Improving Transductive Data Selection Algorithms for Machine Translation

Alberto Poncelas Rodriguez

B.Sc., M.Sc.

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

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing

Supervisors:
Prof. Andy Way, Dr. Gideon Maillette de Buy Wenniger

2019

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:

(Candidate) ID No.: 15211130

Date: July, 2019

Contents

List of Figures

List of Tables

Abbreviations

ANN Artificial Neural Network

BCED Bilingual Cross-Entropy Difference

BLEU Bilingual Evaluation Understudy

BPE Byte Pair Encoding

CBOW Continuous Bag-of-Words Model

CD Context-Dependent

CED Cross-Entropy Difference

CI Context-Independent

CHRF Character n-gram F-score

DW Document-wise

DWDS Density Weighted Diversity Sampling

FDA Feature Decay Algorithms

GRU Gated Recurrent Unit

INR Infrequent N-gram Recovery

LM Language Model

LSTM Long Short-Term Memory

MERT Minimum Error Rate Training

METEOR Metric for Evaluation of Translation with Explicit Ordering

MT Machine Translation

NLP Natural Language Processing

NTA Non-transductive Algorithms

NMT Neural Machine Translation

PBSMT Phrase-Based Statistical Machine Translation

RNN Recurrent Neural Network

RQ Research Question

SGD Stochastic Gradient Descent

SMT Statistical Machine Translation

SW Sentence-wise

TA Transductive Algorithm

TER Translation Edit Rate

TF-IDF Term Frequency–Inverse Document Frequency

Improving Transductive Data Selection Algorithms for Machine Translation

Alberto Poncelas Rodriguez

Abstract

In this work, we study different ways of improving Machine Translation models by using the subset of training data that is the most relevant to the test set. This is achieved by using Transductive Algoritms (TA) for data selection. In particular, we explore two methods: Infrequent *N*-gram Recovery (INR) and Feature Decay Algorithms (FDA). Statistical Machine Translation (SMT) models do not always perform better when more data are used for training. Using these techniques to extract the training sentences leads to a better performance of the models for translating a particular test set than using the complete training dataset.

Neural Machine Translation (NMT) can outperform SMT models, but they require more data to achieve the best performance. In this thesis, we explore how INR and FDA can also be beneficial to improving NMT models with just a fraction of the available data.

On top of that, we propose several improvements for these data-selection methods by exploiting the information on the target side. First, we use the alignment between words in the source and target sides to modify the selection criteria of these methods. Those sentences containing *n*-grams that are more difficult to translate should be promoted so that more occurrences of these *n*-grams are selected. Another extension proposed is to select sentences based not on the test set but on an MT-generated approximated translation (so the target-side of the sentences are considered in the selection criteria). Finally, target-language sentences can be translated into the source-language so that INR and FDA have more candidates to select sentences from.

Acknowledgments

This thesis has been possible thanks to many people. First, I would like to express my deepest gratitude to Professor Andy Way for his guidance and supervision. I am also thankful for the support and advice of Dr. Gideon Maillette de Buy Wenniger, and Dr. Antonio Toral. In addition, I would like to thank my examiners Dr. Kevin McGuinness and Dr. Will Lewis, and chairperson Dr. Sharon O'Brien.

I embarked on this journey along with my colleagues Pintu Lohar, and Eva Vanmassenhove. For the past four years, we followed a similar path in which we never failed to support each other. My sincere gratitude also goes to my colleagues, and post-docs in ADAPT Centre, particularly Dimitar Shterionov, and Chao-Hong Liu who made this work much easier. Additionally, I want to thank Wichaya Pidchamook, who has been very helpful.

I will always feel indebted to Emilio Gedeón Ortiz-García for being an inspirational mentor and a good friend, and for encouraging me to do a PhD.

Last but not least, I would like to thank my family, particularly my parents Cesar Poncelas Poncelas, and Ana Maria Rodriguez Garcia for their endless support during these years, and indeed for my whole life.

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

Chapter 1

Introduction

Machine Translation (MT) is a subfield of machine learning that aims to generate the translation of sentences in one language into another language. In order to accomplish this, MT models are built using sentence pairs that are translations of each other. Models learn from these sentences so they can infer the translation of a new, unseen sentence or document.

As the translations produced by the models are typically post-edited by a professional translator, the quality of the generated translations is of crucial importance in order to minimize the amount of human effort.

Although one would think that by adding more sentence pairs, the model produces better translations, this is not necessarily true. It has been shown (Ozdowska and Way, 2009) that Statistical Machine Translation (SMT) models can perform better when trained with less data but in a closer domain to that of the test set. In order to do that, data-selection algorithms aim to retrieve the subset of data that is closer to a particular domain.

For this reason, data-selection techniques play a major role. These techniques aim not only to reduce the size of the models (and the time required for training), but also to identify the data that belong to a particular domain, so the model can be trained with in-domain data.

1.1 Document-Specific Machine Translation

1.1.1 Domain identification problem

In MT, test sets are usually sampled from a particular set of content (potentially a very large set of content) itself representative from a well-defined source, where that source may be labeled as a 'domain'. Test data are drawn from news sources (blogs, news websites, etc.) thus representing the 'News' domain, or a test data are drawn from medical sources thus representing the 'medical' domain, etc.

In some cases determining the domain of the test set can be difficult, and sometimes the test set can even belong to multiple domains. Fortunately, the selection of the sentence pairs can be executed without the need for identifying the domain. These data-selection methods that consider the test set as the seed in order to retrieve sentences are those classified as Transductive Algorithms (TAs).

1.1.2 Transductive Learning

TAs operate under a very different paradigm to the standard approaches used in machine learning, which are based on inductive learning. Inductive learning is concerned with reasoning from the particular (observed training data) to the general (functions that generalize well to unseen test data). In contrast, transductive learning (Vapnik, 1998) is concerned with the particular to the particular: in our case, from a corpus of annotated MT training data to a specific test set to be translated. Generalization outside this specific test set is not an objective of transductive learning, which can potentially allow transductive methods to outperform inductive models with respect to specific test sets.

As the definition of domain can range from general (e.g. News, Bio, etc) to more particular (such as author profiling), building document-specific MT systems is the most specific example of domain adaptation that might be contemplated.

The use of the test set has been previously investigated in other works. For example, Lu et al. (2007) propose to change the weight of those sentences that are

similar to the test set. Alternatively, they train several MT model candidates and use the test set to select the most suitable one to generate the translation. Biçici (2011) use the test set to select sentences to make *regression based machine translation* computationally more scalable. Lopez (2008) propose *Machine Translation by Patter Matching*, where those entries of the phrase table that match the phrases of the test set are retrieved.

1.1.3 Adaptation of Machine Translation Models

A translation company, when they need to translate a new document they use a MT engine that is the most suitable (e.g. trained in the same domain) to generate a translation. However, if at translation time the test set (or the document to be translated) is known, why not benefit from that? An MT model could be adapted to the current document. We propose to postpone part of the training phase of the MT model until the document to be translated is provided, which would minimize the time and human efforts required to post-edit the output of the MT.

The impacts of this view of the MT process are significant, in that two aspects which are central to how MT is done today are radically redefined:

- offline training is reduced or eliminated;
- the notion of poor quality 'noisy' data largely disappears;
- the notion of 'domain' becomes much more fine-grained and dynamic.

Furthermore, this will completely remove the major barrier and cost associated with MT: personalisation. Today, personalised MT is simply not a possibility, and even for larger institutions, customisation represents a major obstacle.

1.1.4 Cloud-Based Models

As hardware capability continues to improve, we foresee a paradigm shift in the not too distant future where cloud-based models are built on-the-fly in real time

for translation of specific documents. We envisage that following analysis of the translation requirements of the said document, the best-fitting examples in the entire cloud of translation data are selected as data for training of an MT system built specifically to translate that document. In such a scenario, we expect training time to be fast, as the amount of training data required will be small. In this way, we could even think of such models as disposable; once the specific document has been translated, there is no need to keep the MT system any longer.

Taken together, these improvements have the potential to transform the current MT landscape:

- *Speed*: translation systems will be built in real time;
- *Quality*: systems are dynamically adapted on-the-fly, based on the current translation task, and with incremental system-updating in real time during the post-editing process;
- *Personalisation*: training/customization always takes place online, in real time, to the user's specific requirements.

These three improvements are tightly interconnected; by permitting personalisation as a real-time process, we will achieve major improvements in translation quality and speed and considerably enhance the user experience.

1.2 Proposal of the Thesis

While building such an online real-time system-building set-up goes beyond the scope of this thesis, the importance of the optimal selection of the training data becomes paramount. Accordingly, in the context of this thesis, transduction is explored primarily via the use of data selection and data synthesis methods. The key idea is to choose examples from the training corpus that are similar in some way to the test corpus, and then use standard statistical and neural inferential models,

which will be biased toward performing well on the specific test set. This thesis demonstrates that this can lead to improved performance on the test sets.

We explore the performance[1] of TAs when used to build German-to-English MT models. Initially, we set out to explore the effect of TAs on the prevailing state-of-the-art in MT, namely SMT. More recently, of course, Neural Machine Translation (NMT) approaches have become popular as they can outperform SMT models. NMT models tend to perform better than SMT when larger amounts of data are used for training. Nonetheless, this work also shows that a subset of sentences retrieved by TAs can also be beneficial to improve the performance of NMT models.

In addition, we also propose several ways to improve TAs by exploiting information in the target side. These improvements come from three directions, either by (i) altering the selection criterion; (ii) altering the seed used for selecting sentences (use a translation of the test set instead of the test set); or (iii) generating new candidate sentences that TA can select from.

In sum, the primary focus of study in this thesis is the capability of TAs to restrict the amount of training data needed for the building of a high-quality MT system. However, it would be wrong to conclude that TAs can only be used for this task. Accordingly, at the end of the thesis, we start to consider the extent to which the class of algorithms classified in this thesis as TAs can be used inductively; can they be used to generalise over specific data sets and applied to new ones? Can they be utilised for domain adaptation?

1.3 Research Questions

The Research Question (RQs) we are addressing in this thesis aim to improve the performance of models trained on data selected by TAs. The RQs explored are:

1. **RQ1: How can we tailor data-selection algorithms to be most effec-**

[1]In this work, performance of a data-selection algorithm is used to refer to the translation quality of a model trained with the sentences retrieved by the algorithm, measured with automatic evaluation metrics.

tive in combination with NMT?

Although the TAs have a good performance in SMT they are yet unexplored in NMT. NMT approaches require larger amounts of data than SMT to achieve their best performances. For this reason, we want to explore whether these models could also benefit from TAs.

2. **RQ2: Can word-alignment information be useful for improving state-of-the-art TAs?**

   The TAs analyzed in this thesis penalize the n-grams of sentences that have already been selected in order to increase the variability. However, should every n-gram be penalized equally? In every language, there are words or n-grams that are more difficult to be translated and therefore more occurrences are needed in order to learn the proper translation. A way of measuring how complicated is to decide the translation of an n-gram is by computing the alignment entropy, which measures the predictability of the translation by analyzing how the words in the n-gram are mapped to the words in the target side.

3. **RQ3: Can the use of synthetic sentences improve the performance of MT models when used in combination with TAs?**

   Another method to improve the quality of the models is to acquire more candidates sentences to selected from. When additional data are not available there is the option of creating sentences artificially. By doing this, we can augment the size of the candidate pool. We want to explore whether using synthetic data alone or in combination with authentic data is more beneficial than using authentic data only.

   One limitation of the explored TAs is that they select the sentences based on the n-grams in the source side, ignoring the target-side completely. We propose to use synthetic target-side sentences as the seed of TAs so that the

selection is also performed considering the target side. By doing this we want to minimize the effect of selecting noisy sentences (sentence pairs that are not accurate translations of each other) and promote selecting the same *n*-grams in the target side.

1.4 Contributions

In this thesis we use TAs applied in MT and there are several contributions in terms of the exploration and improvements of these methods. Here we present the main contributions of the thesis, but a more detailed list can be found in the introduction of each chapter:

- We perform comparisons of SMT and NMT models that have been trained with different amounts of data.
- We compare different SMT and NMT models using subsets of the training sentences retrieved by TAs.
- We perform an analysis of the performance of different configurations of TAs in SMT and NMT, and explore the impact of changing the values of the parameters of TAs.
- We propose a novel extension for two TAs so that the decay of the *n*-grams (used to promote the variability) becomes dynamic, and so different *n*-grams are penalized differently.
- We discuss the disadvantages of selecting parallel sentences with TAs based only in the source side and introduce a novel technique to execute these methods so they select sentences considering the target side instead.
- We investigate how authentic and synthetic training data can work in combination with TAs to build better models. In addition, we propose two ways of

selecting synthetic data with TAs and how to combine them with the authentic selected-data.

1.5 Outline of the Thesis

This thesis is structured in the following chapters:

- **Chapter 2 (Background)** introduces some concepts that will be used later on in the thesis. In addition, we describe the two leading MT paradigms Phrase-Based Statistical Machine Translation (PBSMT) and NMT, as the experiments carried out involve building models following these approaches. In addition, we provide an overview of the main data-selection algorithms and describe their main characteristics.

- **Chapter 3 (Transductive Algorithms on Statistical Machine Translation)** presents several experiments for a better understanding of PBSMT models. The chapter includes experiments that explore the impact of adding training sentences to build the models. Additionally, in the chapter we investigate the performance when SMT models are built with data selected from TAs. This chapter is also important as we establish the models considered as baselines for SMT in the thesis.

- **Chapter 4 (Transductive Algorithms on Neural Machine Translation)** reports the performance of NMT models when trained with data from TAs. This chapter addresses RQ1. We compare NMT models trained with different sizes of either randomly-selected or TA-selected data. We explore two different ways of using selected data in NMT: (i) using it to build models from scratch; and (ii) using it to tune general-domain models. This chapter also helps establish the baselines to be used in our the experiments, as well as describe how NMT models are constructed in the following chapters.

- **Chapter 5 (The Use of Alignment Entropy)** presents an analysis of the impact of different configurations when used in SMT and NMT. We propose a method to improve the selection criteria of TAs. In particular, we aim to answer RQ2. We suggest three methods to compute alignment entropies and we evaluate TAs when using these values for their parameters.

- **Chapter 6 (The Use of Synthetic Data to Adapt Models)** investigates a set of experiments in which artificial data are involved to answer RQ3. This chapter evaluates the models fine-tuned with synthetic sentences only, as well as in combination with authentic ones.

- **Chapter 7 (Conclusions and Future Work)** summarizes the work conducted in terms of the RQs proposed in the thesis. Finally, we propose several ways to further explore the techniques proposed in this work.

1.6 Publications

The contents of this thesis are based on work published in peer-reviewed international conferences. The papers that are the most related are the following:

1. Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2019b). Transductive data-selection algorithms for fine-tuning neural machine translation. In *The 8th Workshop on Patent and Scientific Literature Translation (PSLT 2019)*, Dublin, Ireland

2. Poncelas, A., Maillette de Buy Wenniger, G., and Way, A. (2018b). Feature decay algorithms for neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 239–248, Alacant, Spain

3. Poncelas, A., Way, A., and Toral, A. (2016). Extending feature decay algorithms using alignment entropy. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 170–182, Seville, Spain. Springer

4. Poncelas, A., Maillette de Buy Wenniger, G., and Way, A. (2017). Applying n-gram alignment entropy to improve feature decay algorithms. *The Prague Bulletin of Mathematical Linguistics*, 108(1):245–256

5. Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G. M., and Passban, P. (2018c). Investigating backtranslation in neural machine translation. In *21st Annual Conference of the European Association for Machine Translation*, pages 249–258, Alacant, Spain

6. Poncelas, A., Popovic, M., Shterionov, D., de Buy Wenniger, G. M., and Way, A. (2019c). Combining SMT and NMT back-translated data for efficient NMT. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 922–931, Varna, Bulgaria

7. Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2019a). Adaptation of machine translation models with back-translated data using transductive data selection methods. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France

8. Poncelas, A., Way, A., and Sarasola, K. (2018d). The ADAPT System Description for the IWSLT 2018 Basque to English Translation Task. In *International Workshop on Spoken Language Translation*, pages 72–82, Bruges, Belgium

9. Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2018a). Data selection with feature decay algorithms using an approximated target side. In *15th International Workshop on Spoken Language Translation (IWSLT 2018)*, pages 173–180, Bruges, Belgium

10. Poncelas, A. and Way, A. (2019). Selecting Artificially-Generated Sentences for Fine-Tuning Neural Machine Translation. In *Proceedings of the 12th International Conference on Natural Language Generation*, Tokyo, Japan

In addition to that, there are other papers published in peer-reviewed conferences in the Natural Language Processing (NLP) field that I have co-authored:

1. Poncelas, A., Sarasola, K., Dowling, M., Way, A., Labaka, G., and Alegria, I. (2019d). Adapting NMT to caption translation in Wikimedia Commons for low-resource languages. In *35th International Conference of the Spanish Society for Natural Language Processing (SEPLN 2019)*, Bilbao, Spain

2. Vanmassenhove, E., Moryossef, A., Poncelas, A., Way, A., and Shterionov, D. (2019). ABI Neural Ensemble Model for Gender Prediction Adapt Bar-Ilan Submission for the CLIN29 Shared Task on Gender Prediction. In *Computational Linguistics of the Netherlands CLIN29*, Groningen, The Netherlands (Share task winner paper)

3. Dowling, M., Lynn, T., Poncelas, A., and Way, A. (2018). SMT versus NMT: Preliminary comparisons for Irish. In *Technologies for MT of Low Resource Languages (LoResMT 2018)*, page 12, Boston, USA

4. Silva, C. C., Liu, C.-H., Poncelas, A., and Way, A. (2018). Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231, Brussels, Belgium

5. Liu, C.-H., Moriya, Y., Poncelas, A., and Groves, D. (2017b). IJCNLP-2017 Task 4: Customer Feedback Analysis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 26–33, Taipei, Taiwan

6. Dzendzik, D., Poncelas, A., Vogel, C., and Liu, Q. (2017). ADAPT centre cone team at IJCNLP-2017 task 5: A similarity-based logistic regression approach to multi-choice question answering in an examinations shared task. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 67–72, Taipei, Taiwan (Share task winner paper)

7. Liu, C.-H., Groves, D., Hayakawa, A., Poncelas, A., and Liu, Q. (2017a). Understanding meanings in multilingual customer feedback. In *Proceedings of First*

Workshop on Social Media and User Generated Content Machine Translation (Social MT 2017), Prague, Czech Republic

Chapter 2

Background

In this chapter, we introduce concepts related to the NLP field that will be used later and provide a background of MT field. In particular, we explain the main state-of-the-art approaches, PBSMT (Koehn et al., 2003) and NMT (Cho et al., 2014; Sutskever et al., 2014), that have been explored in this work.

In addition, as this thesis is related to the data-selection field, in Section 2.5 we summarize the main data-selection techniques. Note that we classify and provide a general overview of major data-selection techniques. Then, in Section 2.5.2 we describe in more detail the transductive methods which are used in the experiments carried out in this thesis.

2.1 Mathematical and NLP Concepts

First, we introduce some mathematical notation and NLP concepts that will be used in this work.

***N*-gram functions** An n-gram is a sequence of n contiguous elements extracted from a longer string. Typically, these elements consist of characters or words. Unless otherwise specified, in this thesis we consider an n-gram to be a sequence of words.

We define $Ngr_{ij}(s)$ as the set of n-grams (where $i \leq n \leq j$) in a sentence s. Similarly, we use the notation $Ngr_{ij}(D)$ as the set of n-grams in a set of sentences D

(equivalent to $Ngr_{ij}(D) = \bigcup_{s \in D} Ngr_{ij}(s)$). For simplification, in the case in which $i = 1$, we will indicate only the value of j in the subscript, so $Ngr_j(s) = Ngr_{1j}(s)$.

Moreover, we use $C_s(ngr)$ as the count of occurrences of an n-gram ngr in a sentence s, and $C_D(ngr) = \sum_{s \in D} C_s(ngr)$ the number of occurrences of ngr in the set of sentences D.

The probability of the occurrence of an n-gram in a set of sentences D is calculated as in Equation (2.1):

$$P_D(ngr) = \frac{C_D(ngr)}{\sum\limits_{ngr_i \in Ngr_{nn}(D)} C_D(ngr_i)}. \tag{2.1}$$

We use $|s|$ to express the number of words contained in the sentence s and $words(D) = \sum\limits_{s \in D} |s|$ as the number of words in the set D.

Parallel Data In the MT field we typically use parallel sentences to build MT models. This data consists of a pair $\langle S, T \rangle$, where S is a set containing sentences in the source language and T a set containing sentences in the target language. These sentences are paired so the i-th sentence $s_i \in S$ and $t_i \in T$ are a translation of each other. The pair $\langle S, T \rangle$ can also be considered a set of unique parallel sentence-pairs $\langle s_i, t_i \rangle$, so we also refer to it as a parallel set.

Following the terminology of the MT field, we also refer to the subsequence of words of a sentence s as a phrase $\bar{s}$. From a sentence pair $\langle s_i, t_i \rangle$ we can also extract phrase-pairs $\langle \bar{f}, \bar{e} \rangle$ (where $\bar{f}$ is a phrase from s_i and $\bar{e}$ is a phrase from t_i). We use $C_{(S,T)}(\bar{f}, \bar{e})$ for counting the number of sentences in which the phrase $\bar{f}$ and $\bar{e}$ occur together in $\langle S, T \rangle$.

Entropy Entropy is a measure of the uncertainty. It is used to evaluate the predictability of the outcomes of a random process. The entropy of a random variable $X = x_1, ...x_N$, where the probability of each outcome x_i is $P(x_i)$, is computed as in

Equation (2.2):

$$entropy(X) = -\sum_{x_i \in X} P(x_i) \log(P(x_i)). \tag{2.2}$$

A larger entropy means that the uncertainty is higher and lower entropies indicate that the outcomes are more predictable. If there is only one predictable outcome ($P(x_1) = 1$), then the entropy is 0. The entropy value can be normalized to be in the range $[0, 1]$ when computed as in Equation (2.3):

$$entropy(X) = -\frac{1}{|X|} \sum_{x_i \in X} P(x_i) \log(P(x_i)). \tag{2.3}$$

TF-IDF Term Frequency–Inverse Document Frequency (TF-IDF) (Salton and Yang, 1973) is a statistic that indicates how relevant a word is for a document in relation to a set of documents. The weight of a term is higher if it is frequent in a document d, but it is also penalized if is also frequent in the other documents of the collection D.

The TF-IDF value of word w_k in a document $d \in D$ is computed as in Equation (2.4):

$$tfidf(w_k, d, D) = C_d(w_k) \log(idf(w_k, D)) \tag{2.4}$$

where idf_k is the inverse document frequency (IDF). This measures the inverse of the frequency of the k-th term in the set of all documents D, computed as $idf(w_k) = \frac{|D|}{|D_{w_k}|}$ where D_{w_k} is the set of documents containing w_k.

TF-IDF is often used as a distance metric between two documents. A document can be seen as a vector $\mathbf{d}$ where each element d_k is $tfidf(w_k, d, D)$. The TF-IDF distance of two vector of two documents $\mathbf{d}_{(1)}$ and $\mathbf{d}_{(2)}$ is defined as the cosine distance between the two vectors computed as in Equation (2.5):

$$dist_{tfidf}(\mathbf{d}_{(1)}, \mathbf{d}_{(2)}) = \cos(\mathbf{d}_{(1)}, \mathbf{d}_{(2)}) = \frac{\mathbf{d}_{(1)} \cdot \mathbf{d}_{(2)}}{|\mathbf{d}_{(1)}||\mathbf{d}_{(2)}|}. \tag{2.5}$$

Language models Language Models (LMs) are models that measure the fluency of a sentence, i.e. how likely the sentence is to have been produced by a native speaker of the language. *N*-gram LMs are based on statistics that indicate how likely words are to follow each other.

However, Equation (2.1) is not useful to estimate the probability as ngr may not be found in D (which is likely if the sequence is too long). Therefore, *n*-gram LMs split this process (smaller statistics) using the chain rule and aim to predict one word at a time as in Equation (2.6):

$$P_{LM}(w_1, w_2, ..., w_{l-1}, w_l) = P(w_1)P(w_2|w_1)...P(w_l|w_1, w_2, ..., w_{l-1}). \tag{2.6}$$

The terms of Equation (2.6) compute the probability of a word conditioned to the sequence of all previous words. Following the Markov assumption, each term $P(w_i|w_1...w_{i-1})$ is approximated as in Equation (2.7):

$$P(w_i|w_1, w_2, ..., w_{i-1}) \approx P(w_i|w_{i-h}, w_2, ..., w_{i-1}) \tag{2.7}$$

where, instead of considering all the previous word of the sequence, only the previous h words are considered. We call h the order of the LM. Each term of Equation (2.7) is computed as in Equation (2.8):

$$P_{LM_d}(w_i|w_{i-h}, ..., w_{i-1}) = \frac{C_D(w_{i-h}, ..., w_i)}{C_D(w_{i-h}, ..., w_{i-1})}. \tag{2.8}$$

In order to evaluate an LM, two metrics are typically used: cross-entropy and perplexity. This metrics estimate how well an LM can predict a sequence of words s. The first metric, cross-entropy, is the average log probability of the words in s computed as in Equation (2.9):

$$H_{LM_d}(s) = -\frac{1}{|s|}\sum_{i=1}^{|s|}\log(P_{LM_d}(w_i|w_{i-n}, ..., w_{i-1})). \tag{2.9}$$

The other metric, perplexity, is a transformation of cross-entropy as in Equation (2.10):

$$PP_{LM_d}(s) = 2^{H_{LM_d}(s)}. \tag{2.10}$$

2.2 Statistical Machine Translation

SMT is the MT paradigm in which the translation problem is considered as a statistical optimization problem.

