	Overall, however, the inequality gap remains acute and in some cases even expansion.	Overall, however, the inequality gap remains deep, and in some cases it expands.
3	Countries that import currently subsidized food will be worse off.	Countries that imports currently will suffer.	Countries that import products currently subsidized will suffer.
4	He ate chocolate and watched NBA games.	He ate chocolate and watched from the NBA games.	He ate chocolate and watched the NBA games.

Table 5.7: Example outputs (French-to-English): Baseline vs Extended model

In example 1, the word '*equally*' is missing in the *Baseline* output. The second example shows that the ending phrase '*in some cases even expansion*' of the output produced by the *Baseline* model is grammatically incorrect whereas the *Extended* model produces the phrase '*in some cases it expands*' which is grammatically correct and semantically equivalent to the phrase '*in many cases, growing*' in the reference translation.

In example 3, both translation outputs are erroneous but the output produced by the *Extended* model is better as it includes the word '*products*' which although is not equivalent to the word '*food*' in the reference translation but at least conveys a little bit of similar meaning. Finally, example 4 shows the case where both translation outputs are mostly correct except the presence of extra prepositions. The phrase ‘*watched the NBA games*’ that is produced by the *Extended* model is better than the phrase ‘*watched from the NBA games*’ produced by the *Baseline* when compared with the reference translation.

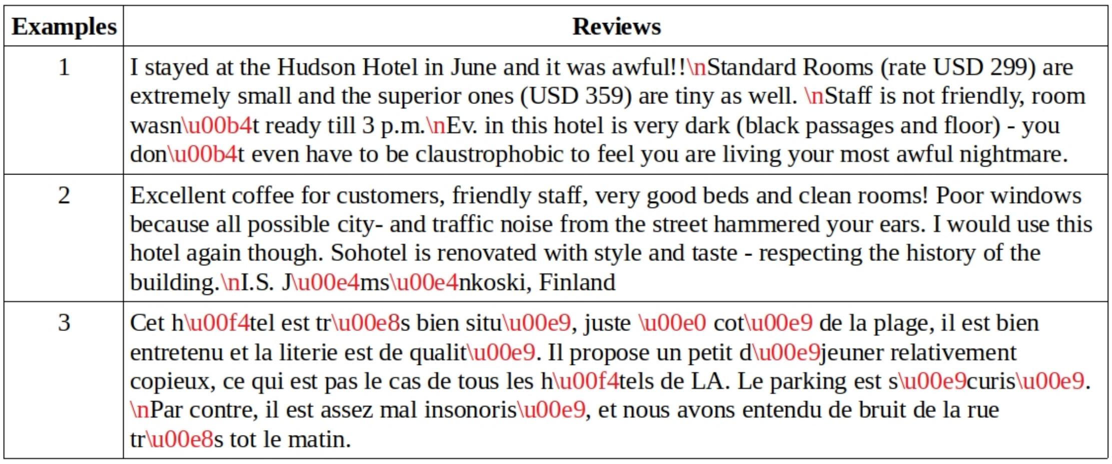
5.4.2 Experiment-2: Extracting parallel data from hotel reviews

In this experiment, we attempt to extract parallel data for UGC, in this case, from the hotel reviews. The data set (described in detail in Section 5.2) consists of a large collection of hotel reviews from 4,333 hotels collected from *TripAdvisor*. Our objective is to extract parallel or semantically similar sentence pairs from this data set and add them to the existing parallel French–English *FourSquare* corpus of restaurant reviews (discussed in Section 5.2) in order to build a translation model with larger training data. It is to be noted that the *Hotel_Review* data set is merely a collection of user reviews written mostly in English and sometimes in French. Initially we extract the French reviews from the whole collection and split them into sentences. In a similar manner, the English sentences are formed from the English reviews. Afterwards, for each French sentence, we find its most semantically equivalent English sentence (i.e. in some cases) using our proposed approach. Finally, these extracted English sentences are used as the parallel counterpart of the French sentences in order to create the additional parallel data for MT training. Although the main methodologies in this experiment are adopted from experiment-1, we introduce some extra steps beforehand. We discuss these steps in the following sections.

5.4.2.1 Data preprocessing:

The *Hotel_Review* data set is provided as a single file where most of the reviews are in English but some French reviews are also mixed with them. Table 5.8 shows three randomly selected example reviews (two English and one French) from this data set.

We highlight the special characters such as newlines and unicode characters in red.



Examples	Reviews
1	I stayed at the Hudson Hotel in June and it was awful!!\nStandard Rooms (rate USD 299) are extremely small and the superior ones (USD 359) are tiny as well. \nStaff is not friendly, room wasn\u00b4t ready till 3 p.m.\nEv. in this hotel is very dark (black passages and floor) - you don\u00b4t even have to be claustrophobic to feel you are living your most awful nightmare.
2	Excellent coffee for customers, friendly staff, very good beds and clean rooms! Poor windows because all possible city- and traffic noise from the street hammered your ears. I would use this hotel again though. Sohotel is renovated with style and taste - respecting the history of the building.\nI.S. J\u00e4ms\u00e4nkoski, Finland
3	Cet h\u00f4tel est tr\u00e8s bien situ\u00e9, juste \u00e0 cot\u00e9 de la plage, il est bien entretenu et la literie est de qualit\u00e9. Il propose un petit d\u00e9jeuner relativement copieux, ce qui est pas le cas de tous les h\u00f4tels de LA. Le parking est s\u00e9curis\u00e9. \nPar contre, il est assez mal insonoris\u00e9, et nous avons entendu de bruit de la rue tr\u00e8s tot le matin.

Table 5.8: Review examples

Note that the newlines characters are not always explicitly present even if a new sentence starts. For instance, in example 2, there are no newline characters before the sentences such as '*Poor windows....*' and '*I would use this....*'. In addition, plenty of unicode characters are present in the hexcode format such as '00b4', '00e9', '00e8' etc. most of which are present in the French review in example 3.

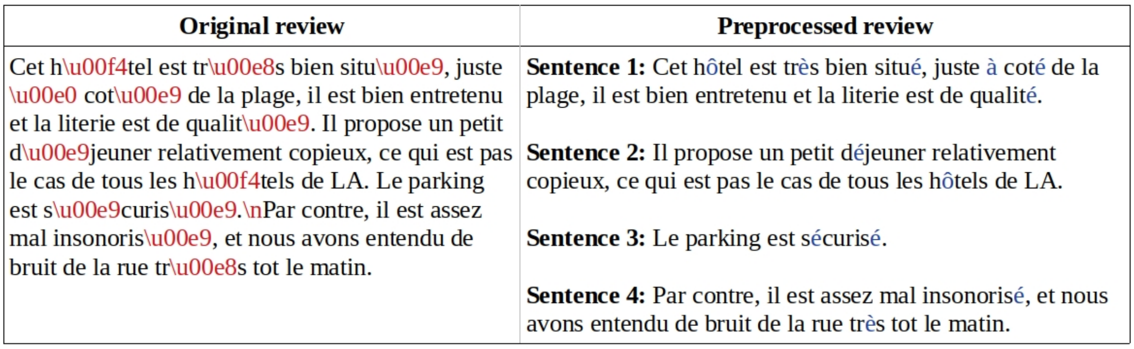
Considering the above observations, we preprocess the data using the following steps.

- **Language detection:** We perform language detection[12] in order to detect and extract the English and French reviews from this data set.
- **Sentence splitting:** As our parallel data extraction system is implemented at sentence level, we split the multi-sentence reviews into different parts (sentences) and consider each part as a single document.
- **Unicode conversion:** We convert[13] the characters given in unicode format into the Latin characters. For example, the character '00f4' is converted into 'ô'.

Table 5.9 shows an original French review (example 3 of Table 5.8) and its preprocessed version. We highlight all the unicode characters in the original review in red and the converted characters in the preprocessed review in blue.

[12] `https://pypi.org/project/langdetect/`

[13] Unicode representation of these characters can be found at: `http://www.fileformat.info/info/unicode/char/search.htm`



Original review	Preprocessed review
Cet h\u00f4tel est tr\u00e8s bien situ\u00e9, juste \u00e0 cot\u00e9 de la plage, il est bien entretenu et la literie est de qualit\u00e9. Il propose un petit d\u00e9jeuner relativement copieux, ce qui est pas le cas de tous les h\u00f4tels de LA. Le parking est s\u00e9curis\u00e9.\nPar contre, il est assez mal insonoris\u00e9, et nous avons entendu de bruit de la rue tr\u00e8s tot le matin.	**Sentence 1:** Cet hôtel est très bien situé, juste à coté de la plage, il est bien entretenu et la literie est de qualité. **Sentence 2:** Il propose un petit déjeuner relativement copieux, ce qui est pas le cas de tous les hôtels de LA. **Sentence 3:** Le parking est sécurisé. **Sentence 4:** Par contre, il est assez mal insonorisé, et nous avons entendu de bruit de la rue très tot le matin.

Table 5.9: An example review before and after preprocessing

Note that 4 sentences are generated from this single review after preprocessing.

5.4.2.2 Experimental setup

• **Data setup:** We already showed the data statistics in Table 5.2 in Section 5.2 but to a less fine-grained extent, especially for the *Hotel_ Review* data set. As mentioned earlier, $878K$ reviews are collected from $4K$ hotels where most of the reviews contain multiple sentences. We obtain millions of English sentences from this data set, each of which can be stored in a single document, thus creating millions of English documents.

However, we found that *FaDA* was unable to index these millions of English documents. Accordingly, we select only a part of the English reviews that contains around $984K$ documents. In case of such an indexing failure, it is possible to apply either of the following two strategies: (i) indexing the corpus in parts and combining the indexes, or (ii) indexing each part and then finding alignments of a source-language document from each index separately in order to consider it for further similarity measurements. We plan to include these strategies in our future experiments. However, all of the $139K$ French documents from this data set are included in this experiment because *FaDA* was capable of indexing all of them. Table 5.10 shows a detailed distribution of these data sets.

Data set	Language	# Sentences	# training	# Dev	# training
FourSquare	English	17,945	14,864	1,243	1,838
	French	17,945	14,864	1,243	1,838
Hotel_Review	English	984,319	/	/	/
	French	139,069	/	/	/

Table 5.10: Detailed data distribution of the FourSquare parallel and the Hotel review data sets used in this experiment

- **Sentence-level document alignment:** This approach is similar to what we described in Section 5.4.1.2, i.e. for each French sentence, we extract its equivalent English sentence using the CLIR- and the word embedding-based similarity in combination with the text similarity. Finally, the sentence pairs whose overall similarity scores are equal to or greater than a threshold are considered as parallel data.

- **Building MT models:** Once the sentence pairs are extracted from the *Hotel_Review* data set, we consider them as additional parallel resource and concatenate them to the parallel training sentences of the *FourSquare* corpus. As a baseline system we use the *Baseline* model for UGC text (already discussed in Section 3.4.4), which is built from the 14,864 parallel training sentences of the *FourSquare* corpus. In addition, we build *Extended* models, which are trained from the concatenation of the *FourSquare* data and the different sets of parallel sentence pairs extracted from the *Hotel_Review* data set.

As in experiment-1, we also explore different threshold values to extract different sets of parallel sentence pairs. As expected, using low threshold values results in allowing too many sentence pairs, whereas a high threshold value discards too many sentence pairs. Considering this situation, we explore only a few threshold values that are neither too low nor too high and obtain different sets of parallel sentences accordingly.

Table 5.11 shows 5 different amounts of parallel sentences obtained using 5 different threshold values starting from 0.3 up to 0.7.

Threshold value	**# Parallel sentences**
0.3	34, 829
0.4	14, 697
0.5	6, 188
0.6	2, 183
0.7	778

Table 5.11: Amounts of data obtained from *Hotel_ Review* corpus using 5 different threshold values

We add each of these sets separately to the parallel training data of the *FourSquare* corpus and build 5 different *Extended* models.

5.4.2.3 Results

Let us first describe how BLEU score varies with the different extended models. Figure 5.3 shows the BLEU score variation using 5 *Extended* models, i.e. the translation models built using the 5 different sets of additional data separately concatenated with the *FourSquare* corpus.

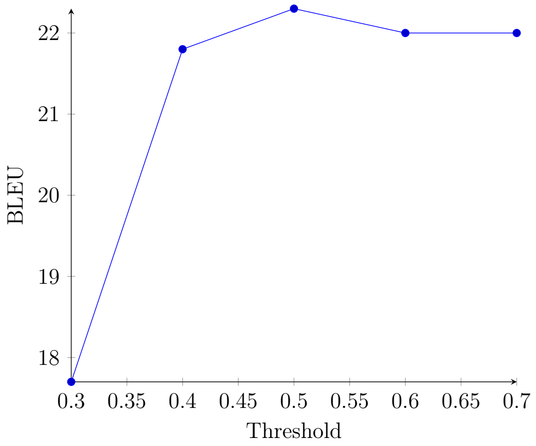


Figure 5.3: BLEU score comparison for different threshold values

We can observe that the highest BLEU score (22.3) is obtained using the threshold value of 0.5 which is, therefore, selected as the optimal threshold. However, for the sake of clarity, we provide the BLEU scores produced by the 5 *Extended* models in Table 5.12.

Threshold value	**Model name**	**BLEU score**
0.3	Ext-1	17.7
0.4	Ext-2	21.8
0.5	**Ext-3**	**22.3**
0.6	Ext-4	22.0
0.7	Ext-5	22.0

Table 5.12: BLEU scores achieved by translation models using different threshold values

As we obtain the highest BLEU score of 22.3 for the *Extended* model (Ext-3) using 0.5 as the threshold value, we consider it as the optimised *Extended* model. The next step is to compare its performance with the *Baseline* model which is built from only the *FourSquare* training data. Table 5.13 shows the final result. Note that we already showed the Baseline result in Section 3.4.4, which we are now comparing with the result of the *Extended* model.