The noisy channel model is a framework of communication where a message is sent from the source to a receiver through a channel which causes the message to suffer a distortion. SMT is based on this framework as it assumes a sentence f in a source language is transformed into a sentence e in the target language when transmitted through the noisy channel. The goal is to infer the translated sentence e from f with the highest probability as in Equation (2.11):

$$e^* = \arg\max_e P(e|f) \tag{2.11}$$

which following Bayes' theorem, can be expressed as in Equation (2.12) (noisy channel model):

$$P(e|f) \propto P(f|e)P(e) \tag{2.12}$$

where we observe two main components:

- $P(f|e)$, translation model probability, which measures adequacy, i.e how much of the meaning is preserved in the translation. This model is built based on bilingual data.
- $P(e)$, language model probability, which measures fluency, i.e. how likely the translation is to have been produced by a native speaker of that language. This model is built based on monolingual (target-side) data.

2.2.1 Word-Based Statistical Machine Translation

Word-based SMT (Brown et al., 1993) is the statistical translation approach that uses words as atomic translation units. It introduced the concept of word alignment, a function defining one-to-one and many-to-one mappings between words of the sentence pairs.

Given a sentence $f = (f_1, ..., f_{l_s})$ in the source language and a sentence $e = (e_1, ..., e_{l_t})$ in the target language, the alignment function a maps each word e_j in the target-side to a word f_i in the source side along with the translation probability. The most popular tools for word alignments are GIZA++ (Och and Ney, 2003) and its variation FastAlign alignment model (Dyer et al., 2013) which introduces a diagonal tension λ. This parameter measures the overall correspondence of word order and an efficient re-estimation of the parameters that makes it around 10 times faster than GIZA++ while still obtaining comparable quality.

2.2.2 Phrase-Based Statistical Machine Translation

PBSMT models (Koehn et al., 2003) may be considered to be an improvement over word-based SMT models. These models use phrases as atomic units for translation (as opposed to individual words). This approach is better able to capture contextual information. Phrases from a source and a target sentence are paired so that every word of the phrase in one side is aligned to a word present in the phrase of the other side or a ⟨NULL⟩ token (but not to words outside the phrase).

The phrase pairs are gathered along with their translation scores in a structure called a phrase-table, which will be used in the decoding step (when translating a document) as a look-up dictionary, for selecting a translation of a phrase of the test set.

The decision to select a phrase pair is based mainly on three components: (i) a translation model (scores of the phrase-table), (ii) a reordering model, and (iii) a LM:

- Translation model: It provides the translation probabilities of a phrase pair (an entry of the phrase table). Commonly 4 scores are computed: "inverse phrase translation probability" ($\phi(f|e)$), "inverse lexical weighting" ($lex(f|e)$), "direct phrase translation probability" ($\phi(e|f)$) and "direct lexical weighting" ($lex(e|f)$)

  - Translation probability: It indicates the probability of a phrase to be the translation of another phrase computed as in Equation (2.13):

$$\phi(\bar{f}|\bar{e}) = \frac{C_{(S,T)}(\bar{f}, \bar{e})}{\sum_{\bar{f}_i} C_{(S,T)}(\bar{f}_i, \bar{e})} \tag{2.13}$$

    where $\bar{f}$ and $\bar{e}$ are the source and target phrase pairs, respectively.

  - Lexical weighting: This is computed in order to avoid the problem of phrases that do not provide reliable probability estimations (e.g. low-frequency phrases). It measures how well the words in the phrases translate to each other (Koehn et al., 2003) computed as in Equation (2.14):

$$p_w(\bar{f}|\bar{e}, a) = \prod_{i=1}^{l} \frac{1}{\{j|(i,j) \in a\}} \sum_{\forall(i,j)\in a} w(f_i|e_j) \tag{2.14}$$

    where $w(f_i|e_j)$ is the lexical weighting defined as in Equation (2.15):

$$w(f_i|e_j) = \frac{C_{(S,T)}(f_i, e_j)}{\sum_{f'} C_{(S,T)}(f', e_j)} \tag{2.15}$$

    where f' are the words in the source language aligned to e_j

- Reordering model: Introduced by Tillmann (2004), this is the model that handles the orientation of a phrase based on the previous adjacent phrase. Koehn et al. (2005) estimates the probability of three different orientations for a phrase: monotone (how likely the phrase follows the previous one), swap (how likely is swapped with the previous one) and discontinuous (how likely it is not to be connected to the previous one).

- LM: An LM models the fluency of the output of the translation computing the probability of a sequence of words as in Equation (2.8):

These scores computed are combined in a weighted logarithmic sum (known as the log-linear model) as in Equation (2.16):

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x). \tag{2.16}$$

As we see in Equation (2.16), the feature functions are weighted according to λ_i. After computing the feature functions, the optimal value of each λ_i needs to be found. The process of finding appropriate values of λ_i is known as *tuning*.

One popular method for tuning is Minimum Error Rate Training (MERT) (Och, 2003). MERT uses a *development set*, a set of parallel sentences not included in the training data, to estimate the optimal weights. In order to estimate them, first initial random values are set, and then several runs are executed (until convergence). In each iteration:

1. Translations of the sentences in the *dev set* are produced and the error of the n-best sentences (using the reference translations) are computed.

2. Each parameter is optimized individually (fixing the values of the other parameters).

2.2.3 Moses Toolkit

The PBSMT tool we use in this work is the Moses Toolkit (Koehn et al., 2007). This tool takes a set of parallel sentences and a language model as input and trains an SMT system. It computes the models explained in Section 2.2.2 and produces a file, *moses.ini*, that contains the features as shown in Listing 2.1.

The first part of the *moses.ini* file contains the paths to the components explained in Section 2.2.2 (translation model, lexical reordering and language model) and other

```
UnknownWordPenalty
WordPenalty
PhrasePenalty
PhraseDictionaryMemory name=TranslationModel0 num-features=4
    path=/path/to/phrase-table.gz input-factor=0 output-factor=0
LexicalReordering name=LexicalReordering0 num-features=6
    type=wbe-msd-bidirectional-fe-allff
    input-factor=0 output-factor=0
    path=/path/to/reordering-table.wbe-msd-bidirectional-fe.gz
Distortion
KENLM lazyken=0 name=LM0 factor=0 path=/path/to/LM order=8


LexicalReordering0= 0.0899023 0.0589253 0.0456796 0.0879397
    0.000122106 0.135896
Distortion0= 0.0294993
LM0= 6.15097e-05
WordPenalty0= -0.0164713
PhrasePenalty0= -0.29107
TranslationModel0= 0.000119481 0.0207173 0.222799 -0.000797186
UnknownWordPenalty0= 1
```

Listing 2.1: Extraction of moses.ini file using default configuration of Moses

```
wenn wir ||| when we ||| 0.2006 0.1772 0.110 0.1551 ||| 0-0 1-1 ||| 648 1177 130 ||| |||
wenn wir ||| when ||| 0.0006015 0.0007428 0.0051 0.1995 ||| 0-0 ||| 9974 1177 6 ||| |||
wenn wir ||| whenever we ||| 0.1818 0.1851 0.003398 0.003376 ||| 0-0 1-1 ||| 22 1177 4 ||| |||
wenn wir ||| where we ||| 0.003875 0.01148 0.0008496 0.005927 ||| 0-0 1-1 ||| 258 1177 1 ||| |||
wenn wir ||| while we ||| 0.01492 0.009151 0.00085 0.002926 ||| 0-0 1-1 ||| 67 1177 1 ||| |||
```

Listing 2.2: Extraction of phrase table file

features such as word and phrase penalty (so the translations are not too long or too short), unknown word penalty and distortion (Brown et al., 1993).

The second part of the file contains the weights of the features (λ_i values in Equation (2.11)). In Listing 2.1 we show the values after tuning.

We can observe in the first part of the *moses.ini* file (Listing 2.1) that the translation model (PhraseDictionaryMemory), reordering model (LexicalReordering) and language model (LM) indicate the files where these models are stored. Note that the translation model and reordering model files have been created by Moses, but the language model is created separately and then provided to Moses at training time.

The translation model is stored in a file called *phrase table*. We show an extraction in Listing 2.2. This file contain five columns (the separator of the table is "|||")

1. Phrase in the source side.

2. Phrase in the target side: the phrase in the target side language that is paired with the source side phrase.

3. Translation model features: The four probabilities explained in Section 2.2.2 in this order: inverse phrase translation probability, inverse lexical weighting, direct phrase translation probability, and direct lexical weighting. We describe the first row of Listing 2.2 as an example of how they are computed:

    - Inverse phrase translation probability ($\phi(f|e)$): this is computed as in Equation (2.13). The counts of occurrences of the phrases are shown in column 5 ("when we" occurs 648 times, "wenn wir" and "when we" occur together 130 times). Therefore the inverse phrase translation probability is $0.2006 = 130/648$.

    - Inverse lexical weighting ($lex(f|e)$): this is computed as in Equation (2.14). The individual lexical weighting is stored in a file called *lex.e2f* created by Moses. In this file we find the values of lexical weighting for the words in the phrases, in the rows *wenn when 0.2658521* and *wir we 0.6666557*. Therefore the inverse lexical weighting is $0.1772 = 0.2658521 \cdot 0.6666557$.

    - Direct phrase translation probability ($\phi(e|f)$): this is computed as in Equation 2.13. "wenn wir" occurs 1177 times, "wenn wir" and " when we" occur together 130 times. Therefore the direct phrase translation probability is $0.110 = 130/1177$.

    - Direct lexical weighting ($lex(e|f)$): this is computed as in Equation (2.14). The individual lexical weighting is stored in a file called *lex.f2e* which contains the rows *when wenn 0.1995174* and *we wir 0.7773823*. Therefore the direct lexical weighting is $0.1551 = 0.1995174 \cdot 0.7773823$.

4. Alignments: How words of the source and target side are aligned individually. For example, in the last row, the pair ⟨"wenn wir , when we"⟩ , "0-0" indicate that the 0-th word in the source side word ("wenn") is aligned to the 0-th target-side word ("when").

5. Phrase counts: Three numbers consisting of target phrase counts, source

```
wenn wir ||| when ||| 0.200000 0.066667 0.733333 0.333333 0.066667 0.600000
wenn wir ||| whenever we ||| 0.272727 0.090909 0.636364 0.090909 0.090909 0.818182
wenn wir ||| where we ||| 0.200000 0.200000 0.600000 0.200000 0.200000 0.600000
wenn wir ||| while we ||| 0.200000 0.200000 0.600000 0.200000 0.200000 0.600000
```

Listing 2.3: Reordering table

phrase counts, source and target intersection count. For example, in the first example “wenn wir , when we”, there are 1177 occurrences of “wenn wir”, and 648 occurrences of “when we”. The phrases “wenn wir” and “ when we” occur together 130 times,

The reordering model is also stored in a separate file. We show an extraction in Listing 2.3. This file contain three columns (the separator of this table is also “|||”):

1. Phrase in the source side.

2. Phrase in the target side. The phrase in the target side language that is paired with the source side phrase.

3. Orientation probabilities: Six probabilities in two sets indicating the orientation (monotone, swap and discontinuous) in both directions (left-to-right and right-to-left), with each set of probability summing to 1.

2.3 Neural Machine Translation

More recently, NMT approaches have become more popular than SMT. Instead of using phrases as translation units like in PBSMT models, in NMT approaches, the sentences are encoded as vectors. These vectors are inputs (and outputs) of a network whose nodes models a function. In the training process, the parameters of the functions are adjusted so the returned vector encodes the sentence corresponding to the translation of the input.

2.3.1 Word Vector Models

A straightforward technique to encode a word as a vector is via the so called one-hot vector encoding. Assuming a vocabulary of size V we can encode the i-th word in the vocabulary as vector $v \in \mathbb{R}^{|V|}$ with 1 in the i-th position and 0 in the other positions.

However, this is a sparse representation in which there is no relationship between the words. For example, given this representation, the distance between two related words such as *football* and *basketball* is the same as *football* and *plane*, even though, the last two are semantically more different than the first pair. Word embeddings aim to find the position of the words in the vector space so that similar words are grouped close to one another.

In Mikolov et al. (2013) two methods for computing word embeddings are proposed (Figure 2.1):

- Continuous Bag-of-Words Model (CBOW): This model tries to predict a word $\mathbf{w}_t$ given the context. This means that the previous sequence of words (vectors $\mathbf{w}_{t-1}...\mathbf{w}_{t-i}$) and the following sequence of words (vectors $\mathbf{w}_{t+1}...\mathbf{w}_{t+i}$)) are used as input to a neural network architecture.

- Skip-gram Model: The objective is to predict the context words given a word $\mathbf{w}_t$. It improves the quality of the resulting word vectors, but it also increases the computational complexity (Mikolov et al., 2013).

The main benefit of these representations is the generalization it brings. Similar words will have similar vectors and so the distance between them will be small (as opposed to one-hot vectors where all vectors are equidistant from one another). Another benefit is that we can represent the words using a lower dimensional vector.

2.3.2 Artificial Neural Networks

Inspired by biological neural networks, Artificial Neural Network (ANN) are computing systems that are made up of interconnected processing elements (called per-

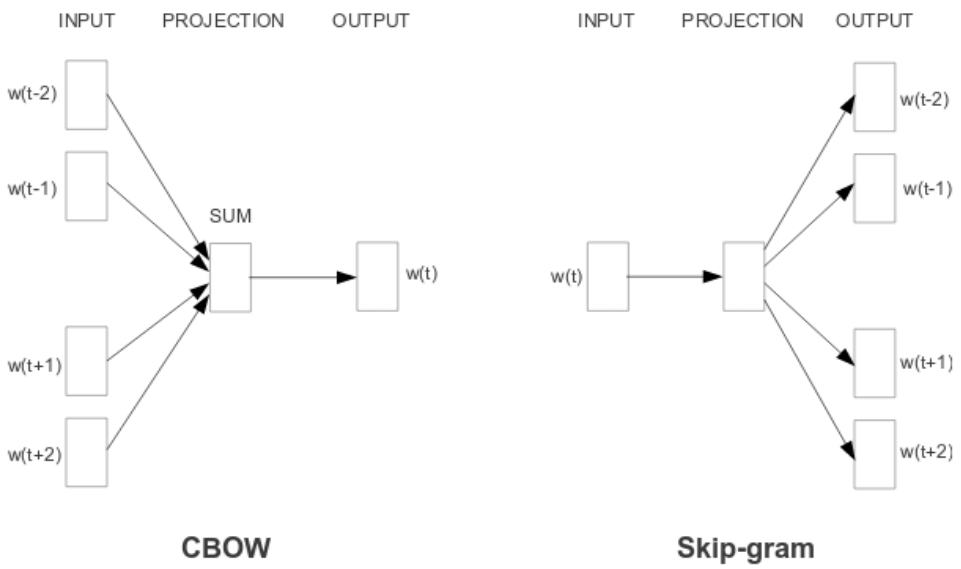

Figure 2.1: CBOW and Skip-gram Word Embedding Models. (Mikolov et al., 2013)

ceptrons).

The networks receive a series of inputs $x_1, x_2...x_{d_x}$ (which can be expressed as elements of a vector $\mathbf{x} \in R^{d_x}$). These inputs are processed by perceptrons and then a series of outputs $y_1, y_2, ...y_{d_y}$ (which can be seen as elements of a vector $\mathbf{y} \in R^{d_y}$) are produced.

In Figure 2.2 we see an example of a simple perceptron. The perceptron applies a weighted sum of the inputs (for simplification purposes we omit the bias), activation function and then feeds forward the results.

An example of a simple network is the multilayer perceptron in which perceptrons are structured in layers, as in Figure 2.3, that are classified in three types:

- Input Layer: A layer in which the perceptrons receive the input and feed it to the next layer.
- Hidden layer (one or several): A layer in which perceptrons gather the inputs from the previous layer (which can be the input layer or another hidden layer), perform the computations and feed the result to the next layer.
- Output Layer: A layer in which perceptrons perform the computations and

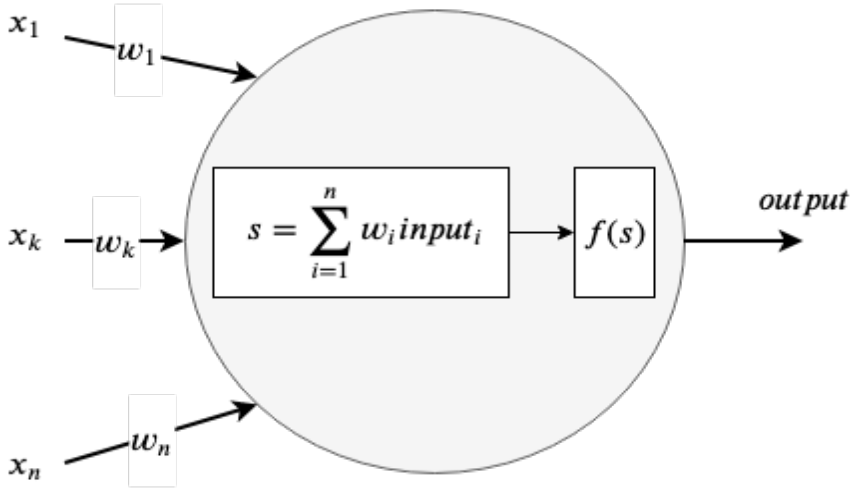

Figure 2.2: Example of a perceptron.

provide the final output of the function that the network is approximating.

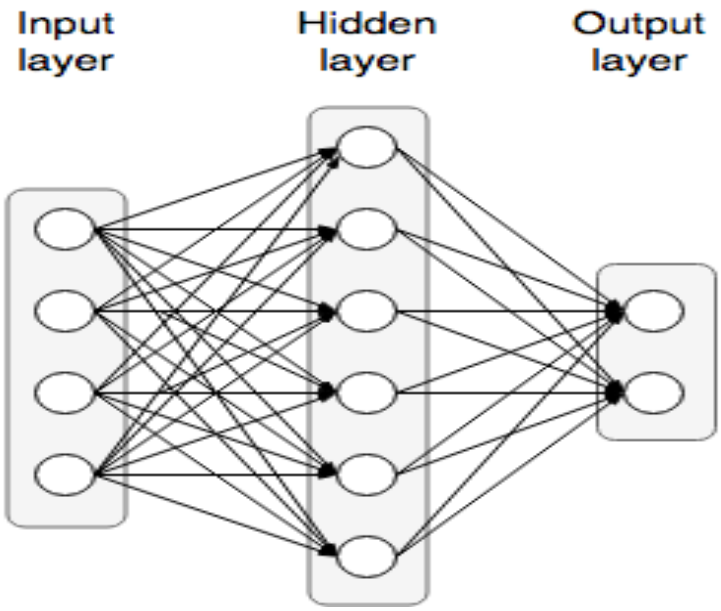

Figure 2.3: Example of an ANN.

As perceptrons are structured on layers, each layer can be modeled as in Equation (2.17):

$$\mathbf{h} = f(\mathbf{W}_{ih}\mathbf{x}) \tag{2.17}$$

where $\mathbf{x}$ is the output of the previous layer (each element in the vector is the output of one perceptron) or the input vector. $f(x)$ is a non-linear activation function such

as the $f(x) = \frac{1}{(1+e^{-x})}$ (also known as sigmoid function or σ) or $f(x) = tanh(x)$. The matrix $\mathbf{W}_{ih} \in \mathbb{R}^{|\mathbf{h}|\times|\mathbf{x}|}$ is composed of the individual weights w_{ij} of the link connecting the output of the j-th perceptron of the previous layer (i.e. $\mathbf{x}_j$) with the i-th perceptron in the hiden layer ($\mathbf{h}_i$).

The output layer is modeled as in Equation (2.18):

$$\mathbf{y} = g(\mathbf{W}_{ho}\mathbf{h}) \tag{2.18}$$

$g(\mathbf{x})$ is also a non-linear activation function. In the case of using the ANN as classification function a popular function of $g(\mathbf{x})$ is the softmax function $g(x_i) = \frac{e^{x_i}}{\sum_j e^{x_i}}$. This will produce an output $\mathbf{y}$ that encodes the probabilities so each y_i encodes the probability of the input belonging to the i-th class (which implies that $\sum y_i = 1$).

The purpose of an ANN is to approximate a function $\phi(\mathbf{x})$. When a certain input $\mathbf{x}$ is provided to the input layer of the ANN, the information is propagated to the next hidden layer. Each hidden layer performs the computations (modeled in Equation (2.17)) and emit the signal to the next layer. Eventually, the output layer will provide an output $\mathbf{y}$. This process is known as *forward propagation*. After executing the forward propagation, the error $\phi(\mathbf{x}) - \mathbf{y}$ is computed. Then, the gradient of the error with respect to the weights $\mathbf{W}$ of the different layers is calculated. This process is known as *backpropagation* as it is computed backward from the output layer to the input layer. Finally, using an optimizer such as Stochastic Gradient Descent (SGD), the error can be reduced by changing the weights in the direction of the gradient.

ANNs are trained using a set of pairs $(\mathbf{x}, \phi(\mathbf{x}))$. After adapting the weights by performing several iterations of *forward propagation* and *backpropagation*, the outputs of the ANN converge to an approximation of the function $\phi(\mathbf{x})$.

2.3.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are ANNs that form directed cyclic graphs. Having this structure allows the RNN to compute a sequence of vectors instead of a single vector.

Each vector $\mathbf{x}_t$ in the sequence $(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{T_x})$ is sequentially processed (Figure 2.4). At the step of processing $\mathbf{x}_t$, the information of the previous output vector h_{t-1} is also gathered. Then, the hidden state $\mathbf{h}_t$ and the output $\mathbf{y}_t$ are produced. The diagram presented in Figure 2.4 can be unfolded to take time out of the equation as in Figure 2.5.

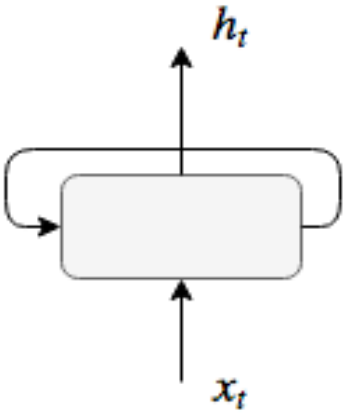

Figure 2.4: Example of an RNN.

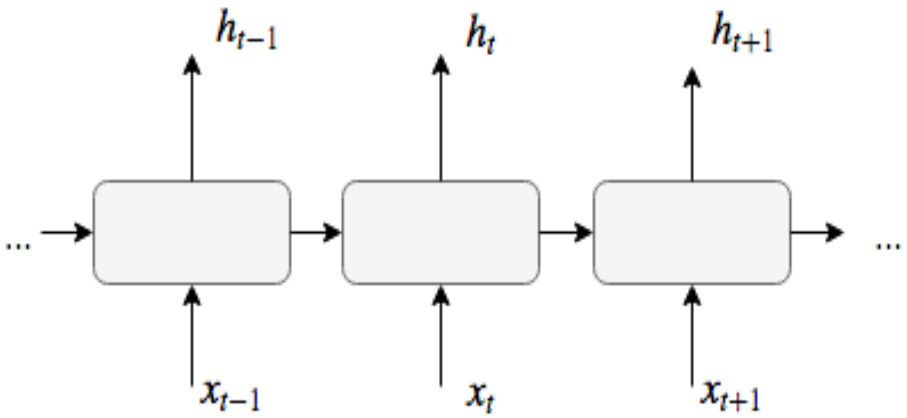

Figure 2.5: Example of an unfolded RNN.

As each cell of the RNN has two inputs (the input $\mathbf{x}_t$ and information of the

previous hidden layer $\mathbf{h}_{t-1}$), Equation (2.17) is extended as in Equation (2.19):

$$\mathbf{h}_t = f(\mathbf{W}_{ih}\mathbf{x}_t + \mathbf{U}_{ih}\mathbf{h}_{t-1}) \tag{2.19}$$

where $\mathbf{W}_{ih}$ and $\mathbf{U}_{ih}$ are the weight matrices to be adjusted during training of the RNN.

RNNs obtain the information of the previous hidden state h_{t-1} when processing x_t, which implicitly contains the information of the previous elements of the sequence. However, the long-distance dependencies are more difficult to learn the more the bigger the gap between x_t and a previous element x_{t-k} is.

In order to solve this long-distance dependency problem Long Short-Term Memories (LSTM) (Hochreiter and Schmidhuber, 1997) was proposed as a variation of RNNs. As the experiments involving RNNs executed in this work consist of LSTM, in Section 2.3.4 we explain them in more detail. Later, alternatives to LSTM such as Gated Recurrent Units (GRUs) (Cho et al., 2014) were proposed. Nonetheless, it has been shown that both approaches have similar performance (Chung et al., 2014).

2.3.4 Long Short-Term Memory

LSTM is an improvement over the general RNN architecture. The particularity of LSTM is that it has two inputs ($\mathbf{h}_{t-1}$ and $\mathbf{c}_{t-1}$) and two outputs ($\mathbf{h}_t$ and $\mathbf{c}_t$), where the signal $\mathbf{c}_t$ contains the long-distance information. As presented in Figure 2.6, $\mathbf{c}_t$ encodes the signal $\mathbf{c}_{t-1}$ with minor updates. This causes the value of $\mathbf{c}_t$ (at the iteration t) to retain information from the previous steps.

The process that is executed in an LSTM cell in Figure 2.6 can be broadly described via three main steps:

1. Forget step: In this step, it is decided whether the information of the memory cell $\mathbf{c}_{t-1}$ should be kept or forgotten. This is measured by the forget gate $\mathbf{f}_t$ as

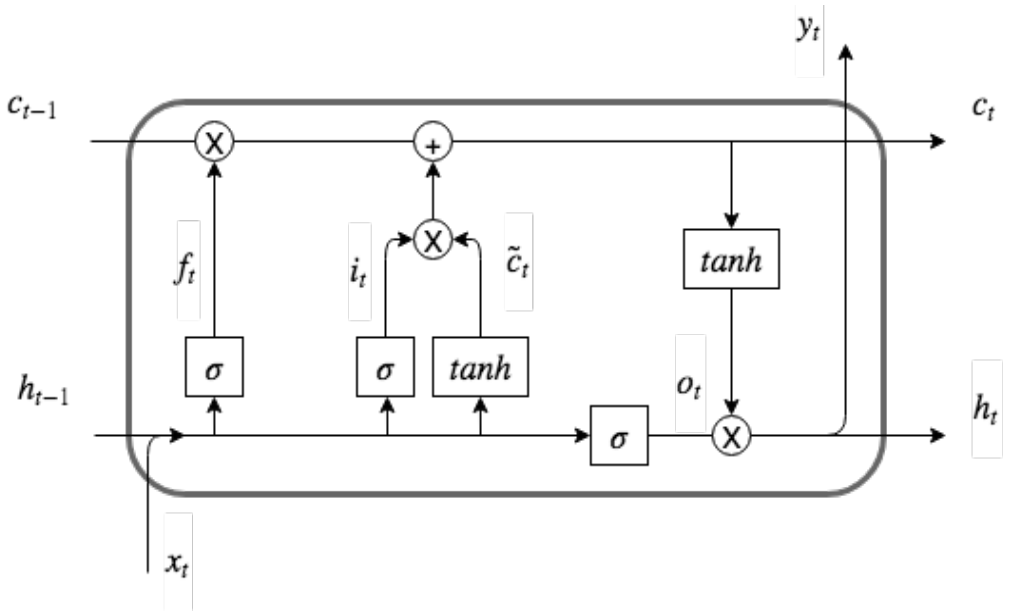

Figure 2.6: Diagram of LSTM.

in Equation (2.20).

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1}). \tag{2.20}$$

2. Update step: In this step, it is decided what should be stored in the memory state. In this step the candidate values $\tilde{\mathbf{c}}_t$ are created and the input gate layer $\mathbf{i}_t$, as Equation (2.21) and Equation (2.22), respectively:

$$\tilde{\mathbf{c}}_t = tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1}) \tag{2.21}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1}). \tag{2.22}$$

3. Output creation step: During the last step, the outputs $\mathbf{c}_t$ and $\mathbf{h}_t$ are produced.

   - $\mathbf{c}_t$ is a combination of how much is forgotten of the previous $\mathbf{c}_{t-1}$ and how much it is updated with new values. It is computed as in Equation (2.23):

$$\mathbf{c}_t = \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \tilde{\mathbf{c}}_t. \tag{2.23}$$

- $\mathbf{h}_t$ combines the cell state $\mathbf{c}_t$ (normalized with tanh function to make the values be in the range $(-1, 1)$) with the output gate $\mathbf{o}_t$ which modulates how much memory content is considered.

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t) \tag{2.24}$$

where $\mathbf{o}_t$ is the output gate. It depends on the current output and the previous hidden state as in Equation (2.25).

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}). \tag{2.25}$$

2.3.5 Encoder-Decoder Architecture

An RNN transforms a sequence of vectors into another target sequence, but both sequences are of the same length. The Encoder-Decoder framework (Cho et al., 2014; Sutskever et al., 2014) is an architecture introduced to solve this problem.

In Figure 2.7 we can find the structure of a basic Encoder-Decoder framework. It consists of two RNNs (an encoder and a decoder) that transform a sequence $X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{T_x})$ into another sequence $Y = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{T_y})$, where the sequence can differ in length.

- Encoder: The encoder converts the sequence $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{T_x})$ into a context vector $\mathbf{c}$ that summarizes the sentence. This vector $\mathbf{c}$ is constructed by considering the hidden vectors $\mathbf{h}_{t_x}^{(s)}$ (where $t_x \in [1, T_x]$) of the encoder (we use $h_i^{(s)}$ to denote the i-th vector of the sequence generated by the encoder and $h_i^{(t)}$ to denote the i-th vector generated by the decoder). When the last element of the sequence $\langle EOS \rangle$ is encoded, the context vector $\mathbf{c}$ is sent to the decoder.

- Decoder: The decoder performs the inverse process of the encoder. Given the context vector $\mathbf{c}$ it produces the vectors $\mathbf{h}_{t_y}^{(t)}$ (until the element $\langle EOS \rangle$ is found) that are decoded as the targets sequence of vectors. $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{T_y})$.

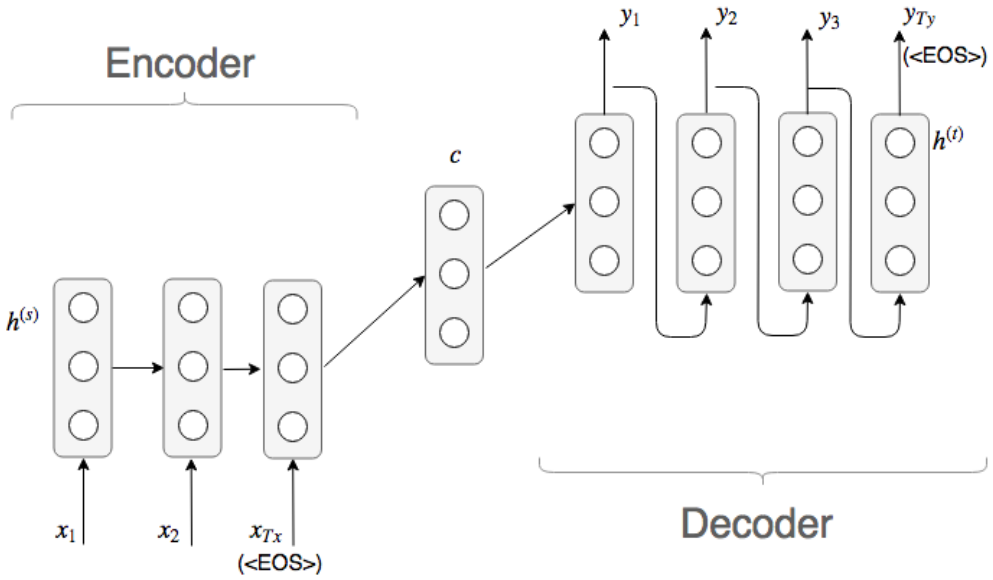

Figure 2.7: Encoder-Decoder model

2.3.6 Attention Model

The problem of the encoder-decoder framework is that it encodes the whole sequence as a single vector. However, especially for the longer sequences, it may lead to an information loss. This has a negative impact on the decoder has it only has access to the vector $\mathbf{c}$ to generate the sequence of sentences.

The Attention Model (Bahdanau et al., 2014) was introduced to solve this problem. Using this mechanism, instead of using a single fixed context vector $\mathbf{c}$ to encode the input sequence, a context vector for each output time step $\mathbf{c}_t$ is created. This helps the decoder to identify the parts in the input sequence that are relevant in order to generate the subsequent words. The vector $\mathbf{c}_t$ is computed as the weighted sum of $h^{(s)}$ as in Equation (2.26):

$$\mathbf{c}_t = \sum_{i=1}^{T_x} \alpha_{ti} \mathbf{h}_i^{(s)} \tag{2.26}$$

where the weights α_{ti} indicate how much attention should y_t pay to each $\mathbf{h}_i^{(s)}$ (i.e. how much y_t is related to the element x_i of the input sequence) and it is computed

as a softmax function as shown in Equation (2.27).

$$\alpha_{ti} = \frac{\exp(a(\mathbf{h}_{t-1}^{(t)}, \mathbf{h}_i^{(s)}))}{\sum_{k=1}^{T_x} \exp(a(\mathbf{h}_{t-1}^{(t)}, \mathbf{h}_k^{(s)}))} \tag{2.27}$$

where the function a is the alignment model. The alignment model we use in this work is computed as in Equation (2.28):

$$a(\mathbf{h}_{t-1}^{(t)}, \mathbf{h}_i^{(s)}) = \mathbf{h}_{t-1}^{(t)} \mathbf{W}_a \mathbf{h}_i^{(s)} \tag{2.28}$$

where $\mathbf{W}_a$ is a weight matrix that is trained jointly with the other parts of the NMT model.

2.4 Translation Performance Evaluation Metrics

Translation performance evaluation metrics are used to evaluate the translation quality of the output of a translation model. These metrics compare the hypothesis (the output of the system) against a reference and determine how similar they are. In this work we are measuring the translated outputs using the following metrics:

- Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) uses matching n-grams between the hypothesis and reference to give a quality score. The score is in the range $[0, 1]$ but often is given as a percentage measure.

  The BLEU score is computed as in Equation (2.29):

  $$BLEU(hyp, ref) = BP \cdot \exp \sum_{n=1}^{N} \frac{1}{N} \log PR_n \tag{2.29}$$

  where N typically has a value of 4, and PR_n is the precision computed as the number of n-grams in common (between the hypothesis and the reference) divided by the number of n-grams in the hypothesis. However, the number of common n-grams should be limited to the maximum number of instances in the reference. For example, for the reference sentence "the airport" the

unigram precision of the hypothesis "the the" would be $\frac{2}{2} = 1$. For this reason the precision is clipped, and is computed as in Equation (2.30):

$$PR_n = \frac{\sum_{ngr \in Ngr_{nn}(hyp)} \min(C_{hyp}(ngr), C_{ref}(ngr))}{\sum_{ngr \in Ngr_{nn}(hyp)} C_{hyp}(ngr)} \tag{2.30}$$

which would cause the unigram precision in the example to be $\frac{1}{2}$.

BP is the brevity penalty defined as $BP = \min(1, e^{1-\frac{|ref|}{|hyp|}})$. It is introduced to penalize those candidates that are shorter than the reference (those longer than reference are already penalized by the precision measure).

- Translation Edit Rate (TER) (Snover et al., 2006) computes the minimum number of edits required to make the hypothesis match the reference. The edits include insertion, deletion or substitution of single words and shifts of word sequences. It is computed as in (2.31):

$$TER = \frac{\#\text{ edits}}{\text{average }\#\text{ of reference words}}. \tag{2.31}$$

- Metric for Evaluation of Translation with Explicit Ordering (METEOR) (Banerjee and Lavie, 2005) provides a score matching words and phrases of the hypothesis and reference. There are three types of match:
  - exact: two unigrams match if they are exactly the same,
  - stem: two unigrams match if they are the same after they have been stemmed,
  - synonymy: two unigrams match if they are synonyms of each other.

  All the words in the candidate must match at most one word in the reference. The number of matches is used to compute the precision $P = \frac{\#\text{match}}{|hyp|}$ and recall

$R = \frac{\#\text{match}}{|ref|}$ to compute the METEOR score as in Equation (2.32):

$$METEOR = \frac{10PR}{R + 9P}(1 - Penalty) \tag{2.32}$$

where $Penalty = 0.5 \left(\frac{\#\text{chunk}}{\#\text{match}}\right)^3$ and #chunk indicates the minimum number of chunks in which all the words in the hypothesis can be grouped. A chunk is the sequence of words (in adjacent positions) in the hypothesis that are also mapped to a sequence of words in the reference.

Unlike the other evaluation metrics we are presenting here, this is a language-dependent evaluation metric as the unigrams are aligned using stemmer and synonym tables.

- Character *n*-gram F-score (CHRF) (Popovic, 2015): It is an evaluation metric that operates at the character level. It computes the F-score as in Equation (2.33):

$$CHRF\beta = (1 + \beta^2)\frac{CHRP \cdot CHRR}{\beta^2 \cdot CHRP + CHRR} \tag{2.33}$$

where:

  - $CHRP$: Percentage of *n*-grams (character level) in the hypothesis which are in the reference (precision).
  - $CHRF$: Percentage of *n*-grams (character level) in the reference which are also present in the hypothesis (recall).
  - β: is a hyperparameter that assigns β times more importance to recall than to precision. In this work we are computing the value where $\beta = 3$, which Popovic (2015) demonstrated to be correlated with human judgment.

2.5 Data Selection

In the context of our work, data-selection methods are techniques that retrieve a subset of sentences from a larger set of sentences on which MT models can be trained. Although using less data for building models may be counter-intuitive, it has several benefits. In some cases, building MT models is restricted by some requirements such as time constraints (in which the model has to be trained or executed in a limited amount of time) or hardware constraints (if the system is deployed in a machine with limited memory or computation power). Other purposes of using data selection methods are:

1. Filtering out noisy sentence pairs (e.g. those that are non-literal translation of each other) that can cause the model to learn incorrect translations.

2. Identifying in-domain data as MT models trained in the same domain as the document to be translated tend to achieve better performance.

The procedure of data-selection algorithms consist of selecting sentences from the candidate pool U and adding them to a selected pool L. Before the execution of the algorithm all sentences are candidate sentences, $U = S$, and the selected pool is empty $|L| = 0$. Data-selection algorithms withdraw one or more sentences from U and add them to L. Note that L contain source-side sentences when the selection is made, but to train the MT models both the source and target side are provided.

This algorithms presented in this section can be classified (based on the work of Eetemadi et al. (2015) and Biçici and Yuret (2011)) according to the following characteristics:

- Transductive algorithms (TA)/Non-transductive Algorithms (NTA): Transductive learning (Vapnik, 1998) approaches consider the test set S_{test} to select the training instances that are relevant, whereas NTA approaches leave out this information instead using the informativeness of sentences for selection. Transductive learning aim to identify the best training sentences to better

classify a given test set. In this context, this algorithms retrieve those parallel sentences that are the most beneficial to train an MT model that translate a given S_{test}. These methods can be sub-divided into two categories:

  - Sentence-wise (SW)/Document-wise (DW): In order to retrieve sentences the test set can be considered as a whole (document-wise) or each sentence individually (sentence-wise). Sentence-wise methods are more fine-grained as they can consider sentence-level characteristics such as word order. However, document-wise methods use information in a more efficient way: by combining the statistical information over all the information in the test set, e.g. n-grams appearing in multiple sentences, they obtain a less sparse and hence statistically more reliable representation for data selection.

- Context-Dependent (CD)/Context-Independent (CI): Context-independent functions retrieve a subset of data without considering the selected pool. Most of these functions come from the quality estimation field, aiming to extract parallel sentences from comparable corpora (bilingual texts that are related but that are not strict translations of each other) by filtering out noisy data and obtaining good quality parallel sentences. In contrast, context-dependent methods consider L for determining which sentence to select next. They pursue to find a balance between exploration (i.e. select sentences to discover new phrases) and exploitation (i.e. estimate more accurately the phrase translation probabilities of the sentences already selected).

The experiments carried out in this thesis are based on an scenario in which the test set is known in advance. Therefore, we are using TAs as they are the only methods benefiting from the information of the test set. However, in order to introduce a general view of data selection algorithms, we provide a summary of main existing methods in all categories (we list in Table 2.1 the data selection techniques presented in this chapter).

	TA/NTA	CD/CI	SW/DW
Length-based Functions	NTA	CI	-
Alignment-based Functions	NTA	CI	-
Language Model-Based Functions	NTA	CI	-
N-gram Coverage	NTA	CD	-
TF-IDF Coverage	NTA	CD	-
DWDS	NTA	CD	-
Log-probability Ratios	NTA	CD	-
Perplexity Ratios	NTA	CD	-
Sentence similarity	TA	CI	SW
Infrequent n-gram Recovery	TA	CD	DW
FDA	TA	CD	DW
ParFDA	TA	CD	DW

Table 2.1: Classification of data-selection algorithms.

2.5.1 Non-Transductive algorithms

Length-based Functions Length-based selection methods aim to select sentences based on the intuition that sentence pairs whose source and target side differ a lot (compared the average length difference) may be noisy and they should be left out.

In the literature we can find different approaches of measuring the difference: Taghipour et al. (2010) propose to use the length difference ($|t| - |s|$) or the proportion ($|t|/|s|$) to remove sentences. Another proposal (Khadivi and Ney, 2005) removes sentences whose length proportion exceeds a certain threshold t (keeping sentences which fulfill the constraint that $\frac{|s|}{|t|} < t$ and $\frac{|t|}{|s|} < t$).

Alignment-based Functions Taghipour et al. (2010) use sentence-alignment entropy to remove noisy data from the training set. In their work, they compute the entropies of the distribution of the alignment links in a sentence.

First, they define the word-alignment probability $P_{al}(w)$. Given a sentence pair

$\langle s, t \rangle$, the probability of a word to be aligned is defined in Equation (2.34):

$$P_{al}(w_s) = \frac{a(w_s)}{\sum_{w_t \in Ngr_1(t)} a(w_t)} \tag{2.34}$$

where w_s is the word $w_s \in Ngr_1(s)$ and $a(w)$ is a function that retrieves how many words in t (or $\langle$NULL$\rangle$ tokens) are aligned to w.

The probabilities computed in Equation (2.34) are used to compute the normalized sentence-alignment entropies as in Equation (2.35):

$$score_{ALIG}(s) = 1 - \frac{-\sum_{i=1}^{|s|} P_{al}(x_i) \log(P_{al}(x_i))}{\log(|s|)}. \tag{2.35}$$

We substract the normalized entropy from 1 so that the value of $score_{ALIG}(s)$ is higher the lower the entropy is. Sentences with high entropy indicate that the alignment of the words is evenly distributed, i.e. the mapping of the words is close to one-to-one mapping.

In contrast, for sentence pairs $\langle s, t \rangle$ with lower entropies, this indicates that words are not evenly aligned, which is an indicator that s and t are inaccurate translations of each other.

For this reason, sentence with lower entropies should be promoted (higher value of $score_{ALIG}$).

Language Model-Based Functions Moore and Lewis (2010) propose to use an in-domain language model LM_I and an out-of-domain language model LM_O to obtain sentences that are closer to the in-domain data. They therefore define the Cross-Entropy Difference (CED) as in Equation (2.36):

$$score_{CED}(s) = H_{LM_I}(s) - H_{LM_O}(s). \tag{2.36}$$

Axelrod et al. (2011) extend Equation (2.36) by using language models in both the source-side and target-side languages, defining the Bilingual Cross-Entropy Dif-

ference (BCED) in (2.37):

$$score_{BCED}(s) = (H_{LM_{I_{src}}}(s) - H_{LM_{O_{src}}}(s)) + (H_{LM_{I_{trg}}}(s) - H_{LM_{O_{trg}}}(s)). \quad (2.37)$$

***N*-gram Coverage** This approach consist of selecting a subset L of sentences aiming to achieve the maximum coverage possible with the minimum amount of sentences. In order to do that, Eck et al. (2005b) select those sentences containing n-grams which have not yet been added to L, rewarding those that are more frequent as defined in Equation (2.38):

$$score_{COV}(s, L) = \frac{\sum_{ngr \notin Ngr_n(L)} C_S(ngr)}{|s|}. \quad (2.38)$$

TF-IDF Coverage The proposal of Eck et al. (2005a) is to retrieve sentences that are the most different to the selected pool L based on TF-IDF distance. Those sentences that differ the most to the selected pool L are the best candidates to be added to L as they are the most informative.

A sentence $s \in U$ is scored as the cosine difference between the TF-IDF vector of the sentence $\mathbf{d}_s$ (each term w is weighted as $tfidf(w, s, S)$) and the vector of the selected pool $\mathbf{d}_L$ (each term weighted as $tfidf(w, L, S)$) as defined in Equation (2.39):

$$score_{TFIDFCOV}(s, L) = dist_{tfidf}(\mathbf{d}_s, \mathbf{d}_L). \quad (2.39)$$

Density Weighted Diversity Sampling (DWDS) (Ambati et al., 2011) selects sentences that contain the most representative n-grams which have not yet been seen in the selected pool. Therefore the score of a sentence s is based on:

- Density $d(s, L)$: This indicates how features are distributed in the selected pool. It is computed as the probability of the n-gram ngr in the candidate pool in proportion to the count of these n-grams ($C_L(ngr)$) in the selected

pool, defined as in Equation (2.40):

$$d(s,L) = \frac{\sum_{ngr \in Ngr_n(s)} P_U(ngr) e^{-C_L(ngr)}}{|Ngr_n(s)|}. \tag{2.40}$$

- Uncertainty $u(s,L)$, the (normalized) number of non-selected n-grams contained in s, defined as in Equation (2.41):

$$u(s,L) = \frac{\sum_{ngr \in Ngr_n(s)} (1 - I_{Ngr_n(L)}(ngr))}{|Ngr_n(s)|} \tag{2.41}$$

  where $I_A(x)$ is the indicator function (which retrieves 1 if the element x is in the set A and 0 otherwise), so $(1 - I_{Ngr_n(L)}(ngr))$ indicates whether ngr is not included in the selected pool L.

Ambati et al. (2011) define $score_{DWDS}(s)$ as the harmonic mean of the density $d(s)$ and the uncertainty $u(s)$ as in Equation (2.42):

$$score_{DWDS}(s,L) = \frac{2d(s,L)u(s,L)}{d(s,L) + u(s,L)} \tag{2.42}$$

Log-probability Ratios Haffari et al. (2009) propose to select sentences that contain n-grams that are common in the candidate pool and rare in the selected pool (the more frequent a phrase is in L, the less important it is). The score of a sentence is measured as in (2.43):

$$score_{LPR}(s,L) = \frac{1}{|Ngr_{nn}(s)|} \sum_{ngr \in Ngr_{nn}(s)} log \frac{P_U(ngr)}{P_L(ngr)}. \tag{2.43}$$

Perplexity Ratio Mandal et al. (2008) propose to select sentences based on the perplexity ratio to select sentences that are novel with respect to an initially selected

pool L. The sentences in the candidate pool U are scored as in Equation (2.44):

$$score_{PPR}(s, L) = \frac{PP_{LM_L}(s)}{PP_{LM_S}(s)}. \tag{2.44}$$

The aim is to select sentences from U that are informative regarding L. Sentences that are rare (high perplexity) in L but common in S (low perplexity) are promoted as they are the most informative. In contrast, those sentences that are rare in both L and S (or only rare in S) are considered outliers and so, the score computed in Equation (2.44) will be low.

2.5.2 Transductive algorithms

Sentence similarity These methods retrieve sentences based on how similar a sentences from S are compared to the test set s_{test}. For every sentence in the test set, the most similar sentences from the training data are retrieved. We present two different approaches of sentence similarity methods:

- Cosine TF-IDF: This method considers the text as a bag-of-words to compute the distance between the test set and the sentences in S. Hildebrand et al. (2005) propose to use cosine between TF-IDF vectors as the distance metric as Equation (2.45):

$$score_{TFIDFsim}(s_{test}, s) = 1/dist_{tfidf}(\mathbf{d}_s, \mathbf{d}_{s_{test}}). \tag{2.45}$$

- Edit Distance: These methods are the most strict as they consider word overlap, order and position. These method measure the sentence similarity using metrics such as Levenshtein distance (Levenshtein, 1966) (which computes the minimum number of insertion, deletions or substitutions of characters that are necessary to transform one sentence into the other) to score the similarities as

in Equation (2.46):

$$score_{levensthein}(s_{test}, s) = 1/dist_{levensthein}(s_{test}, s). \tag{2.46}$$

Note that the edit-distance approach is more strict than cosine TF-IDF as it considers word overlap, word order and position whereas in cosine TF-IDF only the word overlap is weighted (Wang et al., 2014).

Infrequent *n*-gram Recovery Gascó et al. (2012) and Parcheta et al. (2018) propose to extract those sentences containing *n*-grams from the test set that are considered infrequent (so frequent words such as stop words are ignored). In their work, they use this method to retrieve sentences to augment an existing in-domain parallel dataset S_I (so they use $S_I + L$ set to train the models).

A sentence s is scored according to the number of infrequent *n*-grams (both in the in-domain and selected set) shared with the set of sentences S_{test} of the test set. It is computed as in Equation (2.47):

$$score(S_{test}, S_I, s, L) = \sum_{ngr \in Ngr_n(S_{test})} (\min(1, C_s(ngr)) \max(0, t - C_{S_I+L}(ngr))). \tag{2.47}$$

If s does not contain ngr, then the component $\min(1, C_s(s))$ will be 0. t is the threshold of occurrences of an *n*-gram to be considered infrequent. If the number of occurrences is above the threshold t, then ngr is considered to be a frequent *n*-gram and is ignored (the component $\max(0, t - C_S(ngr))$ is 0) and not considered for scoring the sentence.

Feature Decay Algorithms (FDA) (Biçici and Yuret, 2011; Biçici, 2013; Biçici et al., 2015; Biçici and Yuret, 2015) is a method that tries to maximize the variability of *n*-grams in the training set by decreasing their score as they are added to the selected pool.

In order to do that, the n-grams (features) in the test set are assigned a value that is inversely proportional to the number of selected instances, so the more frequent they are in L the more they are penalized. The decay is given by the function in Equation (2.48):

$$decay(ngr, L) = init(ngr)\frac{d^{C_L(ngr)}}{(1 + C_L(ngr))^c} \tag{2.48}$$

where $init(ngr)$ is an initialization function. Biçici and Yuret (2011) propose to use either 1 or the inverted frequency $log(|U|/C_U(ngr))$. The variables d and c are input hyperparameters: the decay factor d is in the range $(0, 1]$ with a default value of 0.5, and the decay exponent c is in the range $[0, \infty)$ with a default value of 0.

The score of a sentence s in the candidate set is the normalized sum of the value of its n-grams as in Equation (2.49):

$$score_{FDA}(S_{test}, s, L) = \frac{\sum_{ngr \in Ngr_n(S_{test})} C_s(ngr)decay(ngr)}{|s|}. \tag{2.49}$$

FDA scores each sentence $s \in U$, and then the sentence with the highest score is removed from U and added to L. This process is repeated every time a sentence is selected until N sentences are added into L. As scoring every sentence each time a sentence is selected is computationally expensive, FDA proceeds as in Algorithm 1.

After initializing the value of the n-grams and the score of the sentences (steps 1 to 8) the sentence-score pairs are stored in a queue Q. This queue is sorted (step 9) by score in descending order. Then, iterativelly the top sentence s of Q is retrieved, the score updated (step 12) and then two scenarios are possible:

- s is selected (step 13; we explain the criteria to select a sentence later), then the values of its n-grams are re-scored. However, the scores of the sentences in Q are not modified, which means that some sentences in Q (those containing the re-scored n-grams) have an outdated score.

- s is not selected (step 18), in which case the pair $\langle s, sc_{new} \rangle$, where sc_{new} is the

Algorithm 1 FDA workflow

$U \leftarrow S$; $L \leftarrow []$; $Q \leftarrow []$;
for all $ngr \in Ngr_n(S_{test})$ **do**
    $ngr \leftarrow init(ngr)$
end for
for all $s \in U$ **do**
    $sc \leftarrow score(s, L)$
    add (s, sc) to Q
end for
$sort(Q)$ by sc in descending order
while $|L| < N$ **do**
    $(s, sc) \leftarrow pop(Q)$
    $sc_{new} \leftarrow score(s, L)$
    if $sc = sc_{new}$ **then**
        add s to L
        for $ngr \in Ngr_n(s)$ **do**
            $ngr \leftarrow decay(ngr)$
        end for
    else
        add (s,sc_{new}) to Q (such that it remains sorted)
    end if
end while

updated score, is inserted in Q in order (so Q remains sorted by score).