Translation Model	**BLEU**
Baseline model	22.1
Ext-3 model	**22.3**

Table 5.13: Baseline vs Extended model

It is worth noting that instead of optimising the threshold using a development set, we directly apply the different thresholds on the test set. Our objective is to investigate what performance can be achieved applying all these thresholds directly on the test data.

We notice a slight improvement in BLEU score (0.2 points) over the *Baseline* model. However, this improvement is not statistically significant as $p = 0.61$, which is greater than 0.1. Moreover, this is less than the improvement of 0.4 BLEU points observed in experiment-1 (Section 5.4.1). One probable reason for this degradation is that the *Euronews* data set used in experiment-1 actually contains some parallel texts. Therefore, extracting and adding them to the existing *News* parallel training data helps improve the BLEU score to some extent. In contrast, the *Hotel_Review* data set does not contain parallel text. In fact, the reviews are generated randomly by different users without translation usage foreseen. However, there exist some texts that are very close in meaning even though they are not parallel. For this reason, such partially semantically similar texts help improve the BLEU score only very slightly. We show some example outputs where the *Baseline* model is outperformed by the best of our *Extended* models (Ext-3) in Table 5.14.

Example	Reference	Baseline model	Extended model
1	Cozy little teahouse, amazing sweets and teas.	Disgusting room, very good cakes and teas.	Small tea room, very good cakes and teas.
2	A nice atmosphere to hang out with friends.	Friendly atmosphere for a relaxed dinner.	Friendly atmosphere for a walk with friends.
3	The sales assistants are super friendly.	The sales assistants are really welcoming.	The sales assistants are super welcoming.
4	Their famous hot chocolate, one of the best in the world, is worth the wait!	Its suggestion hot chocolate, one of the best in the world is worth the wait!	Its legendary chocolate, one of the best in the world is worth the wait!
5	They do really good burgers.	They serve very good burgers.	They do very good burgers.

Table 5.14: Example outputs: Baseline vs Extended model

We can notice in the table that although the phrase '*Small tea room*' (in example 1) produced by the *Extended* model is not a proper translation, it is still much better than the completely wrong translation output '*Disgusting room*' produced by the *Baseline* model. In example 2, the phrase '*hang out with friends*' in the reference translation is semantically closer to the phrase '*walk with friends*' (produced by the *Extended* model) than to the phrase '*relaxed dinner*' (produced by the *Baseline* model). Moreover, '*super friendly*' is more synonymous to '*super welcoming*' than '*really welcoming*' in example 3. Furthermore, the word '*suggestion*' (see example 4) is completely meaningless when used before '*hot chocolate*' that is produced by the *Baseline* model. In contrast, although '*legendary chocolate*' is not a proper translation (produced by the *Extended* model), it is partially similar to '*famous hot chocolate*' in the reference. Finally, both of the translation outputs in example 5 are sensible but the output produced by the *Extended* model is closer to the reference.

5.5 Conclusions

In this chapter, we proposed a parallel data extraction technique from comparable corpora in order to generate additional parallel training data for MT. Many research works employ MT itself to ease this task. However, it is not always a practical solution because in addition to building the MT system in the first place, it also requires a huge amount of time to translate all the source-language documents of the comparable corpus into the target-language in order to be able to perform the text similarity in the target language. Moreover, using an MT system results in computational overhead as it requires training the translation model itself prior to translating the source-language documents. To overcome this situation, we implemented a parallel data extraction system without any help from MT or even any parallel corpus.

In the first experiment, we initially used the CLIR component of *FaDA* tool to extract the candidate target-language sentences for a source-language sentence. We then used the average word-embeddings and text similarity with the help of a bilingual dictionary in order to obtain parallel sentences from the *Euronews* corpus. These extracted sentence pairs were then concatenated to the existing parallel training data to build an extended translation model which outperformed the baseline system that is built from only the existing parallel training data.

In our second experiment, we attempted to extract parallel data from a comparable corpus of UGC, in this case, hotel reviews. This experiment is very similar to the first experiment in terms of the methodologies used except that we used some extra preprocessing steps in this experiment. Firstly, we built a *Baseline* translation model using a small amount of parallel training data from the *FourSquare* corpus. Secondly, we extracted 5 different sets of parallel sentence pairs from the *Hotel_Review* data set using 5 different threshold values for similarity score. Thirdly, we added each of these 5 sets of additional training data separately to the *FourSquare* training data and built 5 different *Extended* translation models from these data sets. Then, we compared the BLEU scores obtained by these models and found that the threshold value of 0.5 yielded the highest BLEU score of 22.3. Finally, we selected this model as the optimal *Extended* translation model and compared its performance against the *Baseline* model. We obtain a slight improvement in BLEU score (0.2 points) over the Baseline.

We noticed that extracting parallel texts from the *Euronews* corpus in the first experiment obtains a slightly higher BLEU score improvement than for the *Hotel_Review* data set in the second experiment. One probable reason is that the *Euronews* data actually contains some parallel texts and so extracting and adding them to the existing *News* parallel training data helps improve the BLEU score to some extent. In contrast, the hotel reviews are extremely unlikely to contain parallel texts as they are randomly generated by different users without translation usage in mind. Although not being strictly parallel, some of them are semantically equivalent, and adding them as extra training data improves the BLEU score very slightly over the *Baseline* model. It is, therefore, expected that the BLEU score can be improved further if there exists a considerable amount of parallel texts in a comparable corpus of UGC.

As we did not use any MT system or any parallel corpus for this task, our proposed system is very simple and can be easily applied to a comparable corpus. Our findings in this research are encouraging as our system relies on only the text similarity, word embeddings and a bilingual dictionary, for which the required resources are easily available online. We believe that our proposed model has the potential to benefit further research in this field.

In the next chapter, we will discuss our third and final research question that deals with one of the most recent applications of UGC translation, namely 'sentiment preservation in MT'.

Chapter 6

Sentiment Preservation

In the previous chapter, we discussed our second research question (RQ-2) that deals with parallel data extraction from comparable corpora. We conducted experiments by using CLIR, word embeddings, text similarity and bilingual lexicon to extract parallel sentences from bilingual comparable documents. In this chapter, we will focus on sentiment preservation, which is one of the most recent applications of UGC translation.

The world has undergone huge evolution with the rapid development of web technology. As a result, there has been a huge growth in the use of social media in the last few years. Internet users are now capable of communicating among themselves from every part of the world. Interacting with social media is nowadays an everyday occurrence for most people. They often generate and share information in the form of UGC that can be categorised into different modalities such as, audio, video or text, e.g. the audio and video files uploaded by the users on 'Youtube',[1] or tweets, feedback and online reviews, all of which are in textual format. A massive amount of UGC is generated every day from all over the world.

[1] https://www.youtube.com/

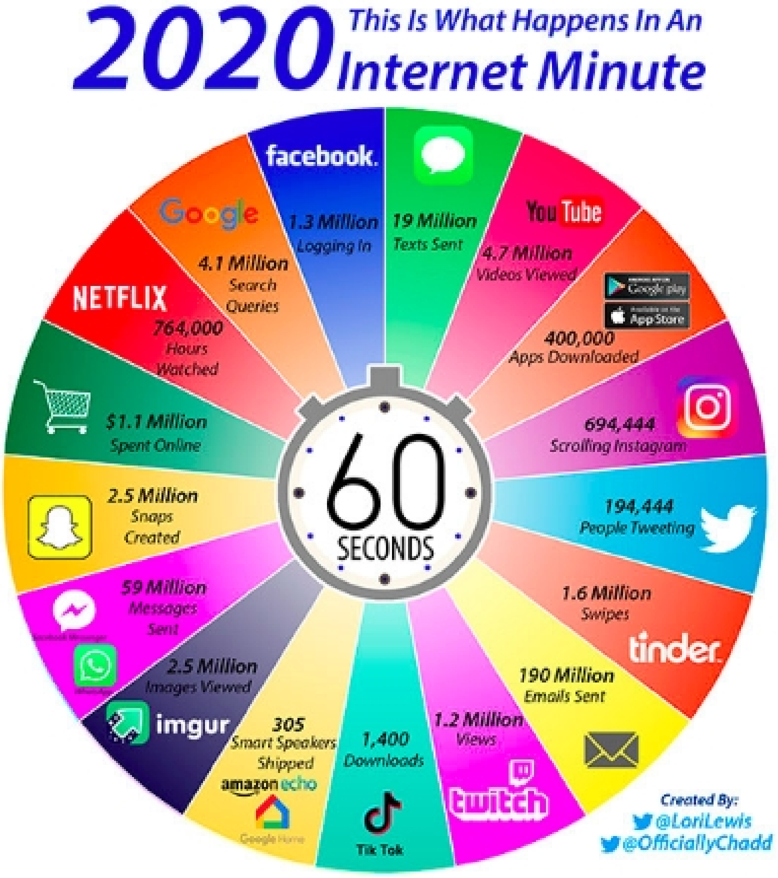


Figure 6.1: Statistics of UGC per minute[2]

To illustrate how fast the different forms of UGC spread on the Internet, we show a snapshot of the approximate amount of UGC generated per minute (published on March 10, 2020) in Figure 6.1. This single snapshot shows different activities of Internet users per minute. It is comprised of different kinds of UGC such as social networking, online shopping, dating, streaming etc. One of the items in this Figure shows that approximately $1.1M$ is spent per minute on online purchases. Furthermore, more than $194K$ people post their tweets on Twitter every single minute. These numbers reveal that a massive amount of UGC is generated in just one minute, so the amount of UGC generated in a week or in a month or even a year is enormous.

In general, UGC is informal in nature and in many cases conveys specific senti-

[2] https://www.allaccess.com/merge/archive/31294/infographic-what-happens-in-an-internet-minute

ment, aspects of which can vary to different degrees. For example, let us assume that a customer is not happy with the product they bought online and posts a feedback with negative sentiment. It is crucial for the vendor to keep track of such feedback as the sale of its products depends upon their customers' satisfaction/dissatisfaction. In another example, consider a person who is watching a football match on the Internet and feels happy to see their team winning the match, so may post a tweet with positive sentiment. In contrast, a supporter of the losing team may post a tweet with negative sentiment. Accordingly, the sentiment associated with tweets can vary from person to person.

Sentiment analysis of UGC is crucial in many real-life applications. For example, it is extremely important for companies like Amazon and eBay to understand the behaviour of their customers in order to keep track of their responses (either positive, negative or neutral). Multinationals like these have to facilitate communication from their customers using an array of different languages. Often, multilingual content might need to be translated in order that it is intelligible inside the company. In such cases, the sentiment of the customer feedback in one language should remain intact when translated into a different language, so that the feedback can be accessed in a crosslingual environment with the same sentiment.

This is called 'sentiment preservation', the ability of a system to preserve the sentiment class. Figure 6.2 illustrates three possible outcomes.

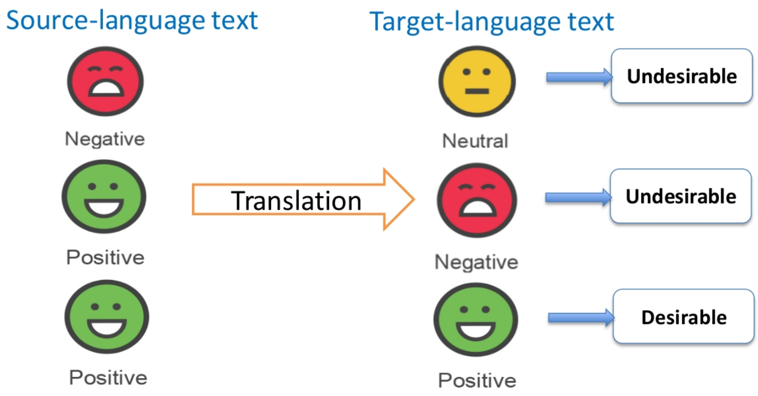


Figure 6.2: Sentiment after translation

The first one shows that the source-language text containing negative sentiment is translated into the target-language text that carries neutral sentiment, which is clearly undesirable. In the second example, the *positively sentimented* source-language text is translated into a *negatively sentimented* text in the target-language which is undesirable too. In contrast, the third example is the desirable outcome of an MT system where the source-language text retains its positive sentiment in the target language.

In this chapter, we propose methods to implement a 'sentiment preservation' MT system for UGC translation. Our major focus is on preserving sentiment during the translation process. Initially we apply sentiment classification on our experimental data set. Afterwards, we build a bunch of sentiment translation models each of which carry a specific sentiment. Finally, we perform translation using these sentiment translation models and evaluate the final output. The details on the sentiment translation system will be discussed in Section 6.4.

6.1 Related Work

A significant amount of work has been done in the area of UGC translation and sentiment analysis. The sentiment analysis research adopts mainly four approaches as follows: (i) translating sentiment resources, (ii) direct MT-based approach, (iii) cross lingual approach, and (iv) sentiment analysis on focus language.

- **Translating sentiment resources:** Hiroshi et al. (2004) use a transfer-based MT engine to translate the text documents into a set of sentiment units. In a similar vein, a graph-based approach using *SimRank* to transfer sentiment information from English to German is presented in Scheible et al. (2010).

- **Direct MT-based approach:** Many works in sentiment analysis is based on the direct application of MT. Mohammad et al. (2016) examine the sentiment of Arabic social media posts by (i) translating the focus language text into a resource-rich language such as English, and applying the English sentiment analysis system on the text, and (ii) translating resources such as sentiment-labeled corpora and sentiment lexicons from English into the focus language, and using them as additional resources in the focus-language sentiment-analysis system. They show that the sentiment analysis of English translations of Arabic texts produces competitive result, with respect to the Arabic sentiment analysis. Note that there system is also based on the first type of sentiment analysis approach we mentioned earlier, that is, 'translating sentiment resources'.

Balahur and Turchi (2012) handle the problem of sentiment detection in three different languages (French, German and Spanish) using three distinct MT systems: Bing,[3] Google,[4] and Moses (Koehn et al., 2007). These systems are used to translate the *training* data for a sentiment classification system so that the English sentiment analysis can be applied to the output. Araujo et al. (2016) show that simply translating the input text from a specific language to English and then using one of the existing methods for English can be better than the existing language-specific efforts evaluated.

- **Crosslingual approach:** A considerable amount of research in crosslingual sentiment analysis (CLSA) has been conducted as well. Lin et al. (2014) develop a model to implement aspect-specific sentiment analysis in a target language using the knowledge learned from a source language. The task of crosslingual sentiment lexicon learning by automatically generating target-language sentiment lexicons from available English sentiment lexicons is addressed in Gao et al. (2015). Jain and Batra (2015) use a recursive auto-encoder architecture to develop a CLSA tool using sentence-aligned corpora between a resource-rich (English) and a resource-poor (Hindi) language. He

[3] `https://www.bing.com/translator`

[4] `https://translate.google.com/`

et al. (2015) propose a semi-supervised learning approach with 'space transfer' to tackle the task of cross-language sentiment classification. It is also shown that the joint use of training data from multiple languages (especially those pertaining to the same family of languages) significantly improves the results of the sentiment classification (Balahur and Turchi, 2013).

- **Sentiment analysis on focus language:** Afli et al. (2017b) perform sentiment analysis of UGC for a low-resource language (Irish) by (i) using existing English sentiment analysis resources for tweets, and (ii) manually creating an Irish-language sentiment lexicon, *Senti-Foclóir*, that is used to build the first Irish sentiment analysis system called *SentiFocalTweet*. Note that their approach is not solely based on MT as one of their sentiment analysis systems that uses *SentiFocalTweet* does not require any translation.

Table 6.1 summarises the related works we discussed so far by dividing them into four different sentiment analysis approaches we mentioned above.

Approach	**Name of work**	**Languages involved**
Translating sentiment units	Hiroshi et al. (2004) Scheible et al. (2010) Mohammad et al. (2016)	Ja–En En–De Ar–En
MT-based approach	Balahur and Turchi (2012) Araujo et al. (2016)	Fr–En, De–En, Es–En Ar, Nl, Fr, De, It, Pt, Ru, Es, and Tr
Cross lingual approach	Lin et al. (2014) Gao et al. (2015) Jain and Batra (2015) He et al. (2015)	Zh, Fr, De, Es, It and Nl Zh and En En and Hi Zh and En
Sentiment analysis on focus language	Afli et al. (2017b)	Ga and En

Table 6.1: Approaches of sentiment analysis with examples

It is also shown that MT can alter the sentiment (Mohammad et al., 2016) during translation. They use Arabic social media posts and show that sentiment analysis of English translations of Arabic texts produces competitive results with respect to Arabic sentiment analysis. Arabic sentiment analysis systems benefit from the use of automatically translated English sentiment lexicons. They also conduct manual annotation studies to examine why the sentiment of a translation is different from the sentiment of the source word or text.

Although the sentiment analysis and translation of UGC is relatively well explored, to the best of our knowledge, the area of sentiment preservation in MT has never been investigated before. We mentioned earlier that preserving sentiment is crucial for many applications that involve UGC translation, which is the goal of our 3^{rd} research question (RQ-3). With this aim, we propose methods to build a suite of sentiment translation engines that focus on preserving the sentiment of the source-language UGC during the translation process.

6.2 Dataset

Our experiments consist of four different data sets, namely: (i) Twitter data, (ii) Flickr data, (iii) News Commentary data, and (iv) Arabic social media posts.

- **Twitter data:** The Twitter data set[5] (Sluyter-Gäthje et al., 2018) consists of $4,000$ English tweets from the FIFA World Cup 2014, plus their manual translations into German. We manually annotate sentiment scores between 0 and 1 to each tweet pair, where 0 represents extremely negative and 1 represents extremely positive sentiment. A value close to 0.5 means that the tweet conveys neutral sentiment. Out of the $4,000$ tweet pairs, we held out a small subset of 50 tweets per sentiment (negative, neutral and positive) for tuning and testing purposes because we wanted to maintain as large an amount as possible for the training purpose. The statistics of the number of parallel data used for training, tuning and testing for Twitter data is shown in Table 6.2. Although you might think that such a tiny parallel training data would produce low-quality MT, we will see in Section 6.4.1.4 that it is still possible to achieve interesting results.

[5] This data is available at: `https://github.com/HAfli/FooTweets_Corpus`

Sentiment	**#Training**	**#Development**	**#Test**
Negative	919	50	50
Neutral	$1,308$	50	50
Positive	$1,473$	50	50

Table 6.2: Statistics of the Twitter data set

Sentiment	**#Training**	**#Development**	**#Test**
Negative	$9,677$	50	50
Neutral	$11,065$	50	50
Positive	$8,258$	50	50

Table 6.3: Statistics of the Flickr data set

- **Flickr data:** As the Twitter data set was small, we decided to add larger corpora as out-of-domain data. The first of the out-of-domain data sets is the 'Flickr30k' data (Young et al., 2014). As we already described in Section 3.2.5 that the Flickr data consists of around $30K$ pictures from Flickr, one description in English and one human translation of the English description into German. We use only the textual part in our experiments. The data distribution is shown in Table 6.3. Note that the development and test sets shown are the same as the Twitter data set, which are the held out data from the Twitter corpus.

- **New commentary data set:** In order to accompany the tiny amount of Twitter data with an even much larger data set, we use the 'New commentary' data (in short 'news' data)[6] which is shown in Table 6.4.

Sentiment	**#Training**	**#Development**	**#Test**
Negative	$111,337$	50	50
Neutral	$14,306$	50	50
Positive	$113,200$	50	50

Table 6.4: Statistics of the News data set

The development and test data here too are the same as those for the Twitter data set.

[6] `http://data.statmt.org/wmt16/translation-task/training-parallel-nc-v11.tgz`

Sentiment	#Training	#Development	#Test
Negative	770	50	50
Positive	514	50	50

Table 6.5: Data statistics of Arabic Social Media Posts

- **Arabic Social Media Posts** This dataset contains $3,909$ *Levantine* Arabic social media posts or comments and their manual translations into English. The further details on this data set is already provided earlier in Section 3.2.5. The data distribution is shown in Table 6.5, where 100 negative and 100 positive posts are held out for the development and testing purposes (50 for development and 50 for test).

Note that the total number of negative and positive posts in Table 6.5 do not sum up to $3,909$ as we have not shown the number of neutral posts, which we do not use in this work.

6.3 Architecture of the Sentiment Translation System

To illustrate the special characteristics of the 'sentiment translation' system, let us show how it differs from the baseline translation engine. Figure 6.3 shows the architecture of the baseline system that is based on the translation of the source-language text by only a single translation engine, the 'Baseline Translation Model'.

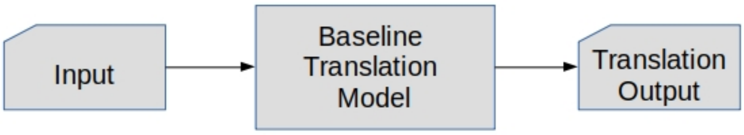


Figure 6.3: Baseline System Architecture

Now, we show the architecture of the sentiment translation system in Figure 6.4.

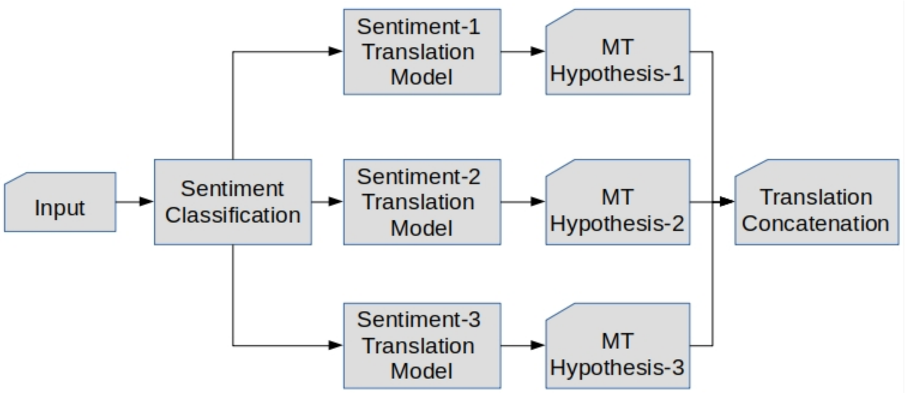

Input
Sentiment Classification
Sentiment-1 Translation Model
Sentiment-2 Translation Model
Sentiment-3 Translation Model
MT Hypothesis-1
MT Hypothesis-2
MT Hypothesis-3
Translation Concatenation

Figure 6.4: Sentiment Translation Architecture

The whole system is composed of the following phases: (i) sentiment classification, (ii) building sentiment translation models, and (iii) sentiment translation and concatenating translations. We will now briefly discuss each of the above phases in the following sections.

6.3.1 Sentiment Classification

This is the initial stage of building the sentiment preservation system. Firstly, we process the in-domain Twitter data (described in Section 6.2) manually and assign a sentiment score between 0 and 1 to each tweet where 0 means extremely negative and 1 means extremely positive sentiment. However, the out-of-domain data sets (Section 6.2) are much larger than the Twitter data so it was impractical to perform manual sentiment annotation on them. We use an automatic sentiment analysis tool (Afli et al., 2017b) that assigns a score between 0 and 1 to each text. We calculate the accuracy of this tool by applying it to the Twitter data set and compared the outputs with the gold standard annotations. It is found that this tool achieves a sentiment classification accuracy of 74.7%. We also calculate the Pearson correlation coefficient of the automatic and the manual sentiment annotations. In general, a value of +1 represents a total positive linear correlation, 0 refers to no linear correlation, and -1 implies a total negative linear correlation. We obtain the coefficient value of

0.603 which can be considered as moderately positive linear correlation. The whole parallel corpus including the in-domain and out-of-domain data for building MT systems (both training and test data) is divided into three different parts based on the sentiment classes (negative, neutral or positive). Finally, a certain sentiment class is assigned to the text pair based on the range of the sentiment score. The following ranges are used per sentiment class: (i) negative: when sentiment score < 0.4, (ii) neutral: when sentiment score $>= 0.4$ and $<= 0.6$, and (iii) positive: when sentiment score > 0.6

6.3.2 Building Sentiment Translation Models

Once the sentiment classes are assigned to the sentence pairs, the corpus is divided into the following three different parts according to the sentiment classes: (i) negative corpus, (ii) neutral corpus, and (iii) positive corpus. Each of these sub-corpora are used to build a particular sentiment translation model adjusted to a specific sentiment. For example, the *negative* corpus is used to build a translation model that conveys negative sentiment, and so is called the 'negative' translation model. Similarly, the neutral and positive corpora are used to build the neutral and positive translation models, respectively.

6.3.3 Sentiment translation and concatenating translations

When it comes to the translation phase, the test data is also divided into three parts according to the sentiment classes. Each part is then translated by the corresponding sentiment translation model. After the translation is complete, all the translated outputs are concatenated. Finally, the concatenated translation is evaluated in terms of both the translation quality as well as the sentiment preservation.

6.3.4 MT Systems Architecture

6.3.4.1 PBMT architecture

We use the Moses toolkit (Koehn et al., 2007) for building the PBMT models and Giza++ (Och and Ney, 2003) tool for the word and phrase alignments with a maximum phrase length of 7. For language modeling, we use the SRILM toolkit (Stolcke, 2002) to build trigram models. The models are tuned using minimum error rate training (Och, 2003).

6.3.4.2 NMT Architecture

The NMT models are built using 'OpenNMT'[7] (Klein et al., 2017) with sequence-to-sequence NMT models (Sutskever et al., 2014) based on recurrent neural networks with an attention mechanism (Luong et al., 2015). We use the default parameter settings: *RNN* as the default type of encoder and decoder, $word\_vec\_size = 500$, $rnn\_size = 500$, $rnn\_type = LSTM$, $global\_attention\_function = softmax$, $save\_checkpoint\_steps = 5000$, $training\_steps = 100,000$ etc.

6.3.4.3 Evaluation

Once the translation phase is complete, the outputs are evaluated both in terms of translation quality as well as in terms of sentiment preservation.

(a) **Translation quality evaluation:** We use widely used automatic MT evaluation metrices such as BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and TER (Snover et al., 2006).

(b) **Sentiment preservation evaluation:** Sentiment preservation is one of the main objectives of our research. It consists of calculating the percentage of translated outputs which carry the same sentiment as the source text. This percentage

[7] `https://github.com/OpenNMT/OpenNMT-py`

reflects to what extent our sentiment translation models are capable of preserving the same sentiment class after translation.

6.4 Experiments

6.4.1 Experiment 1: Maintaining Sentiment Polarity

In this experiment, we propose a strategy for building a suite of sentiment translation engines that attempt to preserve the sentiment in the source-language tweets in the target language during the translation process. We incorporate the sentiment classification within our MT engines to investigate to what extent the sentiment of tweets in the source language is preserved in the target language. We will now detail each phase of building the sentiment translation engines.

6.4.1.1 Sentiment Classification

We use the Twitter, Flickr and the News data (shown in Tables 6.2, 6.3 and 6.4) in this experiment. We already discussed the details on sentiment classification with for these data sets in Section 3.4.5.1. However, we show the statistics of both the in-domain (Twitter) and out-of-domain (Flickr and News) data sets again in Table 6.6 for convenience.