Note that in Q, the sentences with updated and outdated scores are mixed. However we know that the score of a sentence can only be lower or equal after being updated. Therefore, if the score of the top sentence s_{best} of Q remains unchanged, it is guaranteed to be the best sentence (i.e. scores of the other sentences are equal or lower than s_{best} even if they were updated). Therefore, as selection criteria, is that the best sentence in Q remains unchanged after being re-scored.

ParFDA ParFDA (Biçici et al., 2014, 2015; Biçici, 2016) tries to parallelize FDA by executing several independent FDA processes on partitions of the training data. Then the resulting selected data is merged into a single dataset. We show the workflow of ParFDA in Figure 2.8.

However, ParFDA is only an approximation of FDA. Dependencies between the sentences are lost. The strength of FDA is that information of the previous selected sentences are considered to choose the next sentence. If the parallel corpus is divided

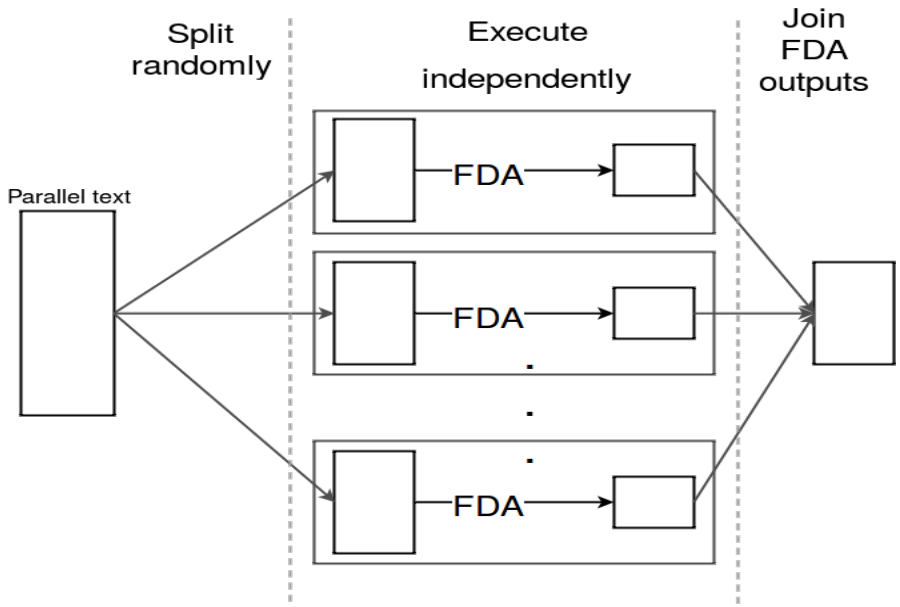

Figure 2.8: ParFDA execution diagram

into different parts, the dependencies between sentences in different parts are lost. Therefore, each FDA process does not have the complete information of the selected pool to decide which sentence to select next. This can hurt performance especially if features are not uniformly distributed over sub-parts of the corpus.

2.6 Conclusions

In this chapter we introduced concepts from NLP, and MT in particular. In addition, we also described the main data-selection algorithms. Although in the experiments we only use those methods classified as transductive, we believe that it is important to have an overview of the main data-selection techniques, their characteristics and how they are organized.

Chapter 3

Transductive Algorithms on Statistical Machine Translation

The scenario we explore in this thesis is how to improve MT models by using less but more suitable data. More specifically, we assume that we only have a set of parallel data available for training a model and a document in the source language (test set) that needs to be translated. We have no other knowledge such as the domain of the sentences, or lexical tags of the words. In fact, although we build German-to-English models here, all the methods described are language-independent, so we could also be unaware of the language pairs themselves.

In this chapter, we demonstrate that using more data to train SMT models is not always the best approach. In fact we show that it is better to use a model trained with fewer sentences but closer to the test set. We show how data-selection algorithms can retrieve a subset of training data that causes an SMT model to obtain better translation performance than a model trained with all data.

As the test set is known, the data-selection algorithms we investigate are only TA (described in Section 2.5.2): TFIDF, INR, and FDA. The purpose of this chapter is to show how these methods can retrieve a subset of training data that can be used to build smaller models that achieve better translation quality for the given test set. The contributions of this chapter are summarized as follows:

- We perform a comparison of SMT models using different amounts of data (Section 3.2).
- We explore the increase in coverage of the test set when using different TAs (Section 3.4).
- We compare SMT models built with the full set of training sentences and a subset retrieved by TA (Section 3.4).

In addition, the outcomes of this chapter are used for planning other experiments involving the construction of SMT models in the following chapter of this thesis.

3.1 Experiment Settings

3.1.1 Data

The experiments performed use the same data as in the German-to-English experiment described in Biçici et al. (2015). The datasets are the following:

- *Training set*: The training data provided in WMT 2015 (Bojar et al., 2015).
- *Development set*: 5K randomly sampled sentences from development sets provided in the WMT Translation Tasks from the years 2010 to 2014.
- *Test set*: We use two different test set to evaluate the models:
  - BIO test: The Cochrane[1] dataset provided in the WMT 2017 biomedical translation shared task (Yepes et al., 2017).
  - NEWS test: The test document provided in the WMT 2015 German-to-English translation task (Bojar et al., 2015).

In Table 3.1 we show the statistics of the datasets mentioned above.

[1] `http://www.himl.eu/test-sets`

	$\|S\|$	$\|W\|$		$\|V\|$	
		DE	EN	DE	EN
Training set	4.48M	110M	116M	2M	971K
Development set	5000	127K	129K	23K	16K
NEWS Test set	2169	44K	46.8K	9.9K	7,8K
BIO Test set	411	8.6K	8.5K	2K	1.6K

Table 3.1: Statistics of the data sets. $|S|$ is the number of sentences, $|W|$ the number of words, and $|V|$ the size of the vocabulary.

3.1.2 SMT Settings

The SMT experiments executed in this thesis are German-to-English SMT models trained with the complete set or subsets of the *Training set*. For tuning, we use the *Development set*.

The SMT models use the same 8-gram Language Model (LM) built using the target-side of the complete *Training set* via the KenLM toolkit (Heafield, 2011) using Kneser-Ney smoothing (Kneser and Ney, 1995). The models are evaluated using BIO and NEWS test sets.

We train SMT models using the Moses toolkit (Koehn et al., 2007) with the standard features (described in Section 2.2.3) and using GIZA++ (Och and Ney, 2003) for word alignment. We also perform tuning using MERT (Och, 2003) with the reported scores based on the average of four MERT runs.

For evaluating the performance of the models we provide the scores of evaluation metrics presented in Section 2.4: BLEU, TER, METEOR, and CHRF3.

3.2 SMT Models using Subsets of Data Sampled Randomly

Before carrying out experiments using TA, we are interested in exploring the performance of SMT models using different sizes of random subsets of data. In particular, we present in Table 3.2 the performance of models built with 100K, 200K, 500K, 1M, 2M and 4.5M (the full set) sentence pairs. The outcomes of Table 3.2 are also

presented in a graphical format in Figure 3.1 and Figure 3.2.

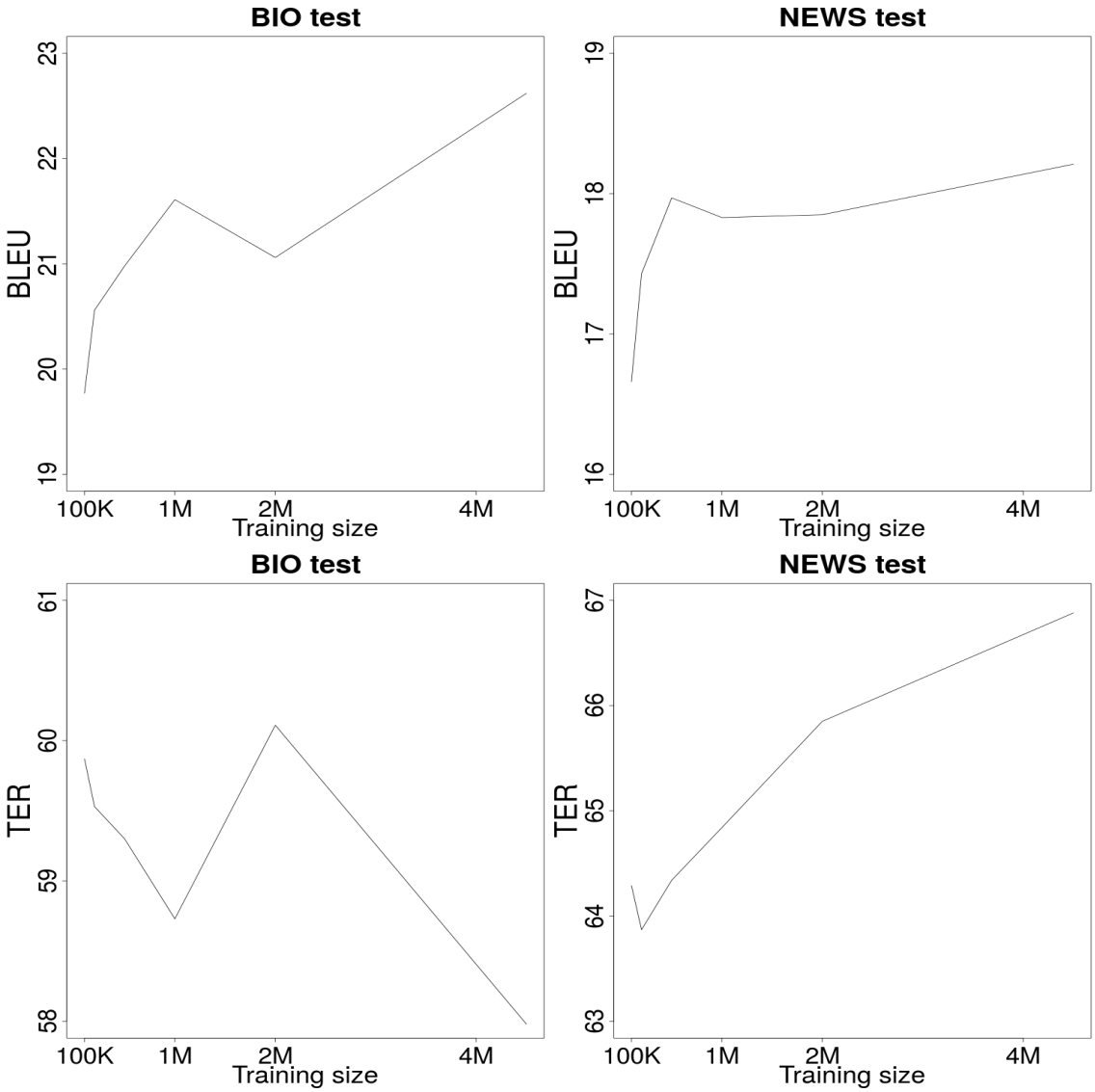

Figure 3.1: Results (BLEU and TER) of SMT models trained in different sizes of data.

The results in Table 3.2, or Figures 3.1 and 3.2 (note that for the plot of TER metric lower values indicate better performance), reveal that although the performance tends to increase when using more data, that is not always the case. For example, in the figures, we can see how for the BIO test set (plots on the left) the performance decreases when moving from 1M sentences to 2M, or with the NEWS test set (plots on the right) the same thing happens when increasing from 500K to 1M sentences.

Note also that the plots presented in Figures 3.1 and 3.2 show that the evaluation

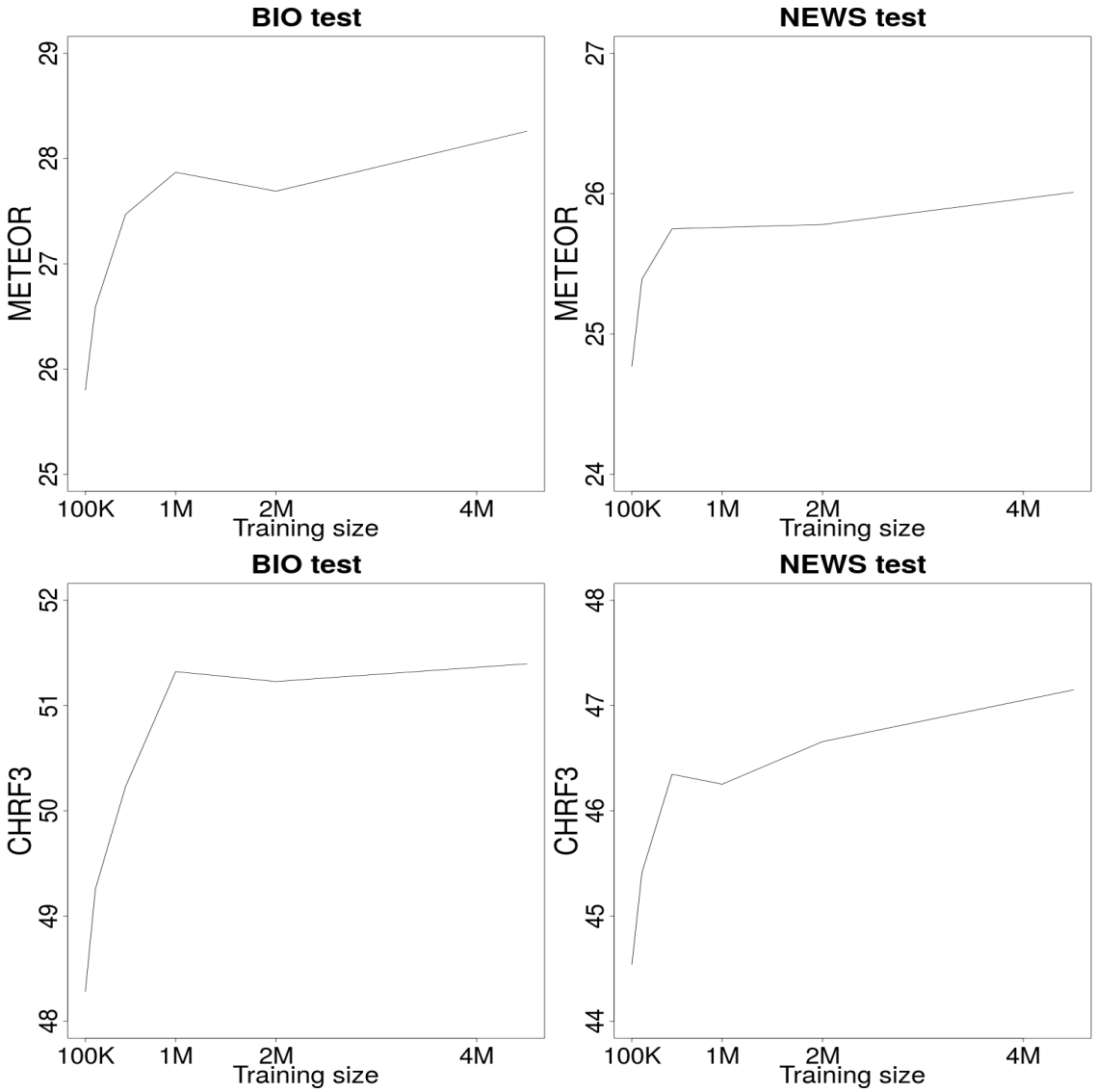

Figure 3.2: Results (METEOR and CHRF3) of SMT models trained in different sizes of data.

metrics tend to be highly correlated (the only exception is TER when evaluated on the NEWS test set). In following graphical representation of the results, we will only present BLEU scores.

		BIO	NEWS
100K lines	BLEU	19.77	16.66
	TER	59.87	64.29
	METEOR	25.80	24.77
	CHRF3	48.28	44.54
200K lines	BLEU	20.56	17.43
	TER	59.53	63.87
	METEOR	26.59	25.39
	CHRF3	49.26	45.41
500K lines	BLEU	20.98	17.97
	TER	59.30	64.34
	METEOR	27.47	25.75
	CHRF3	50.23	46.35
1M lines	BLEU	21.61	17.83
	TER	58.73	64.84
	METEOR	27.87	25.76
	CHRF3	51.32	46.25
2M lines	BLEU	21.06	17.85
	TER	60.11	65.85
	METEOR	27.69	25.78
	CHRF3	51.23	46.66
ALL	BLEU	22.62	18.21
	TER	57.98	66.88
	METEOR	28.26	26.01
	CHRF3	51.40	47.15

Table 3.2: Results of SMT models built with different sizes of (random) data.

3.3 Exploration of Transductive Methods in SMT

Our goal is to select sentences to build models that are smaller yet produce higher quality translation for the test set than larger models trained with all data. For this reason, the data-selection algorithms we choose to explore are those categorized as transductive algorithms (TA in Table 2.1). In particular, we investigate TFIDF, INR and FDA methods using the following configurations:

- TFIDF: This selection method is the only context-independent method explored. In addition, it is a sentence-wise method, so each sentence of the test set is used independently to select training sentence-pairs, which means that there may be overlap between the selected subset for each sentence of the test set. Hildebrand et al. (2005) note this problem as something to be considered.

In their experiments, both models (built with and without duplicates) perform similarly, but models built with data whose duplicated sentences have been kept achieve slightly better results. For this reason, in our experiments, we also keep duplicated sentences.

The number of duplicate sentences retrieved by TFIDF depends on several factors: (i) if sentences in the test set are similar, then the number of duplicates will be higher (as the extracted subsets will be similar to each other); (ii) the larger the retrieved subsets are, the less likely it is that a sentence appears only once across the subsets (Table 3.3 indicates the proportion of unique sentences found for different-sized subsets); and (iii) the larger test set is the more likely it is to extract the same sentences.

Number of sentences	Percentage of unique sentences	
	NEWS	BIO
100K	73%	68%
200K	68%	66%
500K	61%	61%
1M	55%	56%
2M	48%	49%

Table 3.3: Percentage of unique sentences in the data retrieved by TFIDF method.

- INR: The purpose of the INR method is to select data (that is close to the test set S_{test}) by augmenting an initial in-domain data set S_I. However, as mentioned before, we assume to be in the simplest scenario with no knowledge of the domain. For this reason we are not augmenting any set but building it from scratch. In addition, in order to compute how frequent an n-gram is we use the training set S. Therefore the Equation (2.47) of INR is computed as

in Equation (3.1):

$$score(S_{test}, S, s, L) = \sum_{ngr \in Ngr_3(S_{test})} (\min(1, C_s(ngr)) \max(0, t - C_{S+L}(ngr))) \tag{3.1}$$

using an n-gram order of three (we use the default order of FDA to make it more comparable). In order to set a value of t in Equation (3.1), in the original work of Parcheta et al. (2018) they explore different values (between 10 and 40), for the threshold t, but do not provide any default configuration.

For our experiments we perform several executions of INR, starting with a value of $t = 10$ and multiplying by two each run. We keep increasing the value until the execution time exceeds 48 hours. Note that the higher t is, the larger amount of n-grams are considered infrequent (the criteria is less strict), and so more sentences are retrieved. In Table 3.4 we present the number of sentences retrieved by each execution. We see that more sentences are retrieved with larger the values of t, but they do not exceed 300K. As in this work we are comparing SMT models using training sets of sizes up to 2M sentences, we choose the value of t that causes the model to retrieve the highest amount of sentences, i.e. $t = 640$ for BIO test set (275K sentences retrieved) and $t = 80$ for NEWS test set (230K sentences retrieved).

threshold	Number of sentences selected	
	NEWS	BIO
10	27396	4568
20	57092	9644
40	116550	19415
80	229913	38800
160	-	75263
320	-	142779
640	-	274678

Table 3.4: Number of sentences retrieved by INR using different values of threshold t.

- FDA: The configuration of FDA in our experiments uses the default parameters in Equation (2.49), i.e. $d = 0.5$, $c = 0$ with 3-grams as features.

  As ParFDA is an approximation of FDA we do not carry out experiments using this method. However, we propose as future work to repeat the same experiments as here exploring the performance of ParFDA when the data are partitioned into different numbers of parts.

We use the BIO and NEWS test sets to retrieve data closer to these sets. We investigate different sizes of TA-retrieved data to build SMT models and compare their performance to the model trained with the complete training data (BASE).

3.4 Results

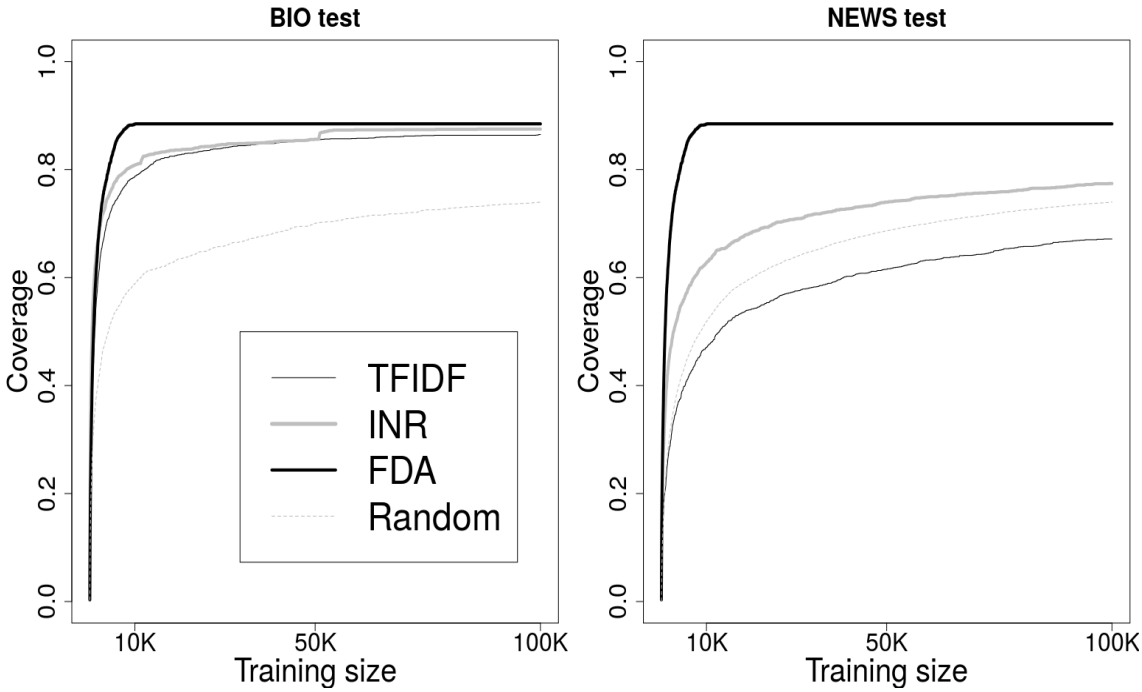

Figure 3.3: Coverage of Transductive methods (up to 100K sentences)

Before inspecting the performance of the models, we present in Figure 3.3 the percentage of the vocabulary of the test set that is covered by each method when different amounts of sentences are selected. The plot considers only the first 100K sentences as this is enough to achieve a plateau. The figure also includes the coverage

of 100K randomly sampled sentences (dotted line), so that we can observe how using TA causes the coverage to increase more rapidly. The exception to this is the TFIDF method for the NEWS test where (as it contain duplicated sentences) the coverage increases more slowly than with random sentences.

Comparing each method individually, we observe that FDA has the steepest curve. This demonstrates the benefits of promoting unseen words over the repetition of words. In contrast, the curves for the INR method increase more slowly and, in the case of the left plot of Figure 3.3, the coverage after selecting 100K sentence is smaller than for FDA.

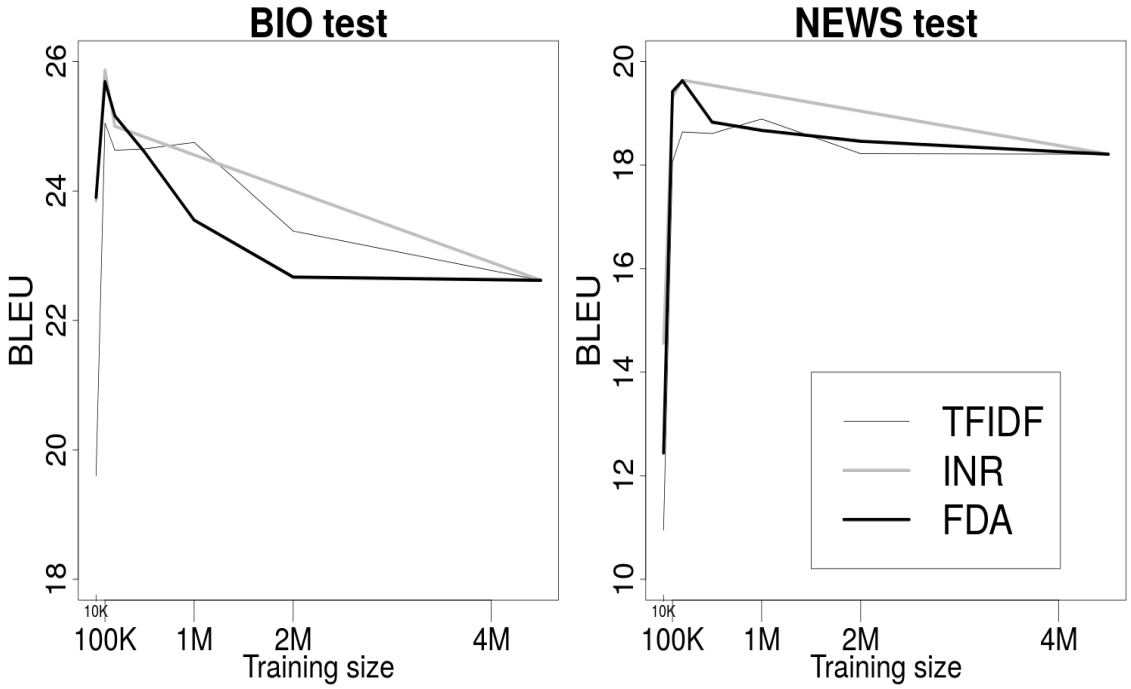

Figure 3.4: Results of TA with different sizes of data for BIO (left) and NEWS (right) test sets.