Data	**Training**			**Development**			**Test**		
	#neg	#neu	#pos	#neg	#neu	#pos	#neg	#neu	#pos
Twitter	919	1,308	1,473	50	50	50	50	50	50
Flickr	9,677	11,065	8,258	50	50	50	50	50	50
News_comm	111,337	14,306	113,200	50	50	50	50	50	50

Table 6.6: Data distribution after sentiment classification

6.4.1.2 Translation models

The translation models are built on three different sentiment classes (negative, neutral and positive); a 'negative sentiment' translation model is built from the 919 negative tweet pairs, a 'neutral sentiment' translation model is built from the 1,308 neutral tweet pairs and a 'positive sentiment' translation models is built from the 1,473 positive tweet pairs. In comparison, the baseline model (same as the baseline model discussed in Section 3.4.5) is built from the whole corpus regardless of sentiment classes. In addition, we add the the Flickr and News data successively to the Twitter data set to build larger sentiment translation models. For example, the negative text pairs from the Flickr and the News corpus are added to the negative tweet pairs of the Twitter corpus and a larger negative translation model is built from this concatenated data. In a similar manner, the larger neutral and positive models are build by adding the neutral and positive text pairs, respectively with the corresponding pairs of the Twitter data set.

6.4.1.3 Sentiment translation architecture

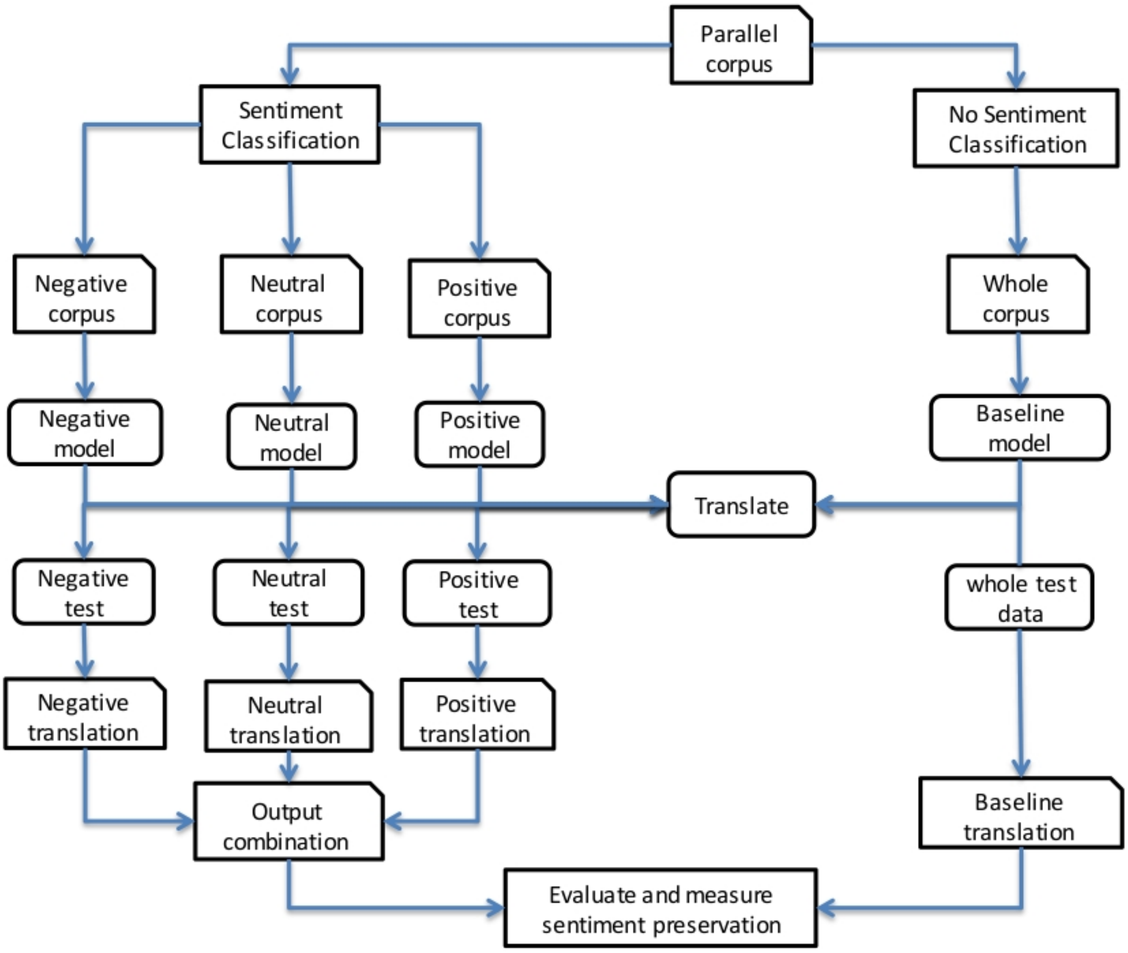


Figure 6.5: Architecture of the Sentiment Translation System

Figure 6.5 illustrates the architecture of the sentiment translation system. The whole system works in following steps:

(i) the sentiment classification approach divides the corpus in three different parts namely, negative, neutral and positive corpus and build corresponding sentiment translation models,

(ii) in contrast, a ‘baseline’ model is built using the whole corpus regardless of sentiment classes,

(iii) the test data is also divided into three parts; negative, neutral and positive classes and each part is translated using the corresponding sentiment translation models,

(iv) on the contrary, the whole test data is translated using the Baseline,

(v) the outputs generated by the sentiment translation models are combined and finally

(vi) the baseline translation and the combined output are evaluated and compared from the MT quality and sentiment preservation perspective.

6.4.1.4 Results

The evaluation is done in terms of both the translation quality and the sentiment polarity preservation. Table 6.7 summarises the results. Note that some of these results (rows 2, 4 and 6) were already shown in Section 3.4.5.2 on our experiments with sentiment preservation evaluation. The second column of this table shows whether the sentiment classification is used for each of the data combinations. Note that, BLEU and METEOR are precision based metrics, that is, the higher the score the better the system. In contrast, TER is an error-based metric, so the lower the score the better the system. We can see in this table that the best BLEU, METEOR and TER scores are obtained when only the Twitter data is used with no sentiment classification (referred to as the ‘Twitter Baseline’). However, the scores increase when the Flickr data is used as additional training data without applying the sentiment classification approach.

Translation model	Sent_Clas.	BLEU ↑	METEOR ↑	TER ↓	Sent_Pres.
Twitter	✓	48.2	59.4	34.2	72.66%
Twitter (Baseline)	×	50.3	60.9	31.9	66.66%
Twitter + Flickr	✓	48.5	59.8	33.9	71.33%
Twitter + Flickr	×	50.7	62.0	31.3	62.66%
Twitter + Flickr + News_Comm	✓	50.3	62.3	31.0	**75.33%**
Twitter + Flickr + News_Comm	×	**52.0** $*$	**63.4** $*$	**30.1** $*$	73.33%
Twitter (wrong MT engine)	✓	46.9	57.9	35.4	47.33%

Table 6.7: Experimental evaluation: With data concatenation

The further addition of News data produces the best BLEU, METEOR and TER scores of 52.0, 63.4 and 30.1, respectively. Apart from calculating BLEU, METEOR and TER, we also measure the statistical significance using MultEval (Clark et al., 2011). We highlight the systems that perform better than the Baseline with $p < 0.05$ using the $*$ sign. The most interesting results are obtained in terms of sentiment preservation which is the main objective of this work. When used with the sentiment classification approach ('Sent_Pres.'), the Twitter data produces a higher sentiment preservation score as compared to the Baseline (from 66.66% to 72.66%, which is 9% relative improvement).

The score further increases up to 75.33% (13% relative improvement over the Baseline and 3.6% over the Twitter sentiment classification model) upon addition of the Flickr and News data as the out-of-domain data set. These observations demonstrate that all the sentiment translation engines built from either only Twitter data or its combination with out-of-domain data set are more capable of preserving sentiment than the Baseline.

The last row of Table 6.7 shows the performance when the wrong MT engines are used, i.e. using MT engines with specific sentiments that are used to translate the texts with different sentiments. For example, consider a situation where we translate (i) negative tweets using the positive model, (ii) neutral tweets using the negative model, and (iii) positive tweets using the neutral model. Such a strategy can drastically change the system performance. It produces the worst scores in terms of both the translation quality (46.9 BLEU score) and sentiment preservation (47.33%).

Ex.	Reference	sentiment translation models	Baseline model
1	*Bosnia and Herzegovina really got f*** over man*	*Bosnia and Herzegowina eliminated echt demolished*	*Bosnia and Herzegovina were a abgezogen*
2	*when USA lost , but were still moving on to the next round*	*even if USA today we in the next round*	*could usa loses the next round*
3	*Brazil 5 WorldCup championship Argentina 2 WorldCup championship so Ill go with Brazil*	*Brazil 5 time world champion Argentina 2 time world champion so Im for Brazil*	*Brazil 5 time world champions Argentina 2 time world champions so for Brazil*

Table 6.8: Comparison of translations by sentiment translation models and Baseline model

Now we will show how the sentiment translation systems are capable of retaining the sentiment more than the Baseline with some real examples. Table 6.8 shows some of the most interesting outputs. In example 1 we can see that although the output generated by the sentiment translation models is not properly matched, its negative sentiment is still retained. In contrast, the Baseline fails to preserve its sentiment in the translation. This holds true for the 2^{nd} and the 3^{rd} examples as well. In contrast, it is difficult to infer the sentiments from the translations produced by the Baseline.

Ex.	Reference	Right MT engine	Wrong MT engine
1	*little break on the #WorldCup for an an amazing #Wimbledon final!*	*small Pause from the #WorldCup for a amazing #Wimbledon final!*	*kleine Pause of the #WorldCup for a erstaunliches #Wimbledon final!*
2	*yes !!!!!*	*yes !!!!!*	*so !!!!!*
3	*a bit boring ...*	*a little boring ...*	*some was ...*

Table 6.9: Comparison between the right and wrong MT engine

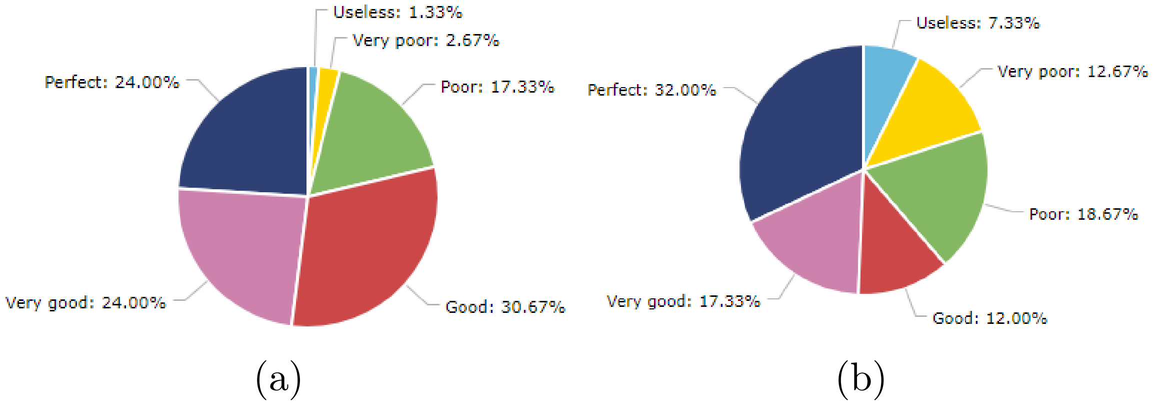
Table 6.9 shows the performance comparison between the *right* and the *wrong* MT engines. It is quite obvious that using the right MT engines always exhibits much better performance than the *wrong* ones. More precisely, it is worth translating a tweet with a specific sentiment using the corresponding sentiment translation models in terms of both translation quality and sentiment preservation.

6.4.1.5 Analysis

We analyse the outputs to find the reason why the sentiment translation system performs better than the baseline in terms of sentiment preservation. Consider the German word ‘abgezogen’ meaning ‘deducted’ in English which can carry different sentiment depending upon the context. For this reason, its translation produced by the baseline model may be unpredictable in terms of sentiment polarity. We notice that when we divide the whole corpus into different sentiment categories, most of the texts that contain this word fall into the negative corpus and so the negative model is most likely to translate this word in the negative context. Consider another example of a phrase ‘kommen wir’ which means ‘we come’ in English. Although this phrase does not carry any sentiment on its own, it can be a part of specific sentiment class depending upon the context. When we divide the corpus we found that many occurrences of ‘kommen wir’ involve positive sentiment whereas it occurs in a wide range of context in the whole corpus. Therefore, in the baseline model, there is a chance of seeing a mistranslation of this phrase in terms of sentiment polarity.

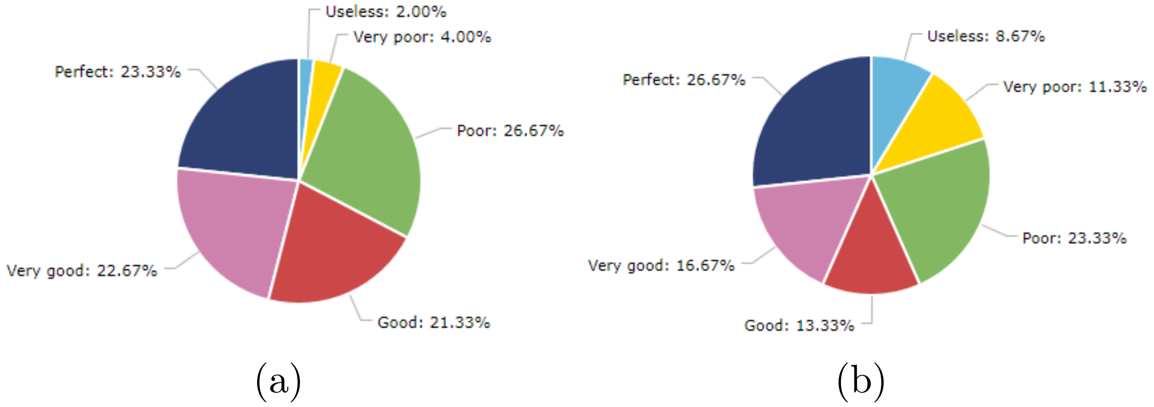
We also perform manual evaluation of the outputs produced by the baseline and sentiment translation models. The evaluation is done in terms of adequacy and fluency of the translation outputs. We divide the outputs into six categories both in terms of adequacy and fluency: (i) Useless: the translation output is completely wrong, (ii) Very poor: the output is of very low quality, (iii) Poor: most words are untranslated or have very low fluency, (iv) Good: when the majority of words are translated or have moderate fluency, (v) Very good: when most words are translated or have high fluency, (vi) Perfect: when all words are translated with excellent fluency.

The evaluation is performed on 150 translation outputs using the above six quality levels. Figure 6.6 and Figure 6.7 show the distribution of different levels of adequacy and fluency of the baseline and sentiment translation outputs, respectively. We can observe that the major part falls under either ‘Good’, ‘Very good’ and ‘Perfect’ categories for baseline outputs from an adequacy perspective (Figure 6.6a). Although a similar observation is made from fluency perspective, the percentage of lower quality translations (‘Poor’, ‘Very poor’ and ‘Useless’) is higher than for the adequacy feature.



(a) (b)

Figure 6.6: Adequacy and fluency of baseline system output



(a) (b)

Figure 6.7: Adequacy and fluency of sentiment translation output

Similar behaviour is seen for the sentiment translation outputs. Let us now compare the baseline and the sentiment translation outputs in terms of adequacy and fluency. In order to make a simple comparison, let us group the ‘Good’, ‘Very good’ and ‘Perfect’ into a single class and refer to it as ‘high quality’ translation and let us consider the combination of ‘Poor’, ‘Very poor’ and ‘Useless’ as ‘low quality' translations. Now, we show the percentage of high quality and low quality translations in terms of adequacy in Table 6.10 for both the baseline and sentiment translation (SentTrans) system.

Translation system	**% of High quality** (Good + Very good + Perfect)	**% of Low quality** (Poor + Very poor + Useless)
Baseline	**78.67**%	21.33%
SentTrans	67.33%	32.67%

Table 6.10: Comparison in terms of adequacy

Translation system	**% of High quality** (Good + Very good + Perfect)	**% of Low quality** (Poor + Very poor + Useless)
Baseline	**61.33**%	38.67%
SentTrans	56.67%	43.33%

Table 6.11: Comparison in terms of fluency

A similar comparison in terms of fluency is shown in Table 6.11. Both Table 6.10 and Table 6.11 show that the percentage of high quality translations both in terms of adequacy and fluency is higher for the baseline system as compared to the sentiment translation system. Considering this observation, even though the baseline outputs are better than the sentiment-translation outputs both in terms of adequacy and fluency, why does the sentiment translation system still obtain a higher sentiment preservation score? The obvious reason is that, although the sentiment translation system produces worse outputs than the baseline system, it manages to produce better translations for the words/phrases that convey specific sentiments. We already mentioned earlier in this section that a sub-corpus with specific sentiment contains words belonging to that specific sentiment and so is able to produce translations with the same sentiment.

6.4.1.6 Further experiments using Transformer model

We perform additional experiments on the Twitter data set using the transformer-based NMT system. To prepare the words for NMT training, all words are segmented into sub-word units using byte pair encoding (BPE) (Sennrich et al., 2016b). The vocabulary size for German is $3.5K$ and that for English is $3.2K$. In our experiments, we employ the base transformer configuration (Vaswani et al., 2017) and the Adam optimiser (Loshchilov and Hutter, 2019).

We compare the new results with the results achieved with our OpenNMT configuration (discussed in Section 6.3.4.2) using the same Twitter corpus.

NMT configuration	**BLEU ↑**	**METEOR ↑**	**TER ↓**	**Sent_Pres.**
OpenNMT	8.9	12.0	86.0	5%
Transformer-based	**18.2**	**20.9**	**71.9**	**17%**

Table 6.12: Experimental evaluation: Transformer based NMT

In addition, we manually evaluate the translation outputs in terms of sentiment preservation. The results are shown in Table 6.12. Note that these results are not comparable with those obtained using PBMT system (see Table 6.7) because $3.7K$ parallel segments are generally too small to be used for training an NMT model. Despite this assumption, it is worth investigating the effects of using the sub-word units for NMT training in a low-resource scenario.

We notice in Table 6.12 that the transformer-based model outperforms the OpenNMT system in all respects. The highest jump is seen in the BLEU score, from 8.9 to 18.2, which is more than double (104% relative improvement). Huge improvements are also seen in METEOR and TER scores. We also perform statistical significance testing using MultEval (Clark et al., 2011). It is found that all these improvements are statistically significant. Moreover, our manual evaluation also shows that the transformer-based model achieves a big improvement in sentiment preservation, from 5% to 17%, a 240% relative improvement. The reason why the transformer-based NMT system performs better than the OpenNMT framework is obvious. Our Twitter corpus is very small (only $4K$ text pairs) and in our configuration of OpenNMT, we use simple tokenisation and the sub-word units are not considered. In contrast, we segment the words into sub-word units with BPE in our transformer-based NMT system, which increases the chance of a word being translated.

6.4.2 Experiment 2: Translation Quality vs Sentiment Preservation

In our previous experiment (discussed in Section 6.4.1) we built a suite of sentiment-specific translation engines and pushed tweets containing either positive, neutral or negative sentiment through the appropriate engine to improve sentiment preservation in the target language. Although we achieved better sentiment preservation accuracy than the Baseline, we witnessed a small deterioration in translation quality. However, for certain use cases, preserving sentiment is far more important than the absolute quality achievable by the MT system.

In this experiment, we expand our sentiment translation models by including the nearest neighbour sentiment corpus. We focus on maintaining the level of sentiment preservation while trying to improve translation quality still further. More precisely, we try to retain the degree of sentiment while at the same time minimizing any loss in translation quality.

The extended sentiment translation system is composed of the following methods: (i) combining the negative and neutral corpus to build a translation system belonging to both negative and neutral sentiments, (ii) considering the neutral corpus separately for building the neutral translation model, and (iii) combining the positive and neutral tweet pairs to build a translation model conveying both of these sentiments. We combine the neutral sentiment class with the negative and the positive classes because the neutral class is relatively closer to both of these, compared to the distance between the negative and positive classes. This process helps increase the size of the negative and the positive sentiment translation models a bit further.

6.4.2.1 Sentiment Translation

In our previous experiment, we held out a small subset of only 50 tweet pairs per sentiment class (negative, neutral and positive totaling 150 pairs) for testing purposes. It is difficult to judge the system's performance with only 150 test pairs. In order to avoid this situation, we consider two different data distributions and use each of them in two different experimental set-ups (*Exp1* and *Exp2*), one with the same as the previous one and the other with a slightly larger amount of data. We assume that increasing the data size will make the analysis clearer and more informative than merely using only 150 test pairs.

Exp. setup	**Training**	**Development**			**Test**		
		#neg.	#neu.	#pos.	#neg.	#neu.	#pos.
Exp1	3,700	50	50	50	50	50	50
Exp2	3,400	100	100	100	100	100	100

Table 6.13: Data statistics

Table 6.13 show the data statistics of two different experimental setups. We again build the translation models exactly as before (discussed in Section 6.4.1.2).

The architecture of the sentiment translation system using the nearest neighbour sentiment class approach is illustrated in Figure 6.8.

The whole system works in the following steps:

(i) the neutral corpus, being the nearest neighbour is grouped with the negative and positive corpus separately to build the 'negative_neutral' and 'positive_neutral' models, respectively and referred to as the nearest sentiment translation models,

(ii) the 'neutral_model' is built from the neutral corpus,

(iii) the test data is divided into three parts; namely negative, neutral and positive test sets,

(iv) the translation process happens in the following order; the negative, neutral and the positive test data are translated by the negative_neutral, neutral and the positive_neutral translation models, respectively,

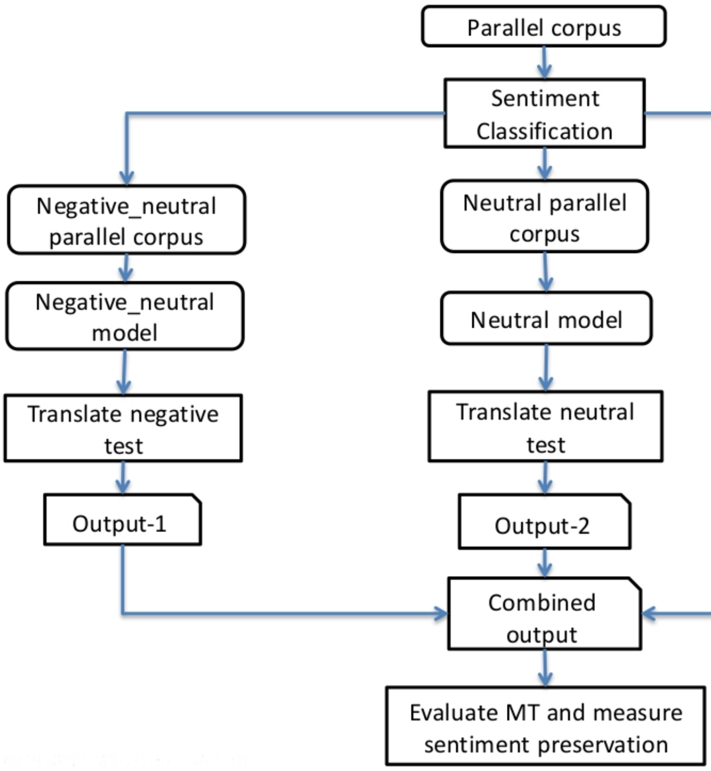


Figure 6.8: Sentiment translation using nearest neighbour sentiment classes

(v) the three outputs are combined and is then evaluated in terms of translation quality and sentiment preservation.

6.4.2.2 Results

We summarise the results of experiment-1 and experiment-2 in Table 6.14. The results in experiment-1 show that the gap between the BLEU score produced by the Baseline and sentiment translation model (50.3 and 48.2) is reduced by our proposed nearest sentiment translation model (TW_NS), as the BLEU score increases from 48.2 to 49.0. However, it cannot improve the sentiment preservation (SP) which remains the same as the Baseline. These results imply that the test data of experiment-1 (only 150 tweet pairs) is not sufficient to demonstrate the usefulness of our approach. In contrast, experiment-2 consisting of a comparatively larger data set (300 tweet pairs) produces better results. Clearly, experiment-1 and experiment-2 are not comparable to each other in terms of scores because the data distributions are different.

System	**Experiment** 1				**Experiment** 2			
	BLEU.	METEOR	TER	SP	BLEU.	METEOR	TER	SP
TW_Base	50.3	60.9	31.9	66.66%	51.3	62.5	31.0	52.33%
TW_SC	48.2	59.4	34.2	72.66%	47.3	59.1	35.2	60.33%
TW_NS	49.0	60.1	34.0	66.66%	48.3	59.6	34.4	60.0%

Table 6.14: Results for Experiments 1 and 2

We can see that the sentiment translation system (TW_SC) obtains the highest sentiment-preservation score of 60.33%. However the BLEU score worsens to exactly 4 points less than the Baseline. METEOR and TER scores are also worse in this case, which implies that the translation quality is degraded in parallel with the improvement in sentiment preservation. However, with the help of our nearest neighbour method (TW_NS), better BLEU, METEOR and TER scores are obtained than the sentiment translation model (TW_SC). At the same time, the sentiment preservation is reduced from 60.33% to 60.0% (only by 0.33%) which is trivial. The most important thing is that the nearest neighbour translation models (TW_NS) in experiment-1 are incapable of increasing the sentiment preservation besides reducing the translation gap between itself and the baseline (TW_Base). In contrast, experiment-2 with larger test data helps accomplish this task, i.e. it reduces the translation gap with the Baseline and results in a trivial loss in sentiment preservation compared to the sentiment translation system, thereby providing a better balance between translation quality and sentiment preservation.

6.4.3 Experiment 3: Sentiment Preservation of Arabic UGC

In our previous two experiments (Section 6.4.1 and Section 6.4.2), we investigated sentiment preservation using the English–German parallel Twitter corpus. In this experiment, we further explore the effects of sentiment translation system for a different language pair and different data set. We conduct experiments on sentiment preservation of parallel Arabic–English corpus of social media posts.

6.4.3.1 Sentiment classification

The data is composed of $3,909$ real Levantine Arabic social media posts or comments and their manual translations into English. The manually sentiment scoring is a bit different from the Twitter data. The scores are assigned as follows: (i) score $= 0.1$ if the post conveys strongly negative sentiment, (ii) $score = 0.3$ if it is negative, (iii) $score = 0.5$ if it is neutral, (iv) $score = 0.7$ if it is positive, and (v) $score = 0.9$ if it is strongly positive.

In this experiment, we consider only the negative (score$<= 0.3$) and positive (score$>= 0.7$) classes as TWB were interested only in considering the posts that imply whether the users are satisfied with the service (service provided by the team *Khabrona*) or not.

6.4.3.2 Translation models

Firstly, the corpus is divided into following two parts: (i) negative pairs (conveying negative sentiment), and (ii) positive pairs (conveying positive sentiment). Secondly, two translations models (negative and positive) are built using the negative and positive text pairs. For example, a 'positive sentiment' translation model is built from the corpus of 514 positive Levantine Arabic posts and their manual translations into English (see Table 6.5). In comparison, the baseline model is built from the whole corpus. Afterwards, the test data is also divided into negative and positive sentiment classes and each part is then translated by the corresponding sentiment translation model. Finally, we conduct the evaluation on translation quality and sentiment preservation similar to our previous experiments.