The performance of the model trained with the selected data is presented in Table 3.5 and the plots of the BLEU scores are displayed in Figure 3.4.

Table 3.5 includes the column *BASE* with the scores of the model trained with the complete training set. We have marked in bold the scores that outperform the baseline and computed the statistical significance (marked with an asterisk) with multeval (Clark et al., 2011) for BLEU, TER and METEOR when compared to the baseline at level p=0.01 using Bootstrap Resampling (Koehn, 2004).

		BASE	TFIDF	INR	FDA
BIO					
100K lines	BLEU	22.62	**25.05***	**25.87***	**25.69***
	TER	57.98	**53.91***	**53.85***	**53.53***
	METEOR	28.26	**30.18***	**30.67***	**30.84***
	CHRF3	51.40	**53.60***	**54.22***	**54.30***
200K lines	BLEU	22.62	**24.63***	**25.00***	**25.16***
	TER	57.98	**54.56***	**55.50***	**54.84***
	METEOR	28.26	**30.22***	**30.12***	**30.36***
	CHRF3	51.40	**53.40***	**53.53***	**53.92***
500K lines	BLEU	22.62	**24.65***	-	**24.60***
	TER	57.98	**55.21***	-	**56.17***
	METEOR	28.26	**30.21***	-	**29.73***
	CHRF3	51.40	**53.58***	-	**53.17***
1M lines	BLEU	22.62	**24.75***	-	**23.55***
	TER	57.98	**55.40***	-	**57.50***
	METEOR	28.26	**30.34***	-	**28.91***
	CHRF3	51.40	**53.68***	-	**52.48***
2M lines	BLEU	22.62	**23.38**	-	**22.67**
	TER	57.98	**56.97***	-	59.05
	METEOR	28.26	**29.27***	-	**28.74**
	CHRF3	51.40	**52.60**	-	**52.53**
NEWS					
100K lines	BLEU	18.21	18.06	**19.31***	**19.42***
	TER	66.88	**63.30**	**63.03***	**62.26***
	METEOR	26.01	25.85	**26.68***	**26.76***
	CHRF3	47.15	45.91	**47.22***	47.06
200K lines	BLEU	18.21	**18.64***	**19.64***	**19.63***
	TER	66.88	**62.38***	**63.14***	**63.27***
	METEOR	26.01	**26.60***	**27.12***	**27.08***
	CHRF3	47.15	46.70	**48.09***	**48.01***
500K lines	BLEU	18.21	**18.61***	-	**18.83***
	TER	66.88	**62.96***	-	**64.44***
	METEOR	26.01	**26.68***	-	**26.58***
	CHRF3	47.15	46.92	-	**47.68**
1M lines	BLEU	18.21	**18.89***	-	**18.67***
	TER	66.88	**62.95***	-	**65.25***
	METEOR	26.01	**26.81***	-	**26.48***
	CHRF3	47.15	**47.40**	-	**47.61***
2M lines	BLEU	18.21	**18.22**	-	**18.46***
	TER	66.88	**64.56***	-	**66.05***
	METEOR	26.01	**26.37***	-	**26.25***
	CHRF3	47.15	**47.24**	-	**47.35***

Table 3.5: SMT models built with different sizes of selected sentences. The results in bold indicate an improvement over BASE. The asterisk means the improvement is statistically significant at p=0.01.

	Sentence	dies sind die bekannten Nebenwirkungen dieser Medikamente.
	Reference	these are recognized side effects of these drugs.
	BASE	these are the pain of these medicines.
100K lines	TFIDF	**this** are the **side effects** of **this medications**.
	INR	these are the **well-known side effects** of **that medication**.
	FDA	these are the **well-known side effects** of **that medication**.
2M lines	TFIDF	these are the **known side-effects , such treatment**.
	INR	-
	FDA	these are the **known side-effects , such treatment**.

Table 3.6: Comparison of outputs produced by SMT models built with TA-selected sentences.

In the table, we discover that most of the models built with TA-selected sentences outperform the baseline. This also implies that the performance is better than those models built with random sentences described in Section 3.2.

Figure 3.4 shows how the performance of the model changes as more TA-selected data are used for training. The plot covers the results when selecting data ranging from 10K sentences (as this is when the plateau of coverage is achieved according to Figure 3.3) to the complete training set. In the figure, we can see that although the maximum coverage is achieved with around 10K sentences, the models trained with this number of sentences achieve the worst results, which are even lower than the BASE. Despite that, when adding a few more sentences, the maximum performance is achieved, i.e. using 100K sentences for the BIO test set or 200K sentences for the NEWS test set. Then, the inclusion of more data causes the performance to decrease. In contrast to random-selected data, we observe that when using data from TA, the reduction is constant and there is no oscillation, so it seems unlikely that adding more data will cause the model to improve further.

In Table 3.6 we include an example of a sentence translated by the model built with all training data (*BASE* row) and the models trained with selected data. In the *100K* subtable we present the output of those models built with the smallest set of TA-retrieved data we have explored, and in subtable *2M* the output of those models built with the largest subset. We have also marked in bold the part of the sentences

that differ with the translation produced by the *BASE* model. Table 3.6 shows how the *BASE* model incorrectly translates the phrase "bekannten Nebenwirkungen" ("recognized side effects") as "the pain", whereas models trained with TA-selected data provide more accurate translations such as "side effects", "well-known side effects" or "known side-effects".

The table also shows how smaller models can translate the sentence better than those trained with larger sizes. For example, the translation provided by models built with 2M TA-selected sentence-pairs omits the phrase "of these" in the translation which causes the sentences to be grammatically incorrect. In contrast, smaller models produce "of this" and "of that" causing the sentence to be closer to the reference. Another example is the word "Medikamente", which is translated as "treatment" (in the *2M* subtable) whereas smaller models produce "medication" or "medications" which are more similar to the word "drugs" in the reference.

3.5 Conclusion and Future Work

In conclusion, we have shown that SMT models trained with more data are not necessarily better. In fact, a model trained with a selected subset of sentences can perform better than a model trained with all data available. This demonstrates that blindly adding data is not always a solution to improve MT models, as it may be possible to find a subset of training data that causes the SMT model to achieve better translation quality. The TAs explored in this chapter have shown themselves to be good methods to find such subsets.

More specifically, we have presented a comparison of the performance of models built with data selected using three TA methods: TFIDF, INR, and FDA. We have shown that context-independent methods such as TFIDF do not perform as well as context-dependent methods (i.e. INR and FDA), as the latter considers more information (from the selected pool) to condition the decision to select new sentences. We selected 100K, 200K and 500K sentences as training data, and showed

that just a small subset of sentences is enough to obtain improved models.

In the future, we want to investigate other configurations of the same methods explored in this chapter, e.g. removing the duplicates when using TFIDF, using smaller values of t for INR or using alternatives such as ParFDA. More recently, the NMT paradigm has exhibited better results when a larger amount of data are available. In order to investigate whether the techniques presented here are also applicable in NMT, we dedicate the next chapter to explore the performance of NMT models trained using TA-selected data.

Chapter 4

Transductive Algorithms on Neural Machine Translation

As explored in the previous chapter, SMT models built with the data retrieved from TA can outperform a model built with the full training set. We want to explore whether it is also applicable for NMT to create improved models. Accordingly, in this chapter we want to address research question **RQ1: How can we tailor data-selection algorithms to be most effective in combination with NMT?**

NMT approaches exhibit better performance than SMT when larger sizes of data are available for training (Koehn and Knowles, 2017). The incorporation of additional training data tend to cause the performance of NMT models to increase (this is further explored in Section 4.3). However, this is only true if the data added are good-quality in-domain sentences. For this reason, we want to explore the impact on the models when using the subset of data retrieved by TA that, although being smaller in size, is closer to the test set. The main contributions of this chapter, based on Poncelas et al. (2018b) and Poncelas et al. (2019b), are the following:

- We provide a summary of main techniques used to build NMT-adapted NMT models (Section 4.1).
- We perform a comparison of NMT models using complete words or sub-words as vocabulary (Section 4.2.2).

- We perform a comparison of NMT models using different sizes of data (Section 4.3).
- We compare models built with randomly-selected data with TA-selected data (Section 4.5.2).
- We analyze models fine-tuned with subsets of training data (Section 4.5.1).

The experiments carried out in this chapter will provide insights on the best configurations so that improvements are achieved using TA-selected data. Consequently, the experiments described in the following chapters will be based on the outcomes of this chapter.

4.1 Domain Adaptation in NMT

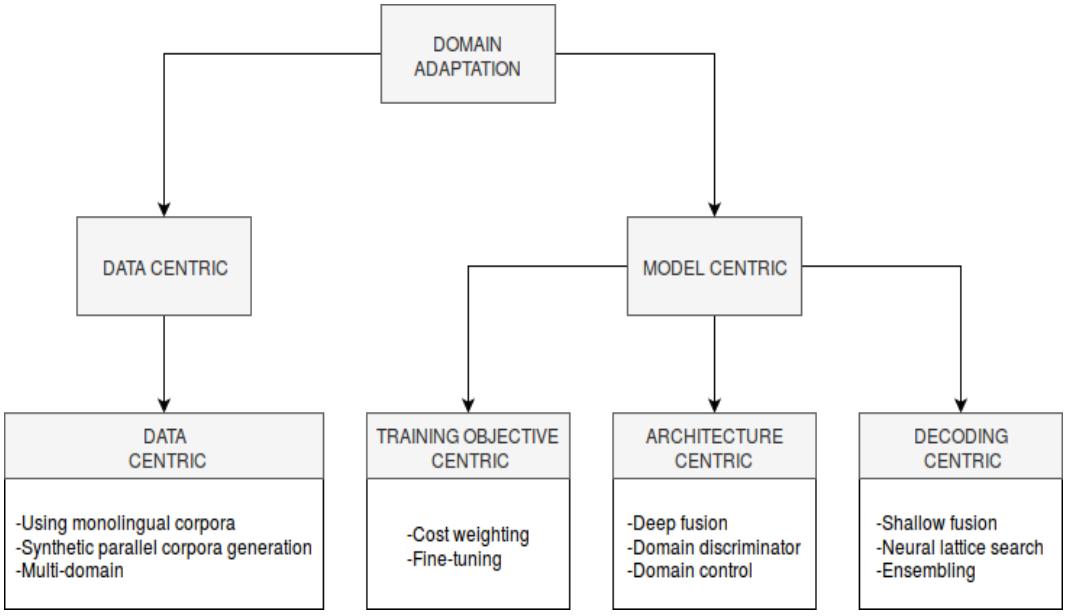

Figure 4.1: Overview of domain adaptation for NMT.

This section provides a general overview of adaptation techniques for NMT which we wil use to design experiments to apply the data selected by TA. According to Chu and Wang (2018), adaptation procedures can be structured into two main groups (see Figure 4.1):

- Data-Centric: Techniques which involve augmenting or modifying the training data. Models can be adapted to a domain by adding monolingual data (*using monolingual corpora*). For example, Sennrich et al. (2016b) add target-language monolingual sentences to the training data (having a ⟨NULL⟩ token in the source-side) that the fluency of the generated sentences is improved. A similar approach consists of creating sentences artificially (*synthetic parallel corpora generation*: Sennrich et al. (2016b)). These sentences can be obtained by using an MT model that translates monolingual sentences in the target language into the source language. Other data-centric approaches involve a modification of the data by appending a tag with its domain to each sentence (*multi-domain*: Chu et al. (2017)).

- Model-Centric: These are techniques which involve modifying the structure or the procedure in which the model is trained. They can be sub-divided into three groups:

  - Training Objective-Centric: These are techniques that alter the cost functions or the procedure of the training. A popular method in this category is *fine-tuning*. We discuss this method and its variants in Section 4.1.1. Another training objective-centric method is *cost weighting* (Wang et al., 2017, 2018) which consists of modifying the NMT cost function so that in-domain sentences have a higher weight (e.g. measured using cross-entropy difference).

  - Architecture-Centric: These approaches consisting of modifying the structure of the NMT model such as concatenating the hidden states of the decoder and an LM trained in the target-language (*deep fusion*: Gülçehre et al. (2015)); adding a discriminator aiming to predict the domain based on the hidden states of the encoder, which forces the encoder to preserve domain-related information (*domain discriminator*: Britz et al. (2017)); or increasing the size of the word embeddings so they include the domain

of the word (*domain control*: Kobus et al. (2017)).

- Decoding-Centric: These are techniques that improve the decoder of the models, such as using an LM trained with target-side language (*shallow fusion*: Gülçehre et al. (2015)), word lattices generated by SMT (*neural lattice search*: Khayrallah et al. (2017)) or ensembling the adapted model with the general-domain model to avoid overfitting (*ensembling*: Freitag and Al-Onaizan (2016)).

4.1.1 Fine-tuning

A method to adapt a general-domain NMT model is the *fine-tuning* technique (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016). This consists of using a small set of in-domain data to train the last epochs of a bigger model trained with more general-domain sentences. For this reason, in this work, we also refer to fine-tuning as adaptation of a model.

There are several variations of applying this fine-tuning. For example, Chu et al. (2017) perform fine-tuning using a dataset tagged with the domain they want to adapt to (i.e. combination of the fine-tuning and the *multi-domain* techniques). This was also used by Poncelas et al. (2019d) to adapt NMT models to the Wikipedia captions domain in Basque and Irish languages, where it was shown that the inclusion of domain-tags in the dataset used for fine-tuning caused the performance to increase.

A related approach is the *transfer learning* method used by Zoph et al. (2016) to build efficient models on low-resource languages. In their work, they train a model on a high-resource language and then train the last epochs with data of the low-resource language of interest.

Another variation is the *gradual fine-tuning* technique proposed by van der Wees et al. (2017). They train the model with a different-sized data set in each epoch. The amount of training sentences is decreased gradually, keeping those that are more in-domain (according to CED weights). The size of the subset of a training data S

at each epoch e is defined as Equation (4.1):

$$n(e) = \alpha \cdot |S| \cdot \beta^{\lfloor (e-1)/\eta \rfloor} \tag{4.1}$$

where α is the relative start size (i.e. the fraction of training data used for the first epochs), β is the retention rate (i.e. fraction of training data kept in the new selection), and η is the number of epochs for which the same subset is used.

A usage of fine-tuning that considers the test set is proposed by Li et al. (2018). In their work, they build one adapted model for each test. They use the set of sentences that is the most similar to each sentence of the test set, retrieved by using three string similarity measures: (i) Levenshtein distance (Levenshtein, 1966); (ii) cosine similarity of the average vector of the word embeddings; and (iii) the cosine similarity between hidden states of the encoder in NMT. This approach is the closest to our work as the test set is considered to retrieve sentences. The main difference is that their adaptations by fine-tuning are sentence-wise (one model for each sentence) whereas in this thesis the adaptations are made on a document-wise basis (one model for each test set). Performing a sentence-wise adaptation has the benefit of being more fine-grained but it also has disadvantages: (i) the total computational cost is higher as the data selection and fine-tuning process are executed multiple times (as many as sentences there are in the test set); (ii) the usage of the data is less efficient as the adaptations are performed independently and the same sentence can be extracted and used for fine-tuning multiple times (to adapt different models); and (iii) translating sentences in the same test set using different models increases the risk of generating inconsistencies through the entire document (e.g. same term can be translated differently in two sentences).

4.2 Experiment Settings

4.2.1 NMT Settings

In this work we construct NMT systems using *OpenNMT-py*, the Pytorch port[1] of OpenNMT (Klein et al., 2017) to train the models. According to the creators of OpenNMT[2] a good baseline for German-to-English WMT 2015 data is the one built with default parameters:

- Vocabulary size of 50000 for each language.
- 2-layer LSTM with 500 hidden units.
- SGD with a learning rate of 1, which decays at 0.5 rate after the 8th epoch.
- Attention model computed as described in Equation (2.27).
- The words in the output that are not in the vocabulary are replaced by the word in the source with the highest attention.

The training process in NMT involves splitting the training data into batches (we use 64 sentences per batch) and using them to update the weights of the models. The process of using every batch to train the model is called an "epoch".

Generally, models are trained for several epochs, until convergence is achieved (the accuracy of the model does not increase when evaluated using a development set). If the model is further trained after convergence it may lead to overfitting. In our experiments, all models are trained for 13 epochs. This is the default number of epochs in the settings of OpenNMT-py, but we also find that the model (trained with all data) actually converges around epoch 13. Figure 4.2 shows the accuracy and perplexity of the model when trained until the 16th epoch. In the figure, we observe that after the 13th epoch the accuracy and perplexity remain approximately the same.

[1] `https://github.com/OpenNMT/OpenNMT-py`
[2] `http://opennmt.net/Models/`

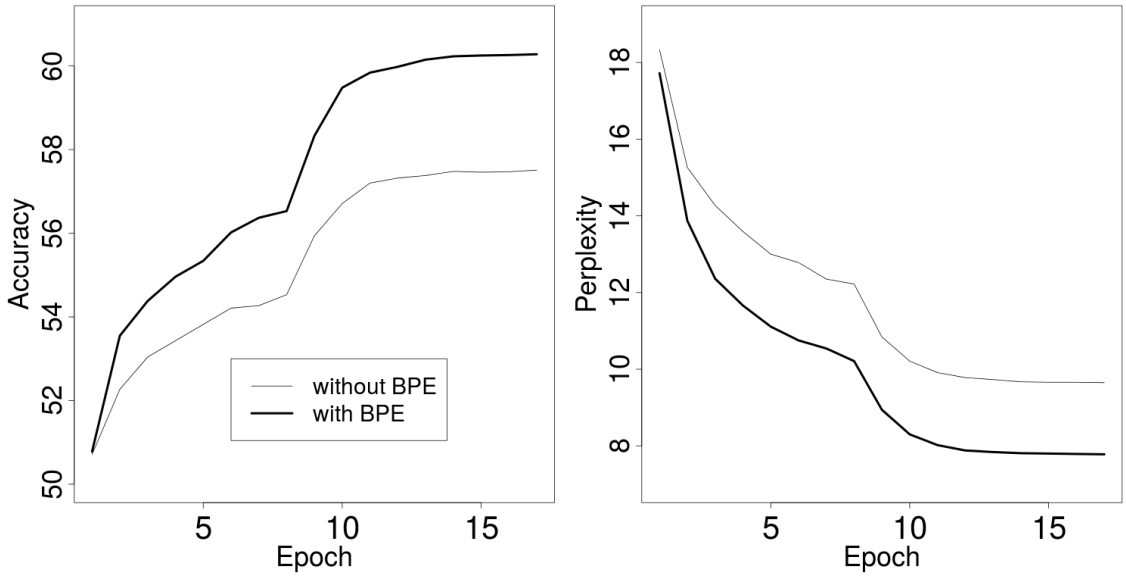

Figure 4.2: Accuracy and Perplexity of the NMT model in each epoch.

In addition, we evaluate the model when trained with different epochs. The results up to the 16th epoch can be seen in Figure 4.3. As we can see, the models are stable around the 13th epoch.

4.2.2 The Use of BPE in NMT

A problem with NMT is that the vocabulary size needs to be established at training time and it remains fixed. At decoding time, the words of the test set that were omitted from the vocabulary are copied from the source (which is the approach we follow) or an UNK token is generated.

In order to solve this problem, Sennrich et al. (2016c) propose to use Byte Pair Encoding (BPE), a technique consisting of segmenting words into sub-word units. The intuition behind BPE is that unknown words may still be translated if they are split into smaller sub-units. For example, the word “daylight” may not be included in the vocabulary of the NMT system (if it is infrequent), but if it is split into “day” and “light”, these sub-words could be part of the vocabulary and be considered in the translation process. In BPE, instead of splitting words into morphemes (Passban, 2017), the division is made by measuring what sequence of characters occur more

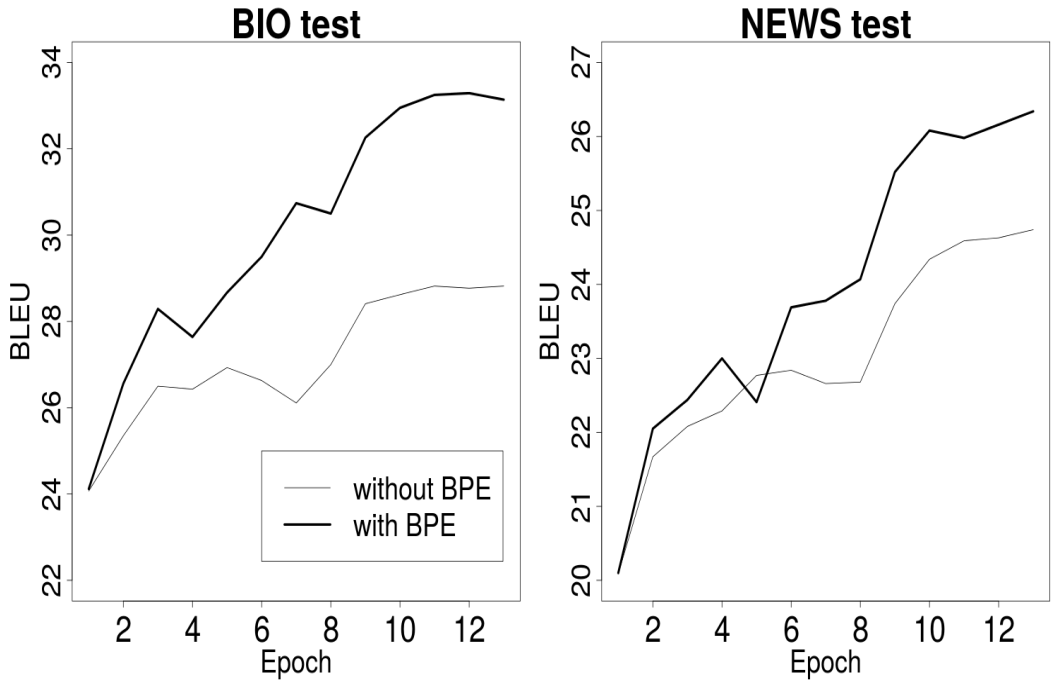

Figure 4.3: Evaluation metrics of the NMT models by epoch.

frequently (it is language-independent), so a sub-word does not necessarily have a meaning.

Applying BPE requires to learn how to split the words first. In order to do this, we can use monolingual texts, in which case it proceeds as follows: The vocabulary, consisting of character n-grams, is initialized with individual characters (character unigrams) in the text, so the words in the data are represented as a sequence of characters. Then, iteratively:

1. The most frequent pair of sequential character n-grams $\langle a, b \rangle$ is identified.

2. All occurrences of $\langle a, b \rangle$ are merged into ab and the new symbol ab is added to the vocabulary.

3. Repeat this process until the maximum number of merge operations is reached.

In our experiments, we use BPE trained jointly on both sides of the training data using 89,500 merge operations (the configuration used in the original work of Sennrich et al. (2016c)). Note that in Table 4.1 we also present the results of the models (built with all data executed for 13 epochs) trained with data without apply-

	without BPE	10,000 operations	89,500 operations
BIO			
BLEU	28.82	**31.49***	**33.14****
TER	50.24	**48.17***	**46.79****
METEOR	29.49	**33.59***	**34.57****
CHRF3	52.94	**57.53***	**59.08****
NEWS			
BLEU	24.74	24.58	**26.34****
TER	55.25	55.59	**54.41****
METEOR	27.98	**29.10***	**30.09****
CHRF3	48.95	**50.31***	**51.71****

Table 4.1: The model using different different merge operations. The results in bold indicate an improvement over the baseline. An asterisk shows that the improvement is statistically significant at p=0.01 when compared to *without BPE*, and double asterisks when compared to both *without BPE* and *10,000 operations*.

ing BPE (*without BPE* column), using 10,000 and 89,500 merge operations (*10,000 operations* and *89500 operations* columns, respectively). We have marked in bold those scores achieving better results than the model trained with data without having applied BPE. The scores tend to be better with data that has had BPE applied. Most of the improvements are statistically significant at level p=0.01 (marked with an asterisk), and in the case of the *89,500 operations* column we have marked with two asterisks those improvements that are statistically significant compared both to models with and without BPE using 10000 operations.

Note also that splitting the words into too many subwords (using a low number of merge operations) can also hurt performance: the model built with data after applying 10,000 operations (smaller sub-words) performs worse than the model without BPE according to BLEU and TER metrics for the NEWS test.

In the remainder of this work, we denote *model with BPE* to those models that have been built using data that has been preprocessed using BPE.

4.3 NMT with Different Sizes of Data

One of the disadvantages of NMT models is that good performance is only achieved when trained with much larger amounts of data compared to SMT (Koehn and Knowles, 2017). In some cases, when low amounts of data are available, they underperform SMT models (Dowling et al., 2018).

In a similar way to what was presented in Chapter 3 for SMT, in this section we analyze the performance of the NMT models built with different sizes of randomly-sampled (without duplicates) data. In Table 4.2 we present the evaluation scores of the models trained with 100K, 200K, 500K, 1M and 2M random sentences. We also include in Figure 4.4 the plot BLEU score of these models.