6.4.3.3 Results

The results obtained for sentiment preservation and MT quality are very similar to our previous results on Twitter data. Table 6.15 summarises these results.

MT system	BLEU	METEOR	TER	Sent_Pres.
Google Translate	**18.1**	**22.1**	**71.7**	59.33%
MT_Baseline	11.6	18.2	84.0	67.33%
MT_Sent_Class	8.2	15.7	89.7	**70.0%**

Table 6.15: Results on Arabic UGC translation

Note that, the results obtained by the Google Translate and MT_Baseline were already shown in Section 3.4.5.2 on our experiments with sentiment preservation in UGC translation. We compare the outputs generated by our sentiment translation system with those by Google Translate in terms of both translation quality and sentiment preservation and obtain good results. Google Translate produces higher BLEU scores than our sentiment translation systems, which is obvious from the fact that Google's MT system clearly has been built on much larger datasets than ours. In contrast, our systems produce significantly higher sentiment preservation scores than Google Translate, from 59.33 to 70.0 which is almost an 18% relative improvement. Such performance is obvious from the fact that our approach makes use of specific sentiment translation systems (negative or positive) depending upon whether the post is negative or positive, whereas Google Translate does not consider this fact. Instead, it uses a single MT system to translate the posts regardless of their sentiment.

6.5 Conclusions

In our first experiment, we measured the translation quality and sentiment preservation for English–German tweet pairs. We employed the sentiment classification approach to divide the parallel corpus into three parts, namely (i) negative, (ii) neutral and (iii) positive corpus. Our evaluation showed that the sentiment classification approach significantly improves sentiment preservation despite having a small deterioration in translation quality. Furthermore, as expected we also found that it is useful to select the proper MT engine that belongs to the same sentiment class as that of the UGC in order to maintain the translation quality and retain the same sentiment during translation.

Our second experiment was based on (i) combining the negative and the neutral tweet pairs to build the *negative_neutral* model, (ii) leaving the neutral model as it was in our preliminary experiment with sentiment translation, and (iii) combining the positive and the neutral tweet pairs to build the *positive_neutral* model. We had two different experimental setups for this experiment. The first setup with only 150 test sentences managed to improve the BLEU, METEOR and TER scores but could not increase the sentiment preservation score. We considered this experiment to have insufficient data to reveal the real results. In order to obtain a clearer picture, we used the second setup with a larger test set of 300 tweet pairs and repeated the experiments. This time we achieved better BLEU, METEOR and TER scores than the Twitter sentiment translation model (TW_SC) system and in addition, the sentiment preservation score was reduced only very slightly (by 0.33%) compared to the sentiment translation systems.

Finally, in the third experiment, we applied the same approach to the Arabic social media posts and their manual translations into English. We achieved better sentiment preservation than even Google Translate. Although the BLEU, METEOR and TER score for Google Translate were much better than with our sentiment translation systems, it is worth noting that the sentiment preservation scores with our translation systems were significantly better.

We had already shown the results of our baseline systems in Section 3.4.5 in terms of sentiment preservation. In this chapter, we revisited those results and tried to improve the sentiment preservation scores. We observed a significant improvement in sentiment preservation using our sentiment translation models as compared to our baseline results.

In this chapter, we addressed our third and final research question that deals with 'sentiment preservation in MT'. It is important to understand the underlying sentiments in UGC and when it comes to analyse the sentiment in the translated version of UGC, it becomes very important to retain the same sentiment during the translation process. To the best of our knowledge, no similar research has been done in the area of sentiment preservation in MT. We developed the first ever sentiment preservation system and obtained interesting findings. The results are very useful because preserving sentiment is crucial in use cases such as tweets, social media posts, especially when sentiment analysis in a multilingual environment is required. We consider this to be an interesting finding of our research as it has the potential to encourage researchers in the field of sentiment analysis of translated UGC.

In the next and the final chapter of this thesis, we conclude our research and point out future possibilities to extend our work.

Chapter 7

Conclusions and Future work

In this thesis, we explored the area of MT of UGC from the following perspectives: (i) conducting general experiments on UGC translation which causes us to set some research goals, (ii) implementation of a sophisticated bilingual document alignment system, (iii) development of an efficient parallel data extraction system, and (iv) sentiment preservation in UGC translation. Although there are several types of UGC available in social media to deal with, we used three types for our research: online posts, tweets and reviews. We also performed statistical significance testing for all the MT-related experiments in this research. We found that most of the results we obtained are statistically significant.

The general experiments on UGC translation involved (i) automatic spelling error correction for Arabic UGC translation, (ii) translation of German tweets into English, (iii) translation of English movie reviews into Serbian, (iv) translation of French restaurant reviews into English, and (v) evaluation of sentiment preservation in UGC translation.

Our first research goal in this thesis was to design a document alignment system in order to find similar bilingual documents from comparable corpora. We initially used text similarity and NE matching for this task. Afterwards, we significantly improved the alignment system by applying more sophisticated approach of CLIR and word embedding-based similarity.

The second research goal was to implement an efficient parallel data extraction system in order to extract parallel or semantically similar bilingual sentences from a comparable corpus of UGC. We applied the CLIR component of *FaDA* and our proposed method of average word vectors and text-based similarity with the help of a bilingual dictionary. Initially we tested the effectiveness of our system on a comparable corpus from news domain and then tested it on the comparable corpus of hotel reviews.

The final part of this thesis concerned the design of an MT-based sentiment preservation system. We performed sentiment classification to classify the parallel corpora into different sentiment classes. These sub-corpora were used to build a suite of sentiment-specific translation models that enable the preservation of sentiment of the source-language texts during the translation process.

In the next subsections, we will revisit our research questions that were introduced in the first chapter and re-discuss them briefly.

7.1 Research questions

We formulated three research questions in Chapter 1 of this thesis. In this section, we explain in brief how we addressed each of them and summarise our findings.

- **RQ-1: Provided with a collection of bilingual comparable documents, can we implement a sophisticated document alignment system that extracts semantically similar document pairs?**

We addressed this research goal in Chapter 4 where we provided a detailed description of our two experiments. The first experiment involved text similarity and NE-based matching to find bilingual similar document pairs. We used the English–French comparable documents from the WMT-2016 test data set to conduct the experiment and obtained a recall value of 0.291. Such a low recall suggested that this naive approach was not sufficient enough to obtain good-quality document alignments. Accordingly, we implemented a more refined document alignment system using CLIR, word embedding-based and text-based similarities. This approach drammatically improved the alignment system and achieved a recall value of 0.66 which is around 127% relative improvement over our previous alignment system.

- **RQ-2: Given the effectiveness of our document alignment system, can we implement an efficient, automatic, good quality parallel data extraction system from a comparable corpus of UGC?**

This research goal is addressed in Chapter 5. Our primary objective was to extract parallel sentences from comparable corpora mainly for UGC. To accomplish this task, we first transformed our document alignment system into a sentence alignment system. This was done by splitting each document (if it contains multiple sentences) into multiple sentences, storing each sentence in a single document and then performing document alignment. More precisely, we used the document alignment system at sentence level. In addition, we applied our proposed approach of average word vectors and text similarity with the help of a bilingual dictionary. Although our main objective was to extract parallel sentences for UGC, we began our experiment on clean texts to investigate the usefulness of our system and then tested it on a comparable corpus of UGC.

The first experiment involved parallel sentence extraction from the French–English comparable documents of the *Euronews* corpus. The extracted sentence pairs were then added to the already available French–English parallel sentences of the *News* *commentary* corpus in order to build an *Extended* translation model from this concatenated data set. In contrast, a baseline translation model was built from the *News commentary* corpus only. We compared these two models and found that the extended model outperformed the baseline model by 0.4 BLEU points.

In our second experiment, we applied similar strategies but this time with a corpus of English and French restaurant reviews. We used the already available French–English parallel *FourSquare* corpus to train the baseline model. In contrast, we built the *Extended* model using the concatenation of the *FourSquare* parallel corpus and the '*parallel sentences*'[1] we extracted from the restaurant reviews. We achieved a BLEU score improvement of only 0.2 points over the baseline model.

We noticed that the improvement in BLEU score in the second experiment was lower than in the first experiment. A probable reason is that the *Euronews* corpus already contains some parallel texts, so extracting and adding our additional parallel data can only marginally improve the BLEU score. In contrast, the restaurant reviews do not contain any parallel data because the reviews were randomly generated by different users without any translation usage foreseen. However, there are a few texts that are partially semantically equivalent and including them as additional data to build the *Extended* model helped improve the BLEU score very slightly. It is, therefore, expected that if there exists a significant amount of parallel sentences in a comparable corpus of UGC, our system can improve the BLEU score to a greater extent.

[1] In our case, the sentences are semantically equivalent instead of being perfectly parallel as it is extremely unlikely for the reviews to be translations of each other.

- **RQ-3: Can we build an MT-based sentiment preservation system using sentiment classification in order to best preserve the sentiment of the source-language texts during translation?**

Our final research goal was based on the implementation of an MT-based sentiment preservation system which is one of the most recent applications of UGC translation. We conducted three different experiments to demonstrate the usefulness of our sentiment preservation system.

In our first experiment, we translated German tweets from a small corpus of *FIFA-2014* World cup tweets into English. Firstly, we divided the whole parallel corpus into three different parts namely; negative, neutral and positive classes. We then built a suite of *sentiment translation* systems using these sub-corpora of different sentiment classes. We also divided the whole test data into three sentiment classes and translated each of them using the corresponding sentiment translation model. Afterwards, the translation outputs were concatenated. As a point of comparison, we built a single baseline model using the whole corpus. We used the whole test data and translated it using the baseline model. These two sets of translation outputs, i.e. one using the sentiment translation models and another using the baseline model, are compared in terms of both translation quality and sentiment preservation. Although the MT quality deteriorated (by 2.1 BLEU points), the sentiment preservation score improved significantly (by 9% relative improvement) by our sentiment translation model over the baseline model. Moreover, the addition of sentiment-classified out-of-domain *Flickr* and *News commentary* data sets to these sub-corpora (where each of these sub-corpora conveys particular sentiment) further improved the sentiment preservation score (up to 13% relative improvement over the baseline) and reached the same BLEU score as the baseline model.

In the second experiment, we aimed to reduce the translation quality gap and at the same time obtain higher sentiment preservation score than the baseline. We conducted experiments on the same German–English tweets but we used a different approach for building sentiment translation models. We introduced the nearest neighbour sentiment class approach to create the sub-corpora. We combined the corpora of the two nearest sentiment classes together to build specific sentiment translation models. For example, the negative corpus was concatenated with the neutral corpus because the neutral sentiment class is closer to the negative class than the distance between the negative and the positive classes. For the same reason, the positive corpus was concatenated with the neutral corpus. The translation model built from the former one was referred to as the *negative_neutral* model and the later one was referred to as the *positive_neutral* model. In contrast, the *neutral* model was built from only the neutral corpus. Finally, we adopted similar strategies as in the first experiment.

The experimental evaluation revealed that our approach of including the nearest neighbour sentiment classes enables the sentiment translation system to reduce the translation quality gap (from 48.2 to 49.0 as compared to 50.0 obtained by the baseline) and at the same time maintain the sentiment preservation score almost the same; this was reduced by only 0.33%; from 60.33% to 60% but was still much better than 52.33% obtained by the baseline model.

Our third experiment was similar to the first experiment but this time we used Arabic social media posts. We obtained similar improvements in this case as well. In addition, we compared our outputs with Google Translate and obtained interesting results. Although the outputs generated by Google Translate achieved a higher BLEU score than our outputs, our system outperformed Google's MT system in terms of sentiment preservation with an almost 18% relative improvement. Such an observation is expected because Google's MT system uses a single translation model regardless of the sentiment classes. However, it is worth noting that our translation model is much smaller than Google's model because we used less than $4K$ parallel sentences (even less when sub-copora are created by sentiment classification) to train our models.

It is important to observe that for our intended use-case, sentiment preservation is more important than translation quality per se, so we were prepared to put up with a small drop in translation performance. We expect that in addition to achieving a higher sentiment preservation score than Google Translate, it is definitely possible to drammatically reduce the translation gap if we can collect a decent amount of parallel UGC data.

The above findings are very interesting because maintaining the sentiment classes is much more important for UGC translation in many cases, specially sentiment analysis in bilingual platform. It does not matter much even if the translation quality deteriorates slightly as long as the correct sentiment is transferred from the original text to the translated text. To the best of our knowledge, we implemented the first ever sentiment translation system that aims at preserving sentiment in UGC translation. We believe that our work in this field has the potential to attract researchers who wish to specialise in this area.

7.2 Future work

In this section we will discuss the future directions to extend our work. Although there are endless possibilities for expanding our research, we will mention some concrete examples in this section.

7.2.1 Further experiments on UGC-translation

- **Expanding Arabic UGC-translation system:** In Section 3.4.1 we discussed our experiments on integrating spelling error correction to Arabic UGC-translation. As future work, we plan to do the reverse, i.e. automatically create errors in the training data and then train NMT systems and investigate their robustness to typos and non-standard grammar. We also plan to build character-level systems. Furthermore, we will normalise the mixing of characters (e.g. in Latin, 'l' and 'O' are used instead of 1 and 0, respectively) in order to resolve the problem of non-standard choice of Unicode characters.

- **Emphasis on terminology translation:** In our experiments on translating English movie reviews into Serbian (Section 3.4.3), we applied a forward translation technique to create synthetic data for MT training. We also mentioned that forward translation is not as fruitful as back translation. We found issues with translating names in Slavic languages. Such problems can be addressed by some filtering of proper names. For example, it is possible to replace all names with some placeholder (e.g. XXX for persons, YYY for places, ZZZ for organisations etc.). This process allows the synthetic data to keep the sentence structure intact without the complication of having to illustrate the named entities correctly. Therefore, the translation systems will leave the terms untranslated by default and so the overall translation quality is not harmed. Another way of addressing this issue is to focus on terminology translation in low-resource conditions (Haque et al., 2019a,b, 2020).