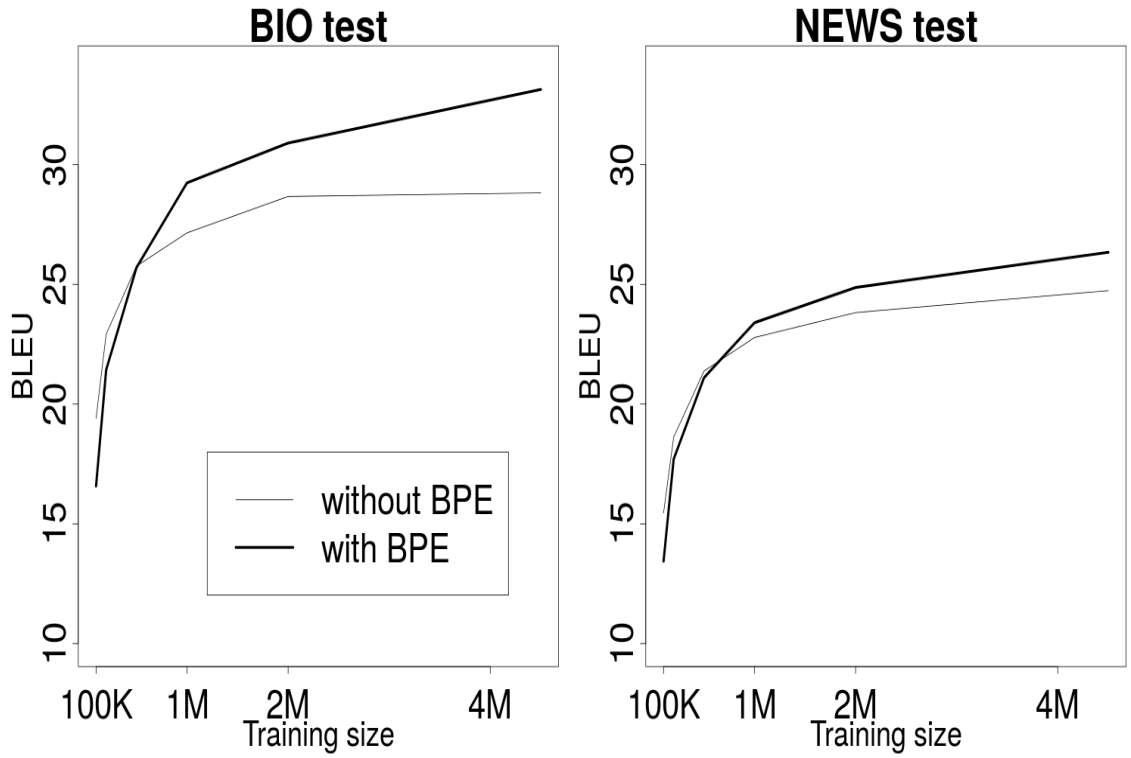

Figure 4.4: NMT models trained in different sizes of data without BPE (thin line) and with BPE (thick line).

In the figure, we see that when using BPE, the translation scores tend to be better. However, it is not the case for smaller models (built with 500K lines or less in Table 4.2) where we can see that the translation scores achieved by the model using BPE are lower than when using the data without word-splitting.

		without BPE		with BPE	
		BIO	NEWS	BIO	NEWS
100K lines	BLEU	19.40	15.46	16.58	13.44
	TER	61.29	67.23	72.11	73.31
	METEOR	22.97	21.20	22.25	19.69
	CHRF3	43.53	38.76	38.59	35.19
200K lines	BLEU	22.92	18.62	21.43	17.70
	TER	56.96	62.37	62.60	65.77
	METEOR	25.82	23.89	26.83	23.60
	CHRF3	47.15	42.60	46.32	41.27
500K lines	BLEU	25.76	21.38	25.70	21.10
	TER	54.69	59.42	55.04	60.47
	METEOR	27.58	25.95	29.78	26.61
	CHRF3	50.08	45.97	51.31	46.04
1M lines	BLEU	27.15	22.78	29.24	23.40
	TER	52.10	57.48	51.24	57.23
	METEOR	28.74	26.90	32.07	28.09
	CHRF3	51.76	47.10	55.07	48.50
2M lines	BLEU	28.67	23.82	30.90	24.87
	TER	50.52	56.46	49.16	55.81
	METEOR	29.14	27.55	33.52	29.16
	CHRF3	52.59	48.15	57.00	50.18

Table 4.2: Results of models built with different sizes of (random) data.

If we compare these models to those of SMT (Section 3.4), we see that the performance of the NMT models (Table 4.2) with random subsets is better than for comparable SMT models (Table 3.2). The exception to this are those models built with small sizes of training data. For example, when comparing the results of the SMT model (*100K lines* column in Table 3.5) with those of the NMT model (*100K lines* column in Table 4.2), we see that SMT obtains better results according to all evaluation metrics.

The performance of NMT models tends to increase as more data are added, whereas that of SMT fluctuates. However, the addition of training data to build NMT models does not necessarily guarantee improvements. The incorporation of poor-quality data may not raise the performance. For example, the outcomes presented in Poncelas et al. (2018c) revealed that the inclusion of too many MT-generated sentences in the training data causes the models to reach an upper bound in performance. Moreover, the incorporation of noisy sentence pairs can hurt the

performance as it causes the model to learn how to produce imperfect translations.

4.4 Experiments

The experiments in Section 4.3 have shown that in NMT it is preferable to use large amounts of data. However, we want to investigate whether using a subset obtained by using TA to train the model can achieve improvements. Note that, the selection algorithms explored in Section 2.5.2 were originally designed for use in SMT. In this chapter, we want to explore whether these methods are also useful in NMT to improve the performance of a model trained with the complete training data.

The experiments we are executing are based on those presented in Section 3.1.2 for SMT. We use TA-retrieved data to build models with increasing amounts of selected data. However, we explore two different procedures in which the selected data can be applied to build improved models:

- *Newly-built*: Build NMT models from scratch, using only the output of TA as training data. In Section 4.3 we have shown that NMT models tend to perform better as more data are used to train. Nonetheless, in the same way as explored in SMT, models built with the selected data may perform better than those built with the full training set.

- *Fine-tuned*: Fine-tune the model (trained with all data) using the output of TA-selected data. In particular, we use the model trained for 12 epochs (*BASE12*) and execute the last epoch using the selected subset, making a total of 13 epochs.

We build the models for translating BIO and NEWS test sets and compare them to the baseline *BASE13* (the model trained with all data for 13 epochs). Note that the goal is again to achieve improvements by using fewer data, and executing less training steps. For example, by reducing the training data by half, building an NMT

model executes the half the number of steps, and when applied fine-tuning for the last epoch we execute $((12 \cdot 1) + (1 \cdot 0.5))/13 = 0.96$ of the steps of BASE13 model.

Note that BPE is applied on the TA-selected data (after the selection). This implies that the execution of the TAs is performed on unsplit data. The main reason to do this is that all the compared models are trained on the same sentences (including those on SMT in Section 3.4).

Nevertheless, we propose as future work to compare these subsets to those retrieved by TAs on data split with BPE (i.e. apply BPE both in the seed and the training data, and execute TA afterwards). This would cause the n-grams in the test set considered by TAs to no longer be a sequence of words, but rather a sequence of sub-words. One of the benefits of this is that new, potentially helpful, n-grams are considered.

We follow the example of the word "daylight" discussed in Section 4.2.2 to illustrate the issue. Imagine the sentence in the test set to be translated is "During the daytime, the sunlight is strong". Although this sentence includes two words, "daytime" and "sunlight", that are expected to be close (in the vector space) to the word "daylight", the TA will not select any sentence containing it (unless the candidate contains other words that overlap with those of the test set). Let us optimally assume that with BPE the words "daylight", "daytime" and "sunlight" are split as: "day" and "light"; "day" and "time"; and "sun" and "light", respectively. Then, if BPE is applied before the execution of the TA the candidate sentence that originally contained "daylight" will be considered by the TA as now it would match the words "day" and "light" of the test set. However, this approach also has its drawbacks because each sub-word n-gram encode less information.

4.5 Results

In Table 4.3 we show the results of the baseline models trained for 12 epochs (*BASE12* columns) and 13 epochs (*BASE13* columns). In the table, we observe

	without BPE		with BPE	
	BASE12	BASE13	BASE12	BASE13
BIO				
BLEU	28.77	28.82	33.29	33.14
TER	50.35	50.24	46.11	46.79
METEOR	29.49	29.49	34.62	34.57
CHRF3	52.97	52.94	59.02	59.08
NEWS				
BLEU	24.63	24.74	26.16	26.34
TER	55.51	55.25	54.41	54.41
METEOR	27.92	27.98	30.00	30.09
CHRF3	48.79	48.95	51.48	51.71

Table 4.3: Results of the model BASE12 and BASE13 (with and without using BPE).

again that the model trained for 13 epochs achieves convergence. If we compare the *BASE12* columns and *BASE13* columns we observe that the improvements are small and we can even find that for the BIO test set, when BPE is applied (*BIO* subtable, *with BPE* column) the model trained for 13 epochs achieves lower results than the model trained for 12 epochs.

The results of the models trained with the selected data are shown in Table 4.4 (models without BPE) and Table 4.5 (models with BPE). In each table we include the scores achieved by the model built using the complete training data trained for 13 epochs (*BASE13* column) and the models trained with the data retrieved by TFIDF, Infrequent *N*-gram Recovery (INR) and Feature Decay Algorithms (FDA) methods. For each method, we present two columns which indicate how the adapted model was built: by training the model from epoch 1 using the selected data (*newly-built* columns), or by fine-tuning the 12th epoch of the general-domain model using the selected data (*fine-tune* columns). The BLEU scores of these models are also presented in Figure 4.5. The two plots at the top show the performance of the models without using BPE, and the two plots in the bottom the results of models with BPE. In each plot, we present both the results of the newly-built models (straight lines) and the fine-tuned models (dotted lines). These plots give a good overview of how the different TAs perform in NMT. We see that the scores achieved by models built

		BASE13	TFIDF		INR		FDA	
			newly-built	fine-tuned	newly-built	fine-tuned	newly-built	fine-tune
BIO								
100K lines	BLEU	28.82	21.74	**29.01**	27.45	**28.99**	28.39	**29.01**
	TER	50.24	63.17	**49.99**	55.64	**49.89**	53.87	**50.09**
	METEOR	29.49	25.66	**29.51**	**29.77**	**29.59**	**29.98**	29.48
	CHRF3	52.94	43.33	52.91	52.27	**52.99**	52.15	52.92
200K lines	BLEU	28.82	25.57	**29.15**	**31.09***	**29.12**	**30.58***	**28.94**
	TER	50.24	56.72	**49.84**	**50.04**	**50.11**	**49.78**	**49.98**
	METEOR	29.49	28.15	29.47	**31.74***	**29.53**	**31.82***	**29.55**
	CHRF3	52.94	49.54	52.77	**55.25**	52.93	**55.12**	**53.08**
500K lines	BLEU	28.82	27.12	**29.23**	-	-	**32.31***	**29.12**
	TER	50.24	55.15	**49.77**	-	-	**48.11***	**49.92**
	METEOR	29.49	29.43	**29.54**	-	-	**32.15***	**29.53**
	CHRF3	52.94	51.90	**53.14**	-	-	**56.15***	52.94
1M lines	BLEU	28.82	27.36	**29.36**	-	-	**31.85***	**29.46***
	TER	50.24	54.42	**49.55**	-	-	**48.57***	**49.91**
	METEOR	29.49	**29.71**	**29.55**	-	-	**31.85***	**29.71**
	CHRF3	52.94	51.64	**53.24**	-	-	**55.99***	**53.34**
2M lines	BLEU	28.82	27.07	**30.03***	-	-	**31.05***	**28.90**
	TER	50.24	54.36	**49.11***	-	-	**48.97***	**50.09**
	METEOR	29.49	29.67	**29.86***	-	-	**31.09***	29.48
	CHRF3	52.94	51.16	**53.79***	-	-	**55.00***	52.83
NEWS								
100K lines	BLEU	24.74	14.00	24.43	19.61	24.65	19.51	24.58
	TER	55.25	71.19	55.50	62.85	55.32	62.43	55.30
	METEOR	27.98	19.66	27.81	24.40	27.92	24.50	27.93
	CHRF3	48.95	34.68	48.52	43.10	48.87	42.98	48.78
200K lines	BLEU	24.74	18.41	24.74	23.08	**24.86**	23.04	**24.79**
	TER	55.25	64.18	**55.23**	58.00	55.29	57.88	**55.23**
	METEOR	27.98	23.69	27.97	27.25	**28.05**	27.22	**28.04**
	CHRF3	48.95	41.63	48.73	47.52	**48.99**	47.27	**49.02**
500K lines	BLEU	24.74	21.44	24.69	-	-	**25.17***	**24.87**
	TER	55.25	61.03	**55.23**	-	-	56.01	**55.18**
	METEOR	27.98	26.04	27.94	-	-	**28.86***	**28.05**
	CHRF3	48.95	45.69	48.73	-	-	**49.83**	**49.09**
1M lines	BLEU	24.74	22.98	**24.78**	-	-	**25.60***	**24.75**
	TER	55.25	58.78	55.36	-	-	**54.97**	**55.24**
	METEOR	27.98	27.01	27.93	-	-	**28.86***	**28.01**
	CHRF3	48.95	47.11	48.80	-	-	**50.09**	48.92
2M lines	BLEU	24.74	23.78	24.74	-	-	**25.85***	24.72
	TER	55.25	57.70	55.36	-	-	**54.54***	**55.22**
	METEOR	27.98	27.65	27.91	-	-	**28.94***	**28.02**
	CHRF3	48.95	48.08	48.79	-	-	**50.09**	48.92

Table 4.4: NMT models fine-tuned with different sizes of selected data (without BPE). The results in bold indicate an improvement over BASE13. The asterisk means the improvement is statistically significant at p=0.01.

		BASE13	TFIDF		INR		FDA	
			newly-built	fine-tuned	newly-built	fine-tuned	newly-built	fine-tuned
BIO								
100K lines	BLEU	33.14	19.99	**33.95***	25.56	**33.52***	27.38	**33.68***
	TER	46.79	65.56	**45.99***	60.85	**45.92***	54.66	**45.97***
	METEOR	34.57	25.01	**34.96***	30.11	**34.77**	30.88	**34.71**
	CHRF3	59.08	43.21	**59.50**	50.91	**59.43**	51.72	**59.24**
200K lines	BLEU	33.14	24.90	**33.97***	29.95	**33.88***	29.79	**33.96***
	TER	46.79	59.18	**46.03***	51.58	**45.90***	52.58	**45.64***
	METEOR	34.57	28.71	**34.89***	33.41	**34.94***	32.80	**35.01***
	CHRF3	59.08	48.97	**59.41**	56.41	**59.56**	55.64	**59.56**
500K lines	BLEU	33.14	27.09	**34.14***	-	-	32.97	**33.75***
	TER	46.79	56.15	**45.60***	-	-	48.58	**45.92***
	METEOR	34.57	30.98	**34.96***	-	-	34.41	**34.92***
	CHRF3	59.08	52.89	**59.69***	-	-	58.62	**59.57**
1M lines	BLEU	33.14	28.55	**34.21***	-	-	**34.31***	**33.29**
	TER	46.79	53.14	**45.65***	-	-	**46.09**	**46.59**
	METEOR	34.57	32.04	**35.15***	-	-	**35.27***	**34.72**
	CHRF3	59.08	54.60	**59.83***	-	-	**59.64**	**59.30**
2M lines	BLEU	33.14	29.72	**34.16***	-	-	**33.83**	**33.73***
	TER	46.79	51.88	**45.71***	-	-	47.52	**46.10***
	METEOR	34.57	32.40	**34.97***	-	-	**35.17**	**34.83***
	CHRF3	59.08	55.48	**59.56***	-	-	**59.57**	**59.36**
NEWS								
100K lines	BLEU	26.34	14.59	**26.41**	18.39	**26.49**	18.92	**26.49***
	TER	54.41	70.61	54.45	65.56	**54.19**	64.81	**54.21**
	METEOR	30.09	20.04	**30.14**	23.55	**30.21**	24.09	**30.21***
	CHRF3	51.71	35.67	51.70	41.03	**51.78**	41.95	**51.80**
200K lines	BLEU	26.34	17.62	26.33	23.00	**26.44**	23.03	**26.55***
	TER	54.41	66.50	54.41	59.42	**54.35**	59.04	**54.17***
	METEOR	30.09	23.37	30.03	27.32	**30.12**	27.62	**30.24***
	CHRF3	51.71	40.71	51.52	47.16	51.67	47.63	**51.89**
500K lines	BLEU	26.34	21.35	**26.44**	-	-	25.60	**26.40***
	TER	54.41	60.94	**54.40**	-	-	55.75	54.47
	METEOR	30.09	26.36	**30.11**	-	-	29.43	**30.10***
	CHRF3	51.71	45.53	51.61	-	-	50.65	**51.71**
1M lines	BLEU	26.34	23.10	**26.46**	-	-	**27.01***	**26.70***
	TER	54.41	59.02	**54.36**	-	-	**53.85***	**54.16***
	METEOR	30.09	27.75	**30.13**	-	-	**30.46***	**30.19**
	CHRF3	51.71	47.82	51.61	-	-	**52.14***	**51.88**
2M lines	BLEU	26.34	24.17	26.32	-	-	**27.42***	**26.39**
	TER	54.41	57.23	54.43	-	-	**53.66***	**54.29**
	METEOR	30.09	28.45	30.01	-	-	**30.66***	**30.12**
	CHRF3	51.71	49.17	51.52	-	-	**52.46***	51.67

Table 4.5: NMT models fine-tuned with different sizes of selected data (using BPE). The results in bold indicate an improvement over BASE13. The asterisk means the improvement is statistically significant at p=0.01.

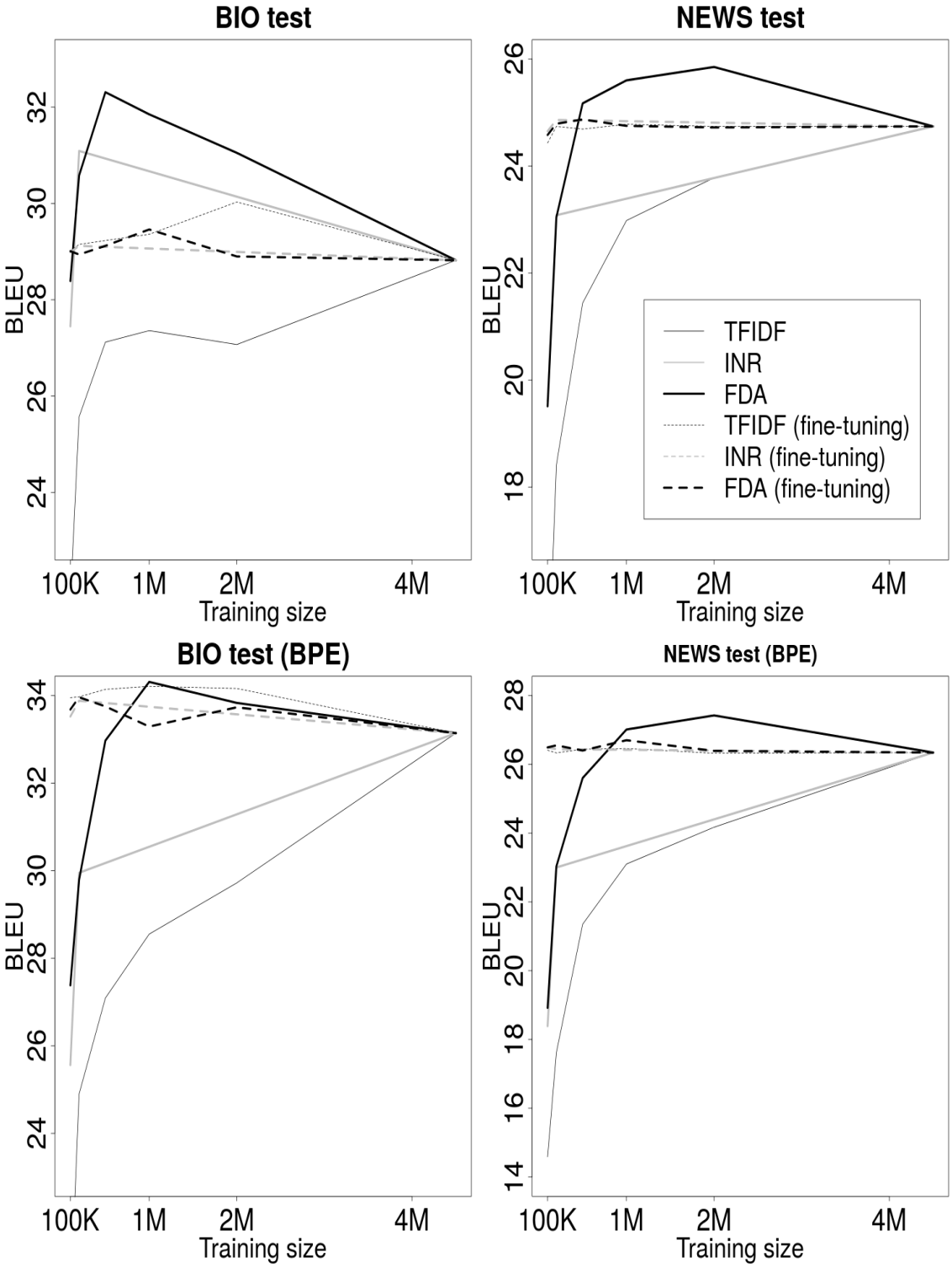

Figure 4.5: Results of the models trained with TA-selected data.

with TA-selected data are higher than those with randomly sampled sentences.

In the following subsections, we analyze the *newly-built* and *fine-tune* approaches individually.

4.5.1 Results of Models Built with Selected Data

We first investigate the results of NMT models built from scratch with subsets of data retrieved by TA. The results of these models can be observed in the *newly-built* columns of Table 4.4 and Table 4.5.

If we compare the scores in these tables with those obtained by SMT in Table 3.5, we see that in general (with the exception of TFIDF newly-built models trained with smaller sets of data) the quality of the translations produced by small NMT models is higher than SMT. In addition, the gap in quality between SMT and NMT increases as more TA-selected data are used to train the model. The *newly-built* models also perform better than those built with the same amount of randomly-sampled sentences (in Table 4.2).

The outcomes in SMT also showed that using larger amounts of TA-retrieved data improved the performance, but additional data then starts to hurt performance. In NMT, models trained with data selected from FDA (newly-built) also shows this effect. We see that in the straight lines of Figure 4.5 the maximum is reached with just a fraction of the data, while using more sentences causes the performance to decrease. In contrast, using the TFIDF method the performance tends to increase as more data are used, but it is always below that of the BASE13 model (i.e. trained with the full set). The performance when using INR is not conclusive. As the data extracted does not exceed 200K sentences it is enough to achieve improvements for the BIO test (without BPE) but insufficient in the other scenarios explored.

As mentioned in Section 4.2.1, the size of the vocabulary considered by the model is set at the beginning of the training step (we use a vocabulary size of 50000 words). Therefore, those words of the test set that are infrequent in the training data may be left out when the NMT model is built. The advantage of using the selected data for training from scratch is that the vocabulary considered by the model is more likely to cover more words of the test set (n-grams of the test set are relatively more frequent in the selected training subset).

The example in Table 4.6 illustrates this issue. We present a sentence of the

	Sentence	Elemente aus Folklore über Klassik bis zu Sport und Akrobatik werden im Jazz Dance verarbeitet.
	Reference	elements ranging from folklore and classical to sports and acrobatics are worked into jazz dance.
without BPE		
100K lines | 2M lines	BASE13	elements from folklore to classical music and Akrobatik are processed in jazz dance.
100K lines	TFIDF	**it is also possible to be used** in jazz **from pop to sports and walks**.
100K lines	INR	**there are** elements from classical music **to sport and acrobatics**.
100K lines	FDA	**artists from classical music** to classical music **to sport and hostels will be** processed in the Jazz Dance.
2M lines	TFIDF	**the Jazz Dance process is performed by classical dance and dance and acrobatics**.
2M lines	INR	-
2M lines	FDA	elements from folklore **through** classical **to sport and acrobatics are used** in jazz dance.
with BPE		
100K lines | 2M lines	BASE13	elements from folklore through classical to sport and acrobatics are processed in jazz dance.
100K lines	TFIDF	elements **of Colombia from classical up** to sport **and advertise in Jazz dance in jazz**.
100K lines	INR	**in the jazz dance of Hivia classic** to sport and acrobatics **in jazz dance**.
100K lines	FDA	**there are currently a variety of styles between** classical **music** to sport and **bikers**.
2M lines	TFIDF	elements from folklore through classical to sport and acrobatics are processed in jazz dance.
2M lines	INR	-
2M lines	FDA	elements from folklore **to** classical **music** and acrobatics are processed in jazz dance.

Table 4.6: Comparison of outputs produced by models built from scratch.

NEWS test set, the reference and the translation generated by the BASE13 model, both with (*with BPE* subtable) and without BPE (*without BPE* subtable). We also include the sentences produced by the newly-built models using subsets of 100K and 2M sentences (the smallest and largest datasets explored in the experiments).

When inspecting the vocabulary considered by BASE13 we discover that "Akrobatik" is not included, because it is not frequent enough in the training set. For this reason, in the produced translation, the word "Akrobatik" is not translated but copied directly from the test set (*BASE13* row in *without BPE* subtable of Table 4.6). In contrast, those models built with the selected 2M sentences correctly translate the word into "acrobatics" because it is included in the vocabulary. The models built with 100K also consider the word "Akrobatik". The sentences generated by models in the *100K* subtable differ significantly from that of BASE13 and (except for INR) the word "acrobatics" is not present.

The problem of limitatiing the vocabulary is (partially) solved by BPE (*with BPE* subtable) as the vocabulary covers more sub-words of the test set.

4.5.2 Results of Fine-Tuned Model

In the *fine-tune* columns of Table 4.4 (without BPE) and Table 4.5 (with BPE), we show the results of the BASE12 model fine-tuned with the selected data. These results can also be seen in the dotted lines in Figure 4.5. In the plots we find that varying the size of data does not impact the quality as much as it does for the newly-built models.

In those models without BPE (*fine-tune* columns of Table 4.4), the fine-tuned models do not achieve any improvements (on the NEWS test set) over the baseline BASE13 or if they achieve improvements (on the BIO test set) those are small and most of them are not statistically significant at p=0.01 (the only exceptions are the models fine-tuned with 500K FDA-selected data and 2M TFIDF-selected data).

In contrast, the effect of fine-tuning when BPE is applied (*fine-tune* columns of Table 4.5) is generally positive when compared to BASE13, as most of the scores are

better. The only exception is TFIDF for the NEWS test set. Note that the NEWS test, which contains 2169 sentences, is larger than the BIO test set, 411 sentences. In general, the use of a larger test set is a disadvantage for the TFIDF technique as it considers each sentence in the test independently. For example, when extracting 100K sentences with TFIDF there are 243 relevant sentences for each sentence of the BIO test set (100000/411 = 243), but only 46 for each sentence of the NEWS test set (100000/2169 = 46).