- **Exploring NMT with less data:** In this thesis, one of our main goals was to address the problem of data scarcity for UGC. We attempted to find parallel sentences for UGC from a comparable corpus of hotel reviews. However, we found that such data is extremely difficult to find because the reviews are generated without any translation usage foreseen. We therefore plan to explore the utility of translating the subword units (Sennrich et al., 2016b) in such low-resource scenarios. In addition, we also leverage the capability of optimised NMT techniques even with small amounts of data (Sennrich and Zhang, 2019).

- **Utilising back-translation for UGC:** We mentioned in Section 1.3 that we would be interested in exploring bilingual comparable corpora instead of using back-translation to resolve the problem of parallel data scarcity for UGC. However, it is worth comparing the performance of our approach and back-translation because the latter exploits fully non-parallel sentences in a comparable corpus, relying on known parts of sentences.

In addition to the above future plans, we are also interested in assessing the quality of the n-gram models used in our translation model using the Twitter data (see Section 3.4.5). We already mentioned in Section 2.1.1 that we used trigram models for our experiments. However, for UGC corpora with spelling errors, the idea of n-gram matching may not work well. For this reason, it would be good to investigate its impact on translating UGC.

7.2.2 Improving document alignment

We already significantly improved the performance of our document alignment system using the word embedding and text similarity-based approach compared to our previous system that employed only text and NE matching. However, there are still plenty of opportunities to extend it considering the limitations of our document alignment system.

- **Application of bilingual word embedding:** We applied monolingual word embeddings in our document alignment system. However, there is a possibility to incorporate bilingual word embeddings in this work. One of the ways to do this is to intermix both the source and target side of a parallel corpus and then train the word vector model on the mixed data. For example, we can take the English–French parallel sentences from the *News commentary* corpus and mix the words of an English sentence with the words of its French counterpart. It is important that the words should be mixed in proper order rather than mixing them randomly. The following example of a proper word mixing will make this clear.

English: *But no country has the luxury of choosing its neighbors.*

French: *Mais aucun pays ne possède le luxe de choisir ses voisins.*

Mixed: *Mais but aucun no pays country ne has possède the le luxury luxe of de choosing choisir its ses neighbors voisins*

The above sentences are taken from the *News commentary* corpus. This sentence pair contains 10 words in English and 11 words in French. We start with the first French word *Mais* and then place the first English word *But* after this word and continue doing this until we reach the end of each sentence. However, there may be cases where the English–French sentence length ratio may be far less than 1. For example, if a French sentence contains 5 words and an English sentence contain 8 words, the ratio becomes 0.6. In this case we can place two English words near each French word and one French word will remain at the end. Afterwards, such intermixed word sequences are formed for all the English–French sentence pairs in the whole corpus and can be fed as input to the word vector training module. The module will treat these as monolingual texts and embed the words in vector space. As a result, co-occurring words (in this case adjacent English and French words) will contain similar vector values. It is, therefore, possible to obtain synonymous words in both English and French. The application of bilingual word embeddings can thus remove the requirement for a bilingual dictionary in our document alignment system as it is not required to translate the words of the French documents into English. We can directly obtain English alignments for each French document without performing any word translation.

Note that the above method of word mixing may not work well for the language pairs with differences in word order. For example, German puts the main verb at the end of the sentence, so it will be a long way away from the corresponding English verb. A possible solution to this problem is to firstly reorder the German sentence according to the word order of English and then intermix the words for word vector training. However, several recent works adopted other strategies for obtaining bilingual word embeddings. For example, Artetxe et al. (2017) propose to reduce the requirement of bilingual resources using a self-learning approach that can be combined with any dictionary-based mapping technique. A shared-private technique to improve the learning of bilingual word embeddings for NMT is proposed in Liu et al. (2019). Goikoetxea et al. (2018) use monolingual corpora and bilingual wordnets to produce a bilingual embedding space. Their approach is based on a random walk algorithm over bilingual wordnets to create bilingual corpus which is then combined with monolingual corpora that is fed into skipgram, generating bilingual embeddings.

• **Text matching with stemmed words:** At present, the text similarity method in our experiments is comprised of simple text matches. However, many words in the documents are in inflected forms which may not be found in the bilingual dictionary. As a result, some of the inflected words in French may be untranslated and thus cannot be matched with the words in the English sentence. To overcome this, we can stem the inflected words in both the English and French documents and then translate the stemmed words in French in order to match with the stemmed words in English. This process can increase the chance of matching similar but inflected words and so the accuracy of the document alignment system can be improved.

7.2.3 Refinement of parallel data extraction system

In this thesis, our parallel data extraction system targets comparable UGC corpora. However, it is extremely difficult to extract parallel UGC sentences as they hardly exist. In fact, such contents are randomly generated without being translations of each other. Nonetheless, we found that there do exist some partially semantically similar texts that can be treated as a parallel resource for MT as there is no sufficiently large parallel corpus available for UGC on the Internet. Our parallel data extraction system is at the beginning phase and there is plenty of room for improvement. The planned work includes but is not limited to the following.

- **Fine tuning:** At this moment, our parallel data extraction system is in its basic form. We plan to perform fine tuning of parameters. We will have to spend a huge amount of time on this, especially for a large comparable corpus. The reason for this is that it can take a significant amount of time to determine the contribution of each component as it is a lengthy process to run the whole experiment for a particular parameter setting.

- **Application of bilingual word embedding and stemmed word matching:** This approach resembles our plan for improving our document alignment system. We apply similar methods but this time at the sentence level, i.e. we apply the bilingual word embeddings as well as the stemmed word matching in order to compute sentence similarity.

- **Combination with other sentence aligners:** The task of parallel sentence extraction can also be viewed as sentence alignment in two languages. One of the drawbacks in our approach is that we have not compared our system with some of the most popular existing sentence alignment systems. Some examples of well-known works in this field are Sennrich and Volk (2010), Gomes and Lopes (2016) and Thompson and Koehn (2019). We plan to explore these approaches and apply their sentence alignment systems for UGC. Our main objective is to combine the best performing system with our system and conduct experiments with the data sets we used in this thesis. Another possibility is to apply all of them separately and select the sentence alignments that are common outputs generated by all or most of these alignment systems.

Apart from refining our parallel data extraction system, we also plan to apply it to other types of UGC such as tweets, customer feedback, movie reviews etc.

7.2.4 Enriching sentiment preservation system

Our third research goal was to implement an MT-based sentiment preservation system. We plan to extend this work by including some tasks that we have not done so far.

• **Impact of parallel data extraction on sentiment preservation:** Our second research question was to implement a parallel data extraction system for UGC. As of now, we have added them as additional data for MT training and found that the BLEU score increased very slightly. However, we have not tested its utility on sentiment preservation yet. Our plan is to investigate how the addition of such extracted parallel resources affect sentiment preservation besides the MT quality per se.

• **Human evaluation of translations:** Although the sentiment translation models in our first experiment increased the sentiment preservation score to some extent, the translation quality is degraded compared to the baseline translation model. We reduced this translation gap in the second experiment with a minimal loss in sentiment preservation. However, we have not performed any human evaluation of the translation outputs generated by our sentiment translation models. Taking this into account, we plan to manually evaluate the outputs in terms of (i) adequacy, and (ii) fluency. This process is very important because we need to determine how much information/readability of the source-language text is retained apart from maintaining sentiment polarity during the translation process. Increasing these values along with preserving the sentiment is definitely more acceptable as the outputs are then more informative as well as maintaining sentiment polarity.

- **Incorporating more fine-grained sentiment classes:** Our sentiment translation systems are built using three sentiment classes so far: negative, neutral and positive. However, there is a possibility to include other classes as well, such as the strong negative and the strong positive, resulting in 5 classes in total. We can then build 5 sentiment translation models and use all of them to translate the texts with corresponding sentiment classes.

- **Building larger sentiment translation models:** As of now, our sentiment translation models are very small because the training corpus contains only $4K$ tweet pairs which is further divided into sub-corpora with different sentiments. However, we increased the corpus size by including the *News commentary* corpus but it is not UGC. It is more sensible to add a larger corpus of UGC to the Twitter data set. We plan to include the *FourSquare* corpus which is still not large enough (only $18K$ sentence pairs) but still much larger than the $4K$ tweet pairs in the Twitter data set. In addition, we will also consider the IMDb review corpus with the negative and positive polarities in order to conduct similar experiments. We will divide these corpora using the same sentiment classification approach and add the sub-corpora to the Twitter data set depending upon their sentiment classes. The sentiment translation models will thus expand and hopefully will produce better translation outputs besides increasing the sentiment preservation score. We will also explore other UGC corpora available on the Internet such as customer feedback, hotel reviews etc. and continue to expand our sentiment translation models.

Bibliography

Afli, H., Aransa, W., Lohar, P., and Way, A. (2016a). From Arabic user-generated content to machine translation: integrating automatic error correction. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–14, Konya, Turkey.

Afli, H., Barrault, L., and Schwenk, H. (2015). Building and using multimodal comparable corpora for machine translation. *Natural Language Engineering*, 22:1–21.

Afli, H., Barrault, L., and Schwenk, H. (2016b). Building and using multimodal comparable corpora for machine translation. *Natural Language Engineering*, 22(4):603 – 625.

Afli, H., Lohar, P., and Way, A. (2017a). MultiNews: A web collection of an aligned multimodal and multilingual corpus. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*, pages 11–15, Taipei, Taiwan.

Afli, H., McGuire, S., and Way, A. (2017b). Sentiment translation for low resourced languages: Experiments on Irish General Election Tweets. In *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–10, Budapest, Hungary.

Alperen, M. S., Kapanadze, O., Ceausu, A., Ramisch, C., and Fotopoulou, A. (2010). South-East European Times : A parallel corpus of Balkan languages , Francis Tyers and. In *Proceedings of the 7th Conference on Language Resources and Evaluation*, pages 1–6, Valleta, Malta.

Araujo, M., Reis, J., Pereira, A., and Benevenuto, F. (2016). An evaluation of machine translation for multilingual sentence-level sentiment analysis. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1140–1145, New York, USA.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15, San Diego, CA, USA.

Balahur, A. and Turchi, M. (2012). Multilingual sentiment analysis using machine translation? In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 52–60, Jeju, Republic of Korea.

Balahur, A. and Turchi, M. (2013). Improving sentiment analysis in twitter using multilingual machine translated data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 49–55, Hissar, Bulgaria.

Balikas, G., Laclau, C., Redko, I., and Amini, M. (2018). Cross-lingual document retrieval using regularized wasserstein distance. In *Proceedings of the 40th European Conference on Information Retrieval Research*, pages 398–410, Grenoble, France.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2011). Domain adaptation in statistical machine translation of user-forum data using component level mixture modelling. In *Proceedings of Machine Translation Summit XIII*, pages 285–292, Xiamen, China.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2012). Domain adaptation in SMT of user-generated forum content guided by OOV word reduction: Normalization and/or supplementary data? In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 169–176, Trento, Italy.

Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M. L., Kamran, A., Kirefu, F., Koehn, P., Ortiz Rojas, S., Pla Sempere, L., Ramírez-Sánchez, G., Sarrías, E., Strelec, M., Thompson, B., Waites, W., Wiggins, D., and Zaragoza, J. (2020). ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2016). Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas.

Berard, A., Calapodescu, I., Dymetman, M., Roux, C., Meunier, J.-L., and Nikoulina, V. (2019). Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong.

Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece.

Bouamor, H. and Sajjad, H. (2018). H2@BUCC18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1–5, Miyazaki, Japan.

Braune, F. and Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 81–89, Beijing, China.

Brown, P., Della Pietra, S., Pietra, V., and Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Buck, C. and Koehn, P. (2016). Quick and reliable document alignment via TF/IDF-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678, Berlin, Germany.

Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. Technical report, Linguistic Data Consortium, University of Pennsylvania, Pennsylvania, USA.

Burlot, F. and Yvon, F. (2018). Using Monolingual Data in Neural Machine Translation: a Systematic Study. In *Proceedings of the 3rd Conference on Machine Translation (WMT 2018)*, pages 144–155, Belgium, Brussels.

Calixto, I., Stein, D., Matusov, E., Lohar, P., Castilho, S., and Way, A. (2017). Using images to improve machine-translating e-commerce product listings. In *Proceedings of the 15th Conference of the European Chapter of the Association* *for Computational Linguistics: Volume 2, Short Papers*, pages 637–643, Valencia, Spain.

Carter, S., Tsagkias, M., and Weerkamp, W. (2011). Twitter hashtags: Joint translation and clustering. *Text - Interdisciplinary Journal for the Study of Discourse*, pages 1–4.

Castilho, S., Moorkens, J., Gaspari, F., Sennrich, R., Sosoni, V., Georgakopoulou, Y., Lohar, P., Way, A., Valerio, A., Miceli Barone, A. V., and Gialama, M. (2017). A comparative quality evaluation of PBSMT and NMT using professional translators. In *Proceedings of the 16th Machine Translation Summit*, pages 116–131, Nagoya, Japan.

Chen, A. (2003). Cross-language retrieval experiments at CLEF 2002. *Advances in Cross-Language Information Retrieval*, pages 28–48.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar.

Chu, C. (2015). *Integrated Parallel Data Extraction from Comparable Corpora for Statistical Machine Translation*. PhD dissertation, Kyoto University.

Chu, C., Nakazawa, T., and Kurohashi, S. (2015). Integrated parallel sentence and fragment extraction from comparable corpora: A case study on Chinese–Japanese Wikipedia. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15(2).

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies:*, pages 176–181, Portland, Oregon, USA.

Dara, A. A. and Lin, Y.-C. (2016a). YODA system for WMT16 shared task: Bilingual document alignment. In *Proceedings of the First conference of Machine Translation*, pages 679–684, Berlin, Germany.