	Sentence	seit knapp zehn Jahren wird auf dem ehemaligen Truppenübungsplatz in Münsingen nicht mehr geschossen.
	Reference	not a shot has been fired in the former military training ground in Münsingen for almost ten years.
without BPE		
	BASE13	for nearly ten years, the former Truppenübungsplatz was no longer shot.
100K lines	TFIDF	**no more shot has been shot** on the former Truppenübungsplatz **in Münsingen for nearly ten years**.
100K lines	INR	for nearly ten years, the former Truppenübungsplatz **has not been shot in Münsingen**.
100K lines	FDA	for nearly ten years, the former Truppenübungsplatz **has not been shot in Münsingen**.
2M lines	TFIDF	**no more shot has been shot** on the former Truppenübungsplatz **in Münsingen for nearly ten years**.
2M lines	INR	-
2M lines	FDA	for nearly ten years, the former Truppenübungsplatz was no longer shot.
with BPE		
	BASE13	since almost ten years, the former troops in Münsingen will no longer be shot.
100K lines	TFIDF	**for** almost ten years, the former troop **location** in Münsingen **is no longer** shot.
100K lines	INR	**for** almost ten years, the former troop **line** in Münsingen **is no longer** shot.
100K lines	FDA	**for** almost ten years, the former troop **line** in Münsingen **is no longer** shot.
2M lines	TFIDF	**for** almost ten years, the former troops in Münsingen will no longer be shot.
2M lines	INR	-
2M lines	FDA	**for** almost ten years, the former troops in Münsingen will no longer be shot.

Table 4.7: Comparison of outputs produced by the baseline (general-domain model on the 13th epoch) and models fine-tuned with selected data.

The outputs produced by the fine-tuned models are closer to the baseline. For example, the translation generated for the sentence "Elemente aus Folklore über Klassik bis zu Sport und Akrobatik werden im Jazz Dance verarbeitet." shown in Table 4.6 are the same as BASE13 for all fine-tuned models regardless of the size,

as well as whether BPE has been applied or not.

An example of a sentence that has been translated differently by fine-tuned models is presented in Table 4.7. We again show the models that have been adapted by using 100K and 2M lines of TA-selected data. We also indicate in bold the parts of the sentence that differ from the baseline.

The disadvantage of the fine-tuning approach compared to newly-built models is that fine-tuning methods use the same vocabulary as BASE12. This implies that new vocabulary is not introduced.

We see in Table 4.7 (*without BPE* subtable) an example of this. The translation of "Truppenübungsplatz" is not learned by adapted models and it is copied directly from the source sentence. The improvements seen by fine-tuned models consist of variations of the sentence such as different word-ordering (e.g. the TFIDF model generates "for nearly ten years" at the end of the sentence instead of the beginning) or different verb conjugations (e.g. "has not been shot" instead of "was no longer shot"). In addition, adapted models generate more accurate translations. For example, they produce the phrase "in Münsingen" which was omitted by BASE13.

When BPE is applied to the data, the model can learn how to translate compound words. We can see an example of this in Table 4.7 (*with BPE* subtable), where the effect of BPE causes the word "Truppenübungsplatz" to be split into "Truppen/üb/ungsp/latz", and hence the translation by the general-domain model is able to infer "troops" as translation (this is translated as "military training ground" in the reference). Note that BPE splits words statistically and hence sub-words do not necessarily carry any meaning.

In the *with BPE* subtable, we see again that the adapted-model provides different grammatical variations (such as "is no longer shot" instead of "will no longer be shot"), and it can also produce information that was excluded by the general-domain model, even if it has been inferred from part of the word (instead of complete words as seen in the *without BPE* subtable).

For example, as a translation of "Truppenübungsplatz" ("military training ground"),

BASE12 generates the phrase "troops", but the "training ground" part is dropped. However, the adapted models generate "troop location" or "troop line" which includes a possible translation for "ground".

4.6 Conclusion and Future Work

In this chapter we have analyzed the impact of using different amounts of training sentences for building NMT models. We have shown that the inclusion of good-quality data is generally beneficial, but it is preferable to use in-domain sentences. The experiments carried out include using data retrieved by TA to either build NMT models from scratch or to fine-tune a general model during the last epoch. The results reveal that it is possible to find a subset of the training data that, when used to build or fine-tune an NMT model, can obtain better performance than a system built with the full training set.

First, we evaluated models built from scratch with subsets of data. As the vocabulary considered by the models are the most frequent words, it might happen that those terms that are relevant in a particular domain are left out as they do not have enough occurrences in the full training set. For this reason, the use of TA-selected data for training causes the models to consider the most relevant vocabulary.

Among the TAs explored we see that FDA is the best for building NMT models from scratch, only a fraction of data selected by this algorithm is enough to achieve higher performance than models trained with the full training data. The INR method can also retrieve a subset that is favorable to use as training data, but the execution time may be prohibitive if the desired subset is too big (as explained in Section 3.3, after executing the method for 48 hours the amount of sentences retrieved does not exceed 300K). Finally, the TFIDF method has the problem of handling each sentence of the test set independently and so for larger test sets it is not useful.

When fine-tuning the models we find that, as the vocabulary is limited by the

initial model, the increases in performance when using a subset of data are small, if any, when BPE is not used. However, with BPE, not is the general performance of the models are higher, but also fine-tuning with small subsets becomes favorable. Most of the datasets retrieved from the TA used to fine-tune the BASE12 model causes the performance to surpass that of BASE13 (if BPE is applied). The only exception to this is the TFIDF method for larger test sets, such as that of the NEWS domain.

In the remainder of the thesis, the experiments involving the use of TA-retrieved data in NMT models will be carried out using BPE and applying the selected subsets (the sizes will be 100K, 200K and 500K sentences so they are comparable to those experiments executed for SMT) for fine-tuning the 13th epoch of the model trained with all sentences. The reason to use this configuration is that it is a more plausible scenario as the cost of building a model for each new document is high both in terms of time and computational resources.

In addition, we limit the TA explored to INR and FDA only as they are the methods that have shown higher performance. As the selection criterion of TFIDF ignores the selected pool, and each sentence in the test set is considered independently, when using larger test sets as seed, the data obtained may not be enough to use in for fine-tuning.

In the future, we want to explore other methods of adapting NMT models presented in Section 4.1 such as deep fusion, shallow fusion or cost weighting. Regarding the fine-tuning process, we propose to investigate other variations such as gradual fine-tuning or alternative configurations of what we have presented in this chapter. For example, instead of executing one additional epoch with the selected subset, models could be fine-tuned for more iterations to investigate whether the performance improves. Alternatively, instead of fine-tuning the BASE12 model (from the 12th iteration), we can fine-tune that model in the previous iteration of training, such as BASE3, BASE8 or BASE12.

Finally, as mentioned in Section 4.2.2, something that is worth-investigating is

to execute the TA selection on data that has been split with BPE.

Chapter 5

The Use of Alignment Entropy

The experiments with data-selection algorithms revealed that it is possible to improve MT models by using a fraction of available parallel data. In addition, they show superior results to other data-selection algorithms such as context-independent methods. In addition, context-dependent TA showed superior results to other data-selection algorithms such as context-independent methods. The central characteristic to these algorithms is that they penalize the n-grams after they are selected to promote sentences with unseen n-grams.

In the experiments explored in the previous chapter, the TA were executed by using the default configuration. For example, for INR we used the same threshold, and in FDA we used the default values of hyperparameters d and c (see Equation (2.49)). In this chapter we evaluate to what extent modifying the default configuration impacts the performance of MT models trained using the selected data.

Moreover, the configuration influences every n-gram equally. For example, in INR, an n-gram is assumed to be frequent (and so is no longer considered for selection) when there are more than t occurrences. This threshold is the same for every n-gram. Similarly, the penalty (decay function) that FDA applies to an n-gram depends on the decay factor and decay exponent hyperparameters, which are the same regardless of the n-grams.

In this chapter, we want to answer RQ2: **Can word-alignment information**

be useful for improving state-of-the-art TAs? We rebuild the methods so that different configurations can be set to each n-gram individually. By doing that, we can penalize more heavily those n-grams that have a more straightforward translation, with the result that they will require fewer instances in the selected data.

The experiments performed in this chapter are based on the work of Poncelas et al. (2016) and Poncelas et al. (2017) and include the following contributions:

- We perform an analysis of the performance of TAs using different values of the hyperparameters in INR (Section 5.1.1).
- We perform an analysis of the performance of TAs using different values of the hyperparameters in FDA (Section 5.1.2).
- We propose a novel extension for TAs so that the decay of the n-grams becomes dynamic (Section 5.2).
- We evaluate SMT models with data retrieved from TA using alignment entropies (Section 5.4.1).
- We evaluate NMT models with data retrieved from TAs using alignment entropies (Section 5.4.2).

The models built in this chapter consist of both SMT (training from scratch with the selected data) and NMT (fine-tuned BASE12 model after applying BPE). We will see that the changes to the configuration have a different impact in each approach. Nonetheless, the results show that NMT models have superior performance. Therefore, in the future chapters, each algorithm will be evaluated using NMT approaches.

5.1 Transductive Data-Selection Algorithms Parametrization

The first part of this chapter explores what is the impact on translation quality when different configurations of TA are used. In this section, we evaluate SMT and NMT models using TA-retrieved data when different values are used for the hyperparameters (we vary the value of one hyperparameter at a time while having the values of the rest fixed with default values).

5.1.1 Infrequent N-gram Recovery Parametrization

The main hyperparameter in INR is the threshold t, which indicates the number of occurrences of an n-gram needed so it is considered frequent. In our work, we use S to compute the initial count of each n-gram, so at selection time (after initialization), the threshold of an n-gram is $t_{ngr} = t - C_S(ngr)$ (inferred from Equation (2.47)).

The configuration of INR can be modified by altering the value of t. To do that, we introduce a hyperparameter $k \in [0, 1]$ to modulate the threshold as $\frac{t_{ngr}}{k}$. An n-gram ngr is considered infrequent as long as it fulfils the condition $\frac{(t - C_S(ngr))}{k} > C_L(ngr)$. The Equation (2.47) can be reformulated as in Equation (5.1):

$$score(S_{test}, S, s, L) = \sum_{ngr \in Ngr_3(S_{test})} (\min(1, C_s(ngr)) \max(0, t - (C_S(ngr) + kC_L(ngr))) \quad (5.1)$$

where the value of k is 1 in the default configuration. The hyperparameter k can be considered a decay factor for INR (analogous to the hyperparameters in FDA), but instead of being used as a penalty, it indicates how susceptible the n-gram is to being considered frequent.

We select up to 200K sentences using the default configuration of INR ($k = 1$) and another execution of $k = 0.5$. In order to understand the differences between

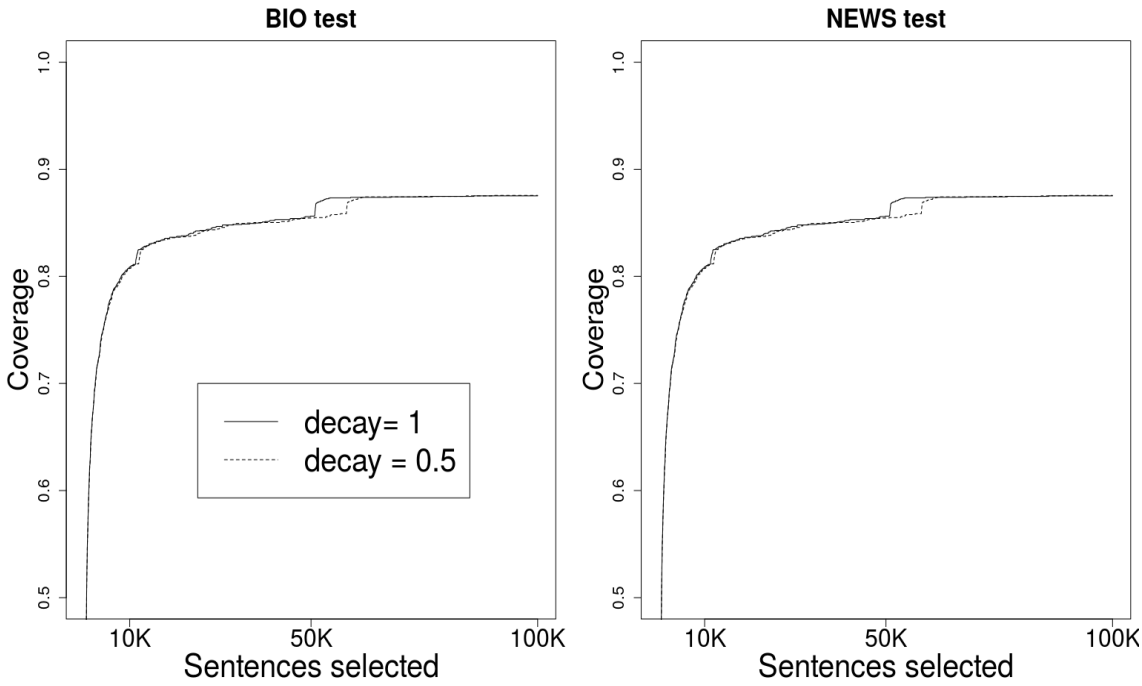

Figure 5.1: Coverage of the test set using different values of k of INR.

the two configurations, in Figure 5.1 we show the coverage (percentage of words of the test set that are present in the selected data) of the first 100K selected sentences selected in the two executions. As we can see, both executions have similar coverage.

In Table 5.1 we see the performance of models built with the data retrieved by these executions of INR. This table is divided into two subtables showing the results for SMT and NMT models (*SMT* and *NMT* columns, respectively). We mark in bold those evaluation scores that indicate a better performance of the model when compared to default (columns including *(default)*) settings and are marked with an asterisk if they are statistically significant at p=0.01. We compare the columns determining that a configuration is better than default when, most of the evaluation metrics for both test sets (both subtables) indicate improvements, and whether they are statistically significant.

However, in the table we see that using $k = 0.5$ generally does not have a positive impact on the results. In SMT, none of the scores indicate improvements when compared to default INR execution. In NMT, only a few scores show improvements but none of them are statistically significant at p=0.01.

		SMT		NMT	
		k=1 (default)	k=0.5	k=1 (default)	k=0.5
BIO					
100K lines	BLEU	25.87	25.47	33.52	**34.00**
	TER	53.85	53.95	45.92	**45.90**
	METEOR	30.67	**30.73**	34.77	34.67
	CHRF3	54.22	**54.38**	59.43	59.32
200K lines	BLEU	25.00	25.00	33.88	33.81
	TER	55.50	**55.41**	45.90	**45.83**
	METEOR	30.12	**30.18**	34.94	34.9
	CHRF3	53.53	**53.87**	59.56	59.51
NEWS					
100K lines	BLEU	19.31	19.20	26.49	26.44
	TER	63.03	63.40	54.19	54.23
	METEOR	26.68	26.63	30.21	30.14
	CHRF3	47.22	47.36	51.78	51.69
200K lines	BLEU	19.64	19.16	26.44	26.43
	TER	63.14	64.74	54.35	**54.20**
	METEOR	27.12	26.65	30.12	**30.19**
	CHRF3	48.09	47.69	51.67	**51.72**

Table 5.1: SMT and NMT models built with different decay of INR. The results in bold indicate an improvement over default configuration $k = 1$.

5.1.2 Feature Decay Algorithms Parametrization

In FDA, the main hyperparameters involved in the decay of the n-grams are the decay factor d and the decay exponent c. In this section, we explore the impact of altering the default values of these hyperparameters.

5.1.2.1 Decay Factor

The decay factor indicates how much the value of the feature decreases after being selected. The value of d must be in the range $(0, 1)$; values greater than 1 would promote the selected n-gram to be promoted instead of penalizing it. Values below 0 would cause the decay to oscillate between positive and negative values. Within the $(0, 1)$ range, lower values of d cause the decay to be faster. The default value of d is 0.5, so the contribution of an n-gram towards the score of the sentence is halved every time it is added to the selected pool. In this section, we explore five different

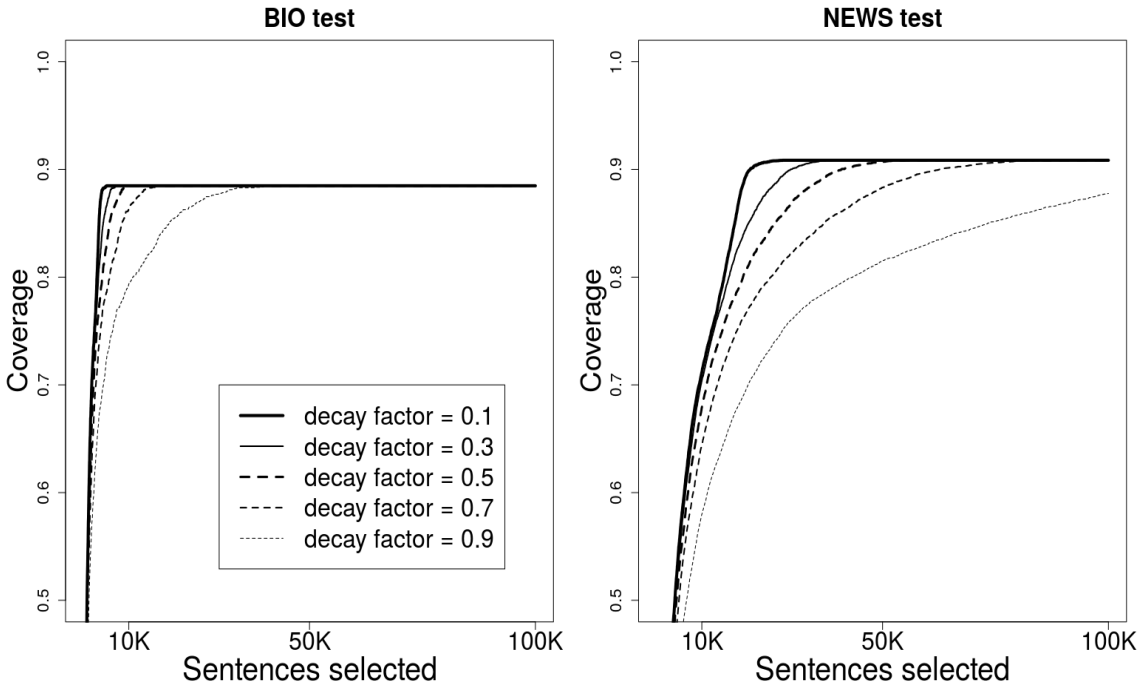

Figure 5.2: Coverage of the test set using different decay factors (values of d in Equation (2.49)).

decay factors: the default value (0.5), two extreme values (0.1 and 0.9), and two values in between (0.3 and 0.7).

In Figure 5.2 we observe how higher decay factors (corresponding to slower decay) cause the method to be more likely to select an n-gram that was already selected instead of choosing those that have not yet been included in the selected pool. This means that higher decay factors correspond to a slower increase in coverage with more data, as can be seen in Figure 5.2.

We also present in Table 5.2 the performance of the five decay factors explored. In general, the results reveal that changes of decay factor in FDA have a bigger impact in SMT than in NMT.

In SMT we see that faster decay values (i.e. smaller values of d) give the best results. For the NEWS test set, most of these scores are statistically significant at p=0.01, and for the BIO test set both using $d = 0.1$ and $d = 0.2$ (trained with 200K and 500K lines, respectively) achieve statistically significant results. We observe that increasing the decay factor value slightly can also lead to improvements. For example, in the 0.7 column, both the model for BIO test built with 500K lines,

		SMT					NMT				
		d=0.1	d=0.3	d=0.5 (default)	d=0.7	d=0.9	d=0.1	d=0.3	d=0.5 (default)	d=0.7	d=0.9
BIO											
100K lines	BLEU	25.58	25.45	25.69	25.42	25.65	**33.71**	**33.79**	33.68	**34.10**	**33.76**
	TER	54.03	**53.46**	53.53	53.86	53.77	46.04	46.30	45.97	**45.91**	**45.89**
	METEOR	30.65	**30.89**	30.84	30.66	**30.86**	**35.08***	**34.75**	34.71	**34.92**	**34.85**
	CHRF3	54.06	**54.42**	54.30	54.12	54.25	**59.66***	**59.28***	59.24	**59.54***	**59.54***
200K lines	BLEU	**25.53***	**25.28**	25.16	25.06	**25.17**	33.78	33.85	33.96	33.70	33.57
	TER	**54.32***	**54.68**	54.84	54.87	**54.80**	46.06	45.83	45.64	46.20	45.98
	METEOR	**30.55***	**30.45**	30.36	**30.43**	**30.45**	34.87	34.86	35.01	34.79	34.83
	CHRF3	**54.21**	53.67	53.92	53.45	53.85	59.42	59.38	59.56	59.30	59.31
500K lines	BLEU	**24.78**	**24.86**	24.60	**24.78**	**25.00**	**33.78**	33.72	33.75	**33.82**	33.49
	TER	**55.72**	**55.44***	56.17	**55.49***	**55.82**	**45.60**	**45.90**	45.92	**45.67**	46.03
	METEOR	29.66	**30.03***	29.73	**29.98***	29.70	34.79	34.88	34.92	34.88	34.91
	CHRF3	53.10	**53.78**	53.17	**53.61**	**53.26**	59.30	**59.62***	59.57	59.38	**59.58**
NEWS											
100K lines	BLEU	**19.65***	**19.76***	19.42	**19.67***	19.28	26.40	**26.52**	26.49	**26.52**	26.43
	TER	**61.60***	**61.84***	62.26	**61.98***	**62.03**	54.31	54.25	54.21	54.25	54.28
	METEOR	**27.02***	**27.06***	26.76	**26.94***	26.58	30.19	30.21	30.21	**30.22**	30.21
	CHRF3	**47.27**	**47.58**	47.07	**47.49**	47.04	51.75	**51.84**	51.80	**51.94**	**51.83**
200K lines	BLEU	**19.89***	**19.88***	19.63	**19.79**	19.53	**26.57**	**26.57**	26.55	**26.56**	**26.63**
	TER	**62.88***	**62.64***	63.27	**62.59**	**62.99**	54.28	54.22	54.17	54.18	**54.06**
	METEOR	**27.17***	**27.24***	27.08	**27.18**	27.05	30.19	30.20	30.24	30.18	30.24
	CHRF3	**48.10**	**48.18***	48.01	**48.22**	47.92	51.81	51.79	51.89	51.76	51.84
500K lines	BLEU	**19.04***	**19.07***	18.83	**19.57***	**18.87**	**26.49**	**26.62***	26.40	26.47	**26.60***
	TER	**64.21**	**64.11***	64.44	**63.80***	**64.30**	**54.20**	**54.10***	54.47	**54.23**	**54.14***
	METEOR	**26.63***	**26.66***	26.50	**26.85***	**26.57**	**30.15**	**30.22***	30.10	**30.15**	**30.28***
	CHRF3	47.59	**47.79**	47.68	**47.85***	**47.75**	**51.72**	**51.90**	51.71	51.70	**51.93**

Table 5.2: SMT and NMT models built with different decay factor of FDA. The results in bold indicate an improvement over default configuration $d = 0.5$. The asterisk means the improvement is statistically significant at p=0.01.

and the models for NEWS test built with 100K and 500K lines, achieve statistically significant improvements. When d is increased to 0.9, none of the models achieve statistically significant improvements at level 0.01.

In NMT we do not find any value of the decay factor that causes the models to clearly outperform that of the default configuration. Using both higher and lower decay factor values, there are evaluation scores that demonstrate improvements (marked in bold).

5.1.2.2 Decay Exponent

The decay exponent is the factor c in Equation (2.49). The range of the decay exponent is $[0, \infty)$ with a default value of 0. Higher values of c cause the selected n-grams to be penalized more heavily (in contrast to the decay factor where higher values cause the decay to be slower). The decay exponent is in the range $[0, \infty)$. Accordingly, in this section, the values investigated are those closer to 0, the lower bound.

In Figure 5.3 we present a comparison of the coverage of different executions of FDA using different decay exponent values. We observe in the figure that the higher the value of the decay exponent is the faster the maximum coverage is achieved. It causes FDA to prioritize the exploration (i.e. diversity of n-grams) over exploitation (i.e. selecting more instances of the same n-gram).

In Table 5.3 we present the results of MT models using the selected data. The outcomes of modifying the decay exponent are similar to what was observed when changing the decay factor.

In SMT, the configuration that achieves maximum coverage most quickly (i.e. using higher values of c) tends to achieve better results. Most of the configurations with $c > 0$ perform better in SMT (numbers in bold in Table 5.3). Overall, the best scores are observed when $c = 1$. Most of the scores in the table have at least one evaluation metric indicating a statistically significant improvement at 0.01, marked with an asterisk. Nonetheless, increasing the value is not a guarantee of better

		SMT				NMT			
		c=0 (default)	c=0.5	c=1	c=2	c=0 (default)	c=0.5	c=1	c=2
BIO									
100K lines	BLEU	25.69	25.47	**25.82**	**25.75**	33.68	**33.71**	**33.79**	**33.95**
	TER	53.53	54.05	53.77	**53.43**	45.97	46.06	**45.69**	**45.76**
	METEOR	30.84	30.48	30.67	30.72	34.71	**34.81**	**34.77**	**34.85**
	CHRF3	54.30	53.87	54.29	54.27	59.24	**59.52***	**59.38***	**59.50***
200K lines	BLEU	25.16	24.96	24.97	25.09	33.96	33.70	**33.98**	33.93
	TER	54.84	55.17	**54.83**	55.13	45.64	45.96	45.84	45.82
	METEOR	30.36	30.23	**30.74***	30.24	35.01	34.83	34.75	34.82
	CHRF3	53.92	53.52	**54.18**	53.49	59.56	59.42	59.36	59.38
500K lines	BLEU	24.60	**24.93**	**24.84**	**24.83**	33.75	33.70	33.54	33.50
	TER	56.17	**55.13***	**55.91**	**55.14***	45.92	46.13	45.98	46.00
	METEOR	29.73	**29.74**	**29.98***	**29.89**	34.92	34.82	34.85	34.86
	CHRF3	53.17	**53.18**	**53.64**	**53.38**	59.57	59.29	59.39	59.50
NEWS									
100K lines	BLEU	19.42	**19.72***	**19.68***	**19.69***	26.49	26.46	26.49	**26.51**
	TER	62.26	**61.36***	**61.77***	**61.37***	54.21	**54.11**	**54.16**	54.24
	METEOR	26.76	**26.90***	**26.94***	**27.09***	30.21	30.20	30.20	30.16
	CHRF3	47.06	**47.25***	**47.54**	**47.42**	51.80	51.78	**51.84**	51.79
200K lines	BLEU	19.63	**19.64**	**19.73**	19.41	26.55	26.48	**26.66**	26.50
	TER	63.27	**63.08**	**62.91***	63.86	54.17	54.17	**54.02**	54.31
	METEOR	27.08	27.08	**27.14**	26.93	30.24	30.19	**30.30**	30.19
	CHRF3	48.01	47.92	**48.05**	47.91	51.89	51.80	**51.95**	51.78
500K lines	BLEU	18.83	**18.95***	**18.89**	**18.95**	26.40	26.44	**26.71***	**26.55**
	TER	64.44	64.50	**64.16***	**64.21***	54.47	**54.06***	**54.17**	**54.21***
	METEOR	26.50	**26.52**	**26.61***	**26.65***	30.10	**30.18**	**30.22**	**30.23***
	CHRF3	47.68	47.60	**47.80**	**47.77**	51.71	**51.76**	**51.92**	**51.85**

Table 5.3: SMT and NMT models built with different decay exponent of FDA. The results in bold indicate an improvement over default configuration $c = 0$. The asterisk means the improvement is statistically significant at p=0.01.

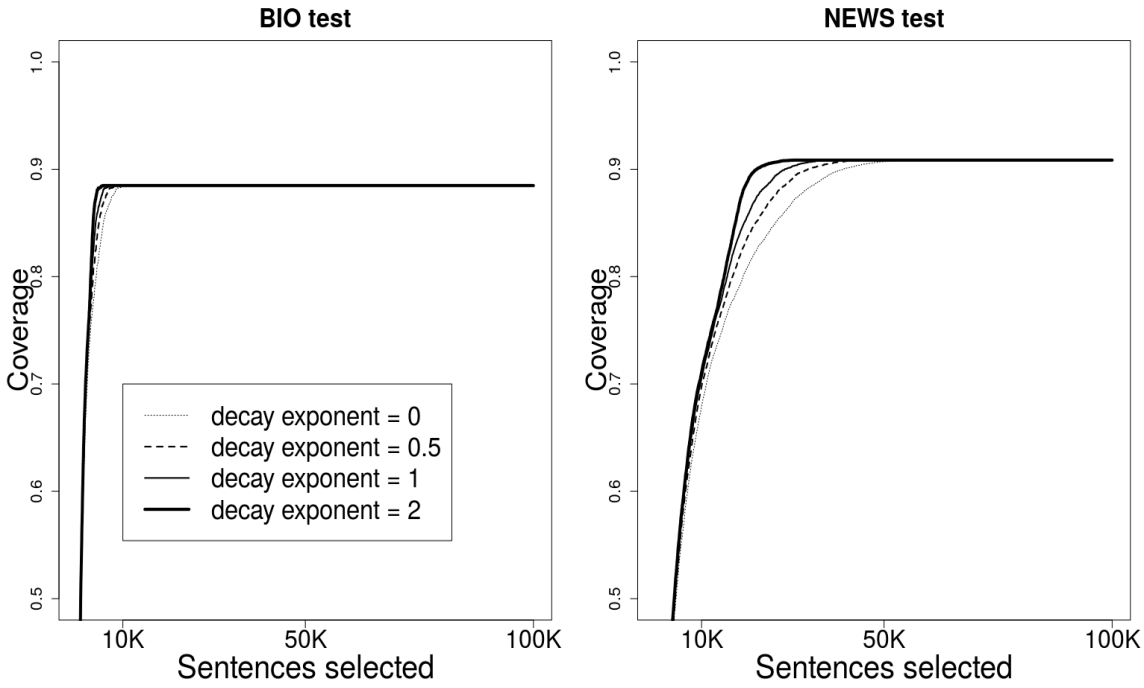

Figure 5.3: Coverage of the test set using different decay exponents (values of c in Equation (2.49)).

results. For example, many scores in $c = 2$ are worse than $c = 1$.

In NMT, varying the configuration of FDA does not seem to have a strong influence on the quality of the models. Changes in the value of c do not seem to have any correlation with the performance. We see for example, that all the configurations where $c > 0$ with 100K sentences for the BIO test achieve improvements, whereas when 500K sentences are used none of the configuration does. Similarly, for the NEWS test set, models trained with 500K sentences achieve statistically significant improvements at p=0.01, whereas none of them do when using 100K or 200K sentences.

5.2 Word Occurrence Balance with Alignment Entropies

In the previous section, we explored the TA when using values of the hyperparameters, but these settings influence all n-grams equally. We contend that different

n-grams should have different decay ratios in order to influence the number of retrieved instances. There should be fewer occurrences of those n-grams that are regularly aligned to the same n-grams in the target language as fewer occurrences are necessary to find a suitable translation. This can be regulated by increasing the decay ratio of those particular n-grams only. For example, a German word such as "Deutschland" should have a more rapid decay as it is expected to be aligned to the same word in English ("Germany").

An indicator of the complexity of translating an n-gram is given by the *average probability of translation ambiguity* (Mohit and Hwa, 2007). Those n-grams with several translations (and with similar translation probabilities) tend to be more ambiguous. We can use these translation probabilities to compute the *alignment entropy* (following Equation (2.49)) as a measure of the ambiguity. Higher entropies would indicate that the n-gram is difficult to translate. In the following subsections, we explore two methods of computing the alignment entropies.

5.2.1 Alignment Entropy based on Translation Probabilities

The first approach consists of using the translation probabilities of each word to compute the alignment entropy (Poncelas et al., 2016). The translation probabilities can be obtained from an alignment tool such as FastAlign or GIZA++. Hence, given a word w_s in the source language, we can retrieve a multiset of translation probabilities TP, where each element $p_i \in TP$ corresponds to the translation probability (computed by an alignment tool) of w_s to be aligned to a word candidate in the target language w_{t_i}. We also add to TP the probability of an n-gram not being aligned to any candidate in order to ensure that $\sum_{p_i \in TP} p_i = 1$. Then, the alignment entropy of w_s can be defined as in Equation (5.2):

$$alignEnt_{unig}(w_s) = \frac{-\sum_{p_i \in TP} p_i \log(p_i)}{\log(|TP|)}. \tag{5.2}$$

For those words whose alignments could not be retrieved, we assign them an

entropy equal to the mean of the entropies of the rest of the words, so that every unigram of the test set has their own associated alignment entropy.

A limitation of Equation (5.2) is that entropies are computed for unigrams only. Extending it to an n-gram is not straightforward as it is not reasonable to expect that, for example, a 3-gram in the source language should always be mapped to a 3-gram in the target language. Therefore, we estimate the entropy of the n-gram as the mean of the entropies of the words in the n-gram as in Equation (5.3):

$$decayEnt(ngr) = \frac{-\sum_{w \in Ngr_1(ngr)} alignEnt_{unig}(w)}{|ngr|}. \tag{5.3}$$

We propose to use the mean because it is reasonable to think that the alignment entropy of an n-gram should be between the boundaries of the values of its individual words. For example, the alignment entropy of the n-gram "John runs" should remain between the entropy of the word "John" (as including the word "runs" increases the range of possibilities to be aligned) and the word "runs" (because adding the word "John" restricts the range of possibilities to be aligned).

5.2.2 Alignment Entropy based on N-gram to Unigram Mapping

The other method we use to compute the alignment probability is to ignore the individual alignment of the words and assume that each n-gram could be aligned to every word in the target sentences (Poncelas et al., 2017). For example, in the pair ⟨"John runs","John rennt"⟩, we assume that the bigram "John runs" is aligned to every unigram: ⟨"John runs","John"⟩ and ⟨"John runs","rennt"⟩. Strictly speaking, the used alignments are not correct. However, as using this approach we expect n-grams with lower entropies to be aligned to a lower variety of words, it provides us with an estimation of how difficult it is to find a translation. In addition, n-grams that tend to appear in domain-specific contexts will have fewer translation candidates and thus be more likely to have lower entropies (under the assumption

of equal translation probability).

Algorithm 2 NGRmap workflow

1: $TP \leftarrow \{\}$
2: Extract the subset $\langle S_{ngr}, T_{ngr} \rangle = \{\langle s_i, t_i \rangle \subset \langle S, T \rangle : ngr \in s_i\}$.
3: **for all** $w_{ti} \in Ngr_1(T_{ngr})$ **do**
4: Add $\frac{C_{T_{ngr}}(w_{ti})}{words(T_ngr)}$ to TP
5: **end for**

The procedure to find the set of translation probabilities TP of an n-gram ngr is presented in Algorithm 2. First, a subset $\langle S_{ngr}, T_{ngr} \rangle$ is retrieved (step 2). This subset consists of all the line-pairs in $\langle S, T \rangle$ including ngr in the source side. Note that despite representing T_{ngr} with subscript ngr, the n-gram is only required (and expected) to be present in S_{ngr}. As we assume ngr could be aligned to any word in the target sentences, we iterate (step 3) over all the unigrams in T_{ngr}. In step 4, we compute the probability of ngr to be aligned to the word w_{ti}, and add it to the set TP. We presume that all the alignments are equally probable, so the translation probability of a word w_{ti} is its frequency in the set TP.

Finally, the aligment entropy is computed using the probabilities from TP as in Equation (5.4):

$$decayEnt(ngr) = \frac{-\sum_{p_i \in TP} p_i \log(p_i)}{\log(|TP|)}. \tag{5.4}$$

5.3 Experiments

We design experiments to investigate the impact on the performance of the MT models when trained with the data retrieved by TA extended with alignment entropies. The alignment entropies are those described in the previous section:

- *FAmean*: alignment entropy as the mean of the alignment entropy using the translation probabilities retrieved by FastAlign (described in Section 5.2.1).
- *GZmean*: alignment entropy as the mean of the alignment entropy using the translation probabilities retrieved by GIZA++ (described in Section 5.2.1).

- *NGRmap*: alignment entropy assuming every *n*-gram can be aligned to each unigram (as described in Section 5.2.2).

These entropies are applied as values in the hyperparameters of TA, as follows:

- INR: In INR it will substitute the value of k in Equation (5.5).

$$score(S_{test}, S, s, L) = \sum_{ngr \in Ngr_3(S_{test})} (\min(1, C_s(ngr)) \max(0, t_{ngr} - decayEnt(ngr) C_L(ngr))). \quad (5.5)$$

- FDA: In FDA it is used as the decay factor, exponent factor, or both. As *n*-grams with higher translation entropies should decay more slowly, $decayEnt(ngr)$ can be directly substituted in d, but when substituting c we use $1-decayEnt(ngr)$ instead, so, for example, when applying the entropies into both hyperparameters, the decay is as in Equation (5.6):

$$decay(ngr, L) = init(f) \frac{decayEnt(ngr)^{C_L(ngr)}}{(1 + C_L(ngr))^{(1-decayEnt(ngr))}}. \quad (5.6)$$

5.4 Results

In Figure 5.4 we show the distribution of the scores of the three methods (*FAmean*, *GZmean* and *NGRmap*) for the BIO and NEWS test sets. We see that *FAmean* has the smallest variation among the three of them. This is caused by the high number of words for which FastAlign did not retrieve an alignment probability, so many *n*-grams have the value of the mean assigned. By contrast, the alignment entropies retrieved by *GZmean* and *NGRmap* are more spread out. Although they have a similar deviation, these methods tend to retrieve higher entropies.

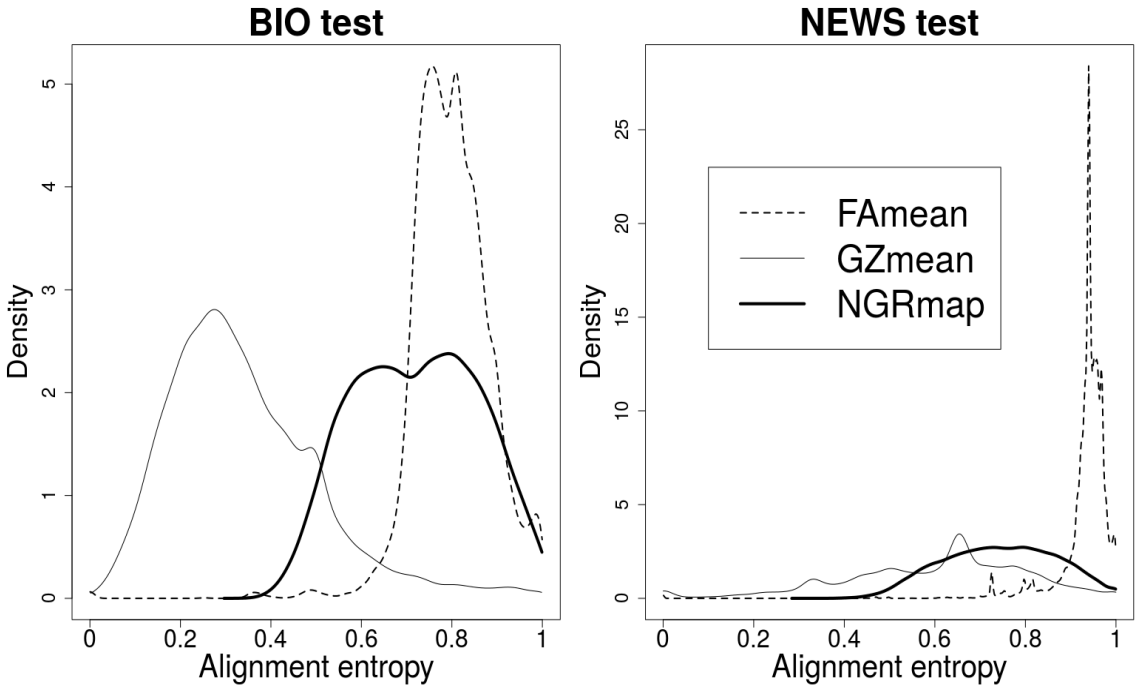

Figure 5.4: Distribution of the alignment entropies.

		INR (default)	FAmean	GZmean	NGRmap
BIO					
100K lines	BLEU	25.87	25.54	25.43	**26.14**
	TER	53.85	53.96	54.25	**53.60**
	METEOR	30.67	**30.81**	30.56	**30.97***
	CHRF3	54.22	**54.34**	54.14	**54.39***
200K lines	BLEU	25.00	24.52	24.73	**25.07**
	TER	55.50	56.01	**55.14**	55.67
	METEOR	30.12	**30.17**	29.93	**30.14**
	CHRF3	53.53	**53.58**	**53.72**	**53.97**
NEWS					
100K lines	BLEU	19.31	**19.33**	**19.35**	19.22
	TER	63.03	**62.69***	**63.02**	63.69
	METEOR	26.68	**26.69**	**26.71**	26.57
	CHRF3	47.22	**47.29**	**47.32**	47.12
200K lines	BLEU	19.64	**19.76**	19.59	**19.78***
	TER	63.14	63.42	63.31	**62.74***
	METEOR	27.12	27.08	26.93	**27.13**
	CHRF3	48.09	**48.10**	47.73	47.87

Table 5.4: Results of SMT models trained with data retrieved by INR method extended with alignment entropies. The results in bold indicate an improvement over default configuration. The asterisk means the improvement is statistically significant at p=0.01.

		FDA (default)	FAmean			GZmean			NGRmap		
			DF	DE	DFE	DF	DE	DFE	DF	DE	DFE
BIO											
100K lines	BLEU	25.69	25.59	**25.70**	24.69	**25.90**	**25.82**	**26.31***	25.61	**26.23**	**25.71**
	TER	53.53	53.81	53.81	54.32	**53.44**	**53.43**	**53.03***	53.82	**53.00***	53.79
	METEOR	30.84	30.17	30.60	30.05	**30.88**	30.81	**31.08***	30.66	**31.02***	**30.73**
	CHRF3	54.30	53.71	54.13	53.67	**54.34**	**54.41**	**54.75**	53.87	**54.49***	54.27
200K lines	BLEU	25.16	24.87	25.15	24.72	25.10	**25.38**	24.99	24.76	**25.48***	**25.30**
	TER	54.84	55.32	54.97	55.11	54.85	**54.46**	54.81	55.62	**54.58**	**54.71**
	METEOR	30.36	**30.49**	**30.62**	30.10	30.31	**30.48**	**30.61**	30.20	**30.55***	**30.48**
	CHRF3	53.92	**54.11**	**54.20**	53.61	53.83	53.90	**54.02**	53.39	**54.27**	53.85
500K lines	BLEU	24.60	23.97	**24.75***	24.15	24.46	24.60	**24.76**	**24.66**	**24.98**	24.28
	TER	56.17	56.39	**55.72**	56.86	**55.81**	**55.89**	**56.05**	**56.00**	**55.59***	56.64
	METEOR	29.73	**29.85**	**30.01**	29.62	**29.75**	**29.74**	**29.85**	**29.82**	**29.99***	29.54
	CHRF3	53.17	**53.47**	**53.49***	53.13	**53.37**	**53.23**	**53.32**	**53.36**	**53.52**	52.96
NEWS											
100K lines	BLEU	19.42	18.87	**20.08***	18.93	18.53	**19.76***	18.04	**19.68***	**19.66***	**19.49**
	TER	62.26	62.27	**60.97***	62.85	62.28	**61.77***	62.98	**61.57***	**62.00***	**61.98***
	METEOR	26.76	26.19	**27.31***	26.20	25.92	**26.96***	25.53	**26.81**	**26.96***	**26.79**
	CHRF3	47.06	46.49	**47.94***	46.59	46.00	**47.49***	45.50	**46.99***	**47.62**	**47.10**
200K lines	BLEU	19.63	18.96	**19.75**	19.10	18.72	19.62	18.30	19.50	19.45	**19.95**
	TER	63.27	63.70	**63.25**	**63.18**	**61.87**	63.27	**63.18**	63.47	63.33	**62.78**
	METEOR	27.08	26.43	27.02	26.47	26.22	**27.13**	26.02	26.93	27.02	**27.21**
	CHRF3	48.01	47.24	47.87	47.25	46.24	48.01	46.22	47.82	47.98	**48.22**
500K lines	BLEU	18.83	**19.42***	18.83	18.37	18.24	**19.39***	18.33	**19.50**	**19.45***	**19.95***
	TER	64.44	**63.30***	64.58	65.21	**64.03***	**63.94***	**63.82***	**63.47**	**63.33***	**62.78***
	METEOR	26.50	**26.91***	26.39	26.16	26.26	**26.74***	26.26	**26.93**	**27.02***	**27.21***
	CHRF3	47.68	**47.87**	47.57	47.02	46.80	**47.90**	46.66	**47.82***	**47.98***	**48.22***

Table 5.5: Results of SMT models trained with data retrieved by FDA method extended with alignment entropies. The results in bold indicate an improvement over default configuration. The asterisk means the improvement is statistically significant at p=0.01.

5.4.1 Results in SMT

The results of extending the INR method with alignment entropies are displayed in Table 5.4, and in Table 5.5 we present the results for FDA. The first columns show the baseline performance, i.e. the model trained with data retrieved from the TA with the default configuration. The remaining columns show the results of models trained with TA-selected data using alignment entropies. In the case of FDA (Table 5.5), each configuration is divided into three columns representing to which hyperparameter the alignment entropies have been applied: decay factor hyperparameter (*DF* columns), decay exponent hyperparameter (*DE* columns), and both decay factor and exponent (*DFE* columns).

We find that using different decays for each n-gram has a positive impact on the models built with the selected data. Among the three techniques explored – *FAmean*, *FAmean* and *NGRmap* – the latter is the method that causes the models to perform better.

In INR, despite the findings in Section 5.1.1 where configurations with k below 1 do not lead to improvements, we can still see that using different values for each n-gram can have benefits. For example, the INR executions with the entropies from *NGRmap* lead to statistically significant improvements at 0.01 for both domains. *FAmean* also achieves improvements for most of the metrics (most of scores in *FAmean* column are in bold) but only the TER score for the NEWS test set using 100K lines is statistically significant. The only method that does not seem to have benefits is *GZmean* as only one experiment achieves improvements for all scores (the model for the NEWS test set using 100K sentence pairs).

The analysis for FDA is more complex as there are two hyperparameters where alignment entropy could be applied. Therefore the first question we want to answer is: where should the alignment entropies be applied? In Table 5.5 we see that those configurations where alignment entropy is involved in the decay factor (i.e. either decay factor only or both decay and exponent) do not perform as well as when they are applied in the decay exponent only. Note that the entropies are in the $[0 - 1]$

range, which implies that when they are applied in the decay factor it can take the extreme values (as the values in the decay factor hyperparameter are also in the $[0-1]$ range) whereas when the entropies are applied in the decay exponent (whose values are in the $[0-\infty)$ range), the values only cover a portion of that range.

By analyzing the *DF* and *DEF* columns, we do not see any experiment where significant improvement at 0.01 for both test sets is achieved, except for the TER metric in the NEWS test set in the *GZmean* column, but we see that most of the metrics indicate worse results. Applying the entropies in the decay exponent, however, has a positive impact on performances as shown in the *DE* columns of both subtables, where most scores are in bold. Moreover we can find statistically significant improvements at p=0.01.

Nonetheless, the performance of the models also depends on the approach that computed the alignment entropies. Using alignment entropies from *FAmean* and *GZmean* approaches in the decay exponent causes some models to achieve statistically significant improvements, but these results are not consistent as they vary depending on the test set. In contrast, the*NGRmap* experiment has a more stable performance and shows its best results when alignment entropies are applied in the decay exponent hyperparameter. In the *DE* column (*NGRmap* experiment), we observe that most of the scores indicate statistically significant improvements at 0.01 for both test sets.

In Table 5.6 we present an example of a sentence and how it is translated by the models trained with data using retrieved by different experiments. We mark in bold those words that differ from the translation generated by the default configuration of the TA.

In the table we see that the extended models perform small changes to the translations such as adding the words “the” or “to”. However, the most interesting difference is how the models translate the German word “sauerstoffreiche”. This word, that according to the reference means “oxygen-rich”, is one of the word with high entropies (0.99, 0.92 and 0.926 values in FAmean, GZmean and NGRmap,

Sentence	dies erschwert es dem Herzen, das sauerstoffreiche Blut in das für die Genesung benötigte Gehirn zu pumpen.
Reference	this makes it harder for the heart to pump the oxygen-rich blood into the brain needed for recovery.
INR	this makes it difficult for the heart, the sauerstoffreiche blood in for the recovery needed brain.
FAmean	this makes it difficult for the heart, the sauerstoffreiche blood in **the** for the recovery required brain **to**.
GZmean	this makes it difficult for the heart, the **oxygen-rich** blood in **this** for recovery needed brain **to**.
NGRmap	this makes it difficult for the heart, the **world** blood in **the** for the recovery needed brain.
FDA	this makes it difficult for the heart, the world blood in it for the recovery needed brain.
FAmean DF	this makes it difficult for the heart, the world blood in **this** for the recovery needed brain **to**.
FAmean DE	this makes it difficult for the heart, the **oxygen-rich** blood in it for the recovery needed brain.
FAmean DFE	this makes it difficult for the heart, the **oxygen-rich** blood in **this** for the recovery needed **to pour** brain.
GZmean DF	this makes it difficult for the heart, the **oxygen-rich** blood in **the** for the recovery needed brain.
GZmean DE	this makes it difficult for the heart, the **oxygen-rich** blood in it for the recovery needed brain.
GZmean DFE	this makes it difficult for the heart, the world blood in **this** for the recovery needed brain.
NGRmap DF	this makes it difficult for the heart, the world blood in **the** for the recovery needed brain **to**.
NGRmap DE	this makes it difficult for the heart, the **oxygen-rich** blood in **the** for the recovery needed brain.
NGRmap DFE	this makes it difficult for the heart, the world blood in **the** for the recovery needed brain **to**.

Table 5.6: Comparison of outputs of the SMT models (100K lines) with data retrieved from INR and FDA. The configurations shown correspond both to default and extended with alignment entropies. In FDA these are applied as decay factor (DF), decay exponent (DE), or decay factor and exponent (DFE).

respectively). In the models trained with data from the default INR and FDA settings, the word "sauerstoffreiche" is kept untranslated (in INR) or incorrectly translated as "world" (in FDA). In contrast, most models extended with alignment entropies (i.e. *GZmean* in INR subtable; and *FAmean DE*, *FAmean DFE*, *GZmean DE* or *NGRmap DE* in FDA) propose the correct "oxygen-rich" translation.

5.4.2 Results in NMT

		INR (default)	FAmean	GZmean	NGRmap
BIO					
100K lines	BLEU	33.52	**33.97**	**33.53**	**33.61**
	TER	45.92	**45.65**	46.23	45.98
	METEOR	34.77	**34.91**	**34.78**	34.77
	CHRF3	59.43	**59.53**	59.37	59.38
200K lines	BLEU	33.88	**33.93**	**34.16***	**33.97**
	TER	45.90	**45.85**	**45.46***	**45.80**
	METEOR	34.94	**34.97**	**34.97**	**34.85**
	CHRF3	59.56	**59.68**	**59.66**	59.52
NEWS					
100K lines	BLEU	26.49	26.48	26.38	26.47
	TER	54.19	**54.13**	54.35	54.23
	METEOR	30.21	30.21	30.18	30.15
	CHRF3	51.78	**51.85**	51.69	51.74
200K lines	BLEU	26.44	**26.56**	**26.46**	**26.48**
	TER	54.35	**54.14**	**54.29**	**54.25**
	METEOR	30.12	**30.24***	**30.13**	**30.20**
	CHRF3	51.67	**51.85**	51.62	**51.78**

Table 5.7: Results of NMT models trained with data retrieved by INR method extended with alignment entropies. The results in bold indicate an improvement over default configuration