Dara, A. A. and Lin, Y.-C. (2016b). YODA system for WMT16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 679–684, Berlin, Germany.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, USA.

Di Gangi, P. and Wasko, M. (2009). The co-creation of value: Exploring user engagement in user-generated content websites. In *Proceedings of Pre-ICIS 8th Annual JAIS Theory Development Workshop*, Phoenix, Arizona, USA.

El-Kishky, A., Chaudhary, V., Guzmán, F., and Koehn, P. (2019). A massive collection of cross-lingual web-document pairs. *ArXiv*, abs/1911.06154.

El-Kishky, A. and Guzmán, F. (2020). Massively multilingual document alignment with cross-lingual sentence-mover's distance. *ArXiv*, abs/2002.00761.

Esplà-Gomis, M. (2009). Bitextor: a Free/Open-source Software to Harvest Translation Memories from Multilingual Websites. In *The twelfth Machine Translation Summit*, pages 1–8, Ottawa, Canada.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, page 363–370, Ann Arbor, Michigan.

Foster, G. and Kuhn, R. (2007). Mixture-model Adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic.

Gao, D., Wei, F., Li, W., Liu, X., and Zhou, M. (2015). Cross-lingual sentiment lexicon learning with bilingual word graph label propagation. *Computational Linguistics*, 41(1):21–40.

Goikoetxea, J., Soroa, A., and Agirre, E. (2018). Bilingual embeddings with random walks over multilingual wordnets. *Knowledge-Based Systems*, 150:218–230.

Gomes, L. and Lopes, G. P. (2016). First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2228–2231, Portorož, Slovenia.

Gomes, L. and Pereira Lopes, G. (2016). First steps towards coverage-based document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 697–702, Berlin, Germany.

Grégoire, F. and Langlais, P. (2018). Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the*

27th International Conference on Computational Linguistics, pages 1442–1453, Santa Fe, New Mexico, USA.

Guo, M., Yang, Y., Stevens, K., Cer, D., Ge, H., Sung, Y.-h., Strope, B., and Kurzweil, R. (2019). Hierarchical document encoder for parallel corpus mining. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 64–72, Florence, Italy.

Gupta, D., Raveendran, V., and Yadav, R. (2014). Domain biased bilingual parallel data extraction and its sentence level alignment for english-hindi pair. *Research Journal of Applied Sciences, Engineering and Technology*, 7:1001–1012.

Habash, N. and Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, page 573–580, Ann Arbor, Michigan, USA.

Hangya, V. and Fraser, A. (2019). Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1224–1234, Florence, Italy.

Haque, R., Hasanuzzaman, M., and Way, A. (2019a). Investigating terminology translation in statistical and neural machine translation: A case study on english-to-hindi and hindi-to-english. In *Proceedings of RANLP 2019: Recent Advances in Natural Language Processing*, pages 437–446, Varna, Bulgaria.

Haque, R., Hasanuzzaman, M., and Way, A. (2019b). Terminology translation in low-resource scenarios. *Information*, 10(9):273–300.

Haque, R., Hasanuzzaman, M., and Way, A. (2020). Analysing terminology translation errors in statistical and neural machine translation. *Machine Translation*, 35.

He, R., Lee, W., Ng, H., and Dahlmeier, D. (2019). An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy.

He, X., Zhang, H., Chao, W., and Wang, D. (2015). Semi-supervised learning on cross-lingual sentiment analysis with space transfer. In *Proceedings of the IEEE First International Conference on Big Data Computing Service and Applications*, pages 371–377, Washington, DC, USA.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Hiroshi, K., Tetsuya, N., and Hideo, W. (2004). Deeper sentiment analysis using machine translation technology. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 494–500, Geneva, Switzerland.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA, USA.

Jain, S. and Batra, S. (2015). Cross lingual sentiment analysis using modified brae. In *Proceedings of the 20th International Conference on Empirical Methods in Natural Language Processing*, pages 159–168, Lisbon, Portugal.

Jehl, L., Hieber, F., and Riezler, S. (2012). Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 410–421, Montreal, Canada.

Jiang, J., Way, A., and Haque, R. (2012). Translating user-generated content in the social networking space. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*, pages 1–9, San Diego, USA.

Junczys-Dowmunt, M., Heafield, K., Hoang, H., Grundkiewicz, R., and Aue, A. (2018). Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia.

Karimi, A., Ansari, E., and Sadeghi Bigham, B. (2018). Extracting an English-Persian parallel corpus from comparable corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3477–3482, Miyazaki, Japan.

Khalil, S. and Fakir, M. (2017). RCrawler: An R package for parallel web crawling and scraping. *SoftwareX*, 6:98 – 106.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1367–1373, Geneva, Switzerland.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th annual meeting of the Association for Computational Linguistics*, pages 67–72, Vancouver, Canada.

Klempová, H., Novák, M., Fabian, P., Ehrenberger, J., and Bojar, O. (2009). Získávání paralelních textú z webu. In *Conference on Theory and Practice on Information Technologies*, page 1==8, Dolny Kubin, Slovakia.

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

Koehn, P. (2009). *Phrase-Based Models*, page 127–154. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

Koehn, P. and Knowles, R. (2017). Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, Edmonton, Canada.

Kúdela, J., Holubová, I., and Bojar, O. (2017). Extracting Parallel Paragraphs from Common Crawl. *Prague Bulletin of Mathematical Linguistics*, 107:39–56.

Le, T., Vu, H. T., Oberländer, J., and Bojar, O. (2016). Using term position similarity and language modeling for bilingual document alignment. In *Proceedings of the first conference on Machine Translation*, pages 710–716, Berlin, Germany.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Li, B. and Gaussier, E. (2010). Improving Corpus Comparability for Bilingual Lexicon Extraction from Comparable Corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 644–652, Beijing, China.

Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 605–612, Barcelona, Spain.

Lin, Z., Jin, X., Xu, X., Wang, W., Cheng, X., and Wang, Y. (2014). A cross-lingual joint aspect/sentiment model for sentiment analysis. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1089–1098, Shanghai, China.

Ling, W., Marujo, L., Dyer, C., Black, A. W., and Trancoso, I. (2014). Crowdsourcing High-Quality Parallel Data Extraction from Twitter. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 426–436, Baltimore, Maryland, USA.

Ling, W., Xiang, G., Dyer, C., Black, A., and Trancoso, I. (2013). Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 176–186, Sofia, Bulgaria.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Morgan & Claypool Publishers.

Liu, X., Wong, D., Liu, Y., Chao, L., Xiao, T., and Zhu, J. (2019). Shared-Private Bilingual Word Embeddings for Neural Machine Translation. In *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics*, pages 3613–3622, Florence, Italy.

Lohar, P., Afli, H., Liu, C.-H., and Way, A. (2016a). The ADAPT bilingual document alignment system at WMT16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 717–723, Berlin, Germany.

Lohar, P., Afli, H., and Way, A. (2017a). Maintaining sentiment polarity in translation of user-generated content. *The Prague Bulletin of Mathematical Linguistics*, 108(1):73 – 84.

Lohar, P., Afli, H., and Way, A. (2017b). Maintaining sentiment polarity in translation of user-generated content. *The Prague Bulletin of Mathematical Linguistics*, 108(1).

Lohar, P., Afli, H., and Way, A. (2018a). Balancing translation quality and sentiment preservation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pages 81–88, Boston, MA, USA.

Lohar, P., Afli, H., and Way, A. (2018b). Balancing translation quality and sentiment preservation (non-archival extended abstract). In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 81–88, Boston, MA, USA.

Lohar, P., Dutta Chowdhury, K., Afli, H., Hasanuzzaman, M., and Way, A. (2017c). ADAPT at IJCNLP-2017 task 4: A multinomial Naive Bayes Classification approach for customer feedback analysis task. In *Proceedings of The 8th International Joint Conference on Natural Language Processing, Shared Tasks*, pages 161–169, Taipei, Taiwan.

Lohar, P., Ganguly, D., Afli, H., Way, A., and Jones, G. J. (2016b). FaDA: Fast Document Aligner using Word Embedding. *The Prague Bulletin of Mathematical Linguistics*, 106(1):169–179.

Lohar, P., Popovic, M., Afli, H., and Way, A. (2019a). A systematic comparison between SMT and NMT on translating user-generated content. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–11, La Rochelle, France.

Lohar, P., Popović, M., and Way, A. (2019b). Building English-to-Serbian machine translation system for IMDb movie reviews. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 105–113, Florence, Italy.

Lohar, P. and Way, A. (2020). Parallel data extraction using word embeddings. In *International Conference on NLP Techniques and Applications*, London, United Kingdom (Accepted).

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*, pages 1–18, New Orleans, LA, USA.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA.

Medved, M., Jakubícek, M., and Kovár, V. (2016). English-French Document Alignment Based on Keywords and Statistical Translation. In *Proceedings of the First Conference on Machine Translation*, pages 728–732, Berlin, Germany.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the international conference on Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, USA.

Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55(1):95–130.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, USA.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, Spain.

Papavassiliou, V., Prokopidis, P., and Piperidis, S. (2016). The ILSP/ARC submission to the WMT 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 733–739, Berlin, Germany.

Papavassiliou, V., Prokopidis, P., and Thurmair, G. (2013). A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 43–51, Sofia, Bulgaria.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Park, J., Song, J., and Yoon, S. (2017). Building a Neural Machine Translation System Using Only Synthetic Parallel Data. *Computing Research Repository.*

Poncelas, A., Lohar, P., Way, A., and Hadley, J. (2020). The impact of indirect machine translation on sentiment classification. In *The 14th biennial conference of the Association for Machine Translation in the Americas*, Online.

Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G. M., and Passban, P. (2018). Investigating Back translation in Neural Machine Translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 249–258, Alicante, Spain.

Popović, M. (2011). Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *Prague Bulletin of Mathematical Linguistics*, 96:59–68.

Popović, M. (2015). chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Computational Linguistics*, 29:349–380.

Rubino, R., Foster, J., Kaljahi, R. S. Z., Roturier, J., and Hollowood, F. (2013). Estimating the Quality of Translated User-Generated Content. In *Proceedings of 6th International Joint Conference on Natural Language Processing*, pages 1167–1173, Nagoya, Japan.

Ruiter, D. (2019). Online parallel data extraction with neural machine translation. Masters thesis, Saarland University.

Sajjad, H., Darwish, K., and Belinkov, Y. (2013). Translating dialectal arabic to english. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1–6, Sofia, Bulgaria.

Scheible, C., Laws, F., Michelbacher, L., and Schütze, H. (2010). Sentiment translation through multi-edge graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1104–1112, Beijing, China.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Sennrich, R. and Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas*, pages 1–11, Denver, Colorado, USA.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy.

Shi, L., Niu, C., Zhou, M., and Gao, J. (2006). A DOM Tree Alignment Model for Mining Parallel Data from the Web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Sydney, Australia.

Simon, D., Castilho, S., Lohar, P., Afli, H., and Way, A. (2020). Transcasm: A bilingual corpus of sarcastic tweets. In *The 6th edition of "Using Corpora in Contrastive and Translation Studies Conference" (Accepted).*

Sluyter-Gäthje, H., Lohar, P., Afli, H., and Way, A. (2018). FooTweets: A bilingual parallel corpus of world cup tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1–5, Miyazaki, Japan.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Asso-*

ciation for Machine Translation in the Americas, pages 223–231, Massachusetts, USA.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Proceedings of Interspeech Conference*, volume 2002, pages 901–904, Colorado, USA.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Computing Research Repository*, abs/1409.3215.

Thompson, B. and Koehn, P. (2019). Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1342–1348, Hong Kong.

Toral, A. and Sánchez-Cartagena, V. M. (2017). A Multifaceted Evaluation of Neural versus Statistical Machine Translation for 9 Language Directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1063–1073, Valencia, Spain.

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 417–424, Philadelphia, Pennsylvania.

Uszkoreit, J., Ponte, J., Popat, A., and Dubiner, M. (2010). Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1101–1109, Beijing, China.

Utiyama, M. and Isahara, H. (2003). Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79, Sapporo, Japan.

Vanmassenhove, E., Shterionov, D., and Way, A. (2019). Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 222–232, Dublin, Ireland.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2007). Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing*, pages 247–258, Borovets, Bulgaria.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Thirty-first Conference on Neural Information Processing Systems*, pages 1–11, Long Beach, CA, USA.

Vicente, I. S., naki Alegria, I., na Bonet, C. E., Gamallo, P., Oliveira, H. G., Garcia, E. M., Toral, A., Zubiaga, A., and Aranberri, N. (2016). TweetMT: A Parallel Microblog Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 2936–2941, Portorož, Slovenia.

Šubert, E. and Bojar, O. (2014). Twitter crowd translation — design and objectives. In *Proceedings of the Translating and the Computer 36*, pages 217–227, London, United Kingdom.

Wolk, K., Rejmund, E., and Marasek, K. (2016). Multi-domain machine translation enhancements by parallel data extraction from comparable corpora. *Computing Research Repository*, abs/1603.06785.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *Computing Research Repository*, abs/1609.08144.

Yang, C. C. and Li, K. W. (2003). Automatic Construction of English/Chinese Parallel Corpora. *Journal of the American Society for Information Science and Technology*, 54:730–742.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Zaghouani, W., Mohit, B., Habash, N., Obeid, O., Tomeh, N., Rozovskaya, A., Farra, N., Alkuhlani, S., and Oflazer, K. (2014). Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 2362–2369, Reykjavik, Iceland.

Zhang, B., Nagesh, A., and Knight, K. (2020). Parallel corpus filtering via pre-trained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online.

Zhao, B. and Vogel, S. S. (2002). Adaptive parallel sentences mining from web bilingual news collection. In *Proceedings of the IEEE International Conference on Data Mining*, pages 745–748, Florida, USA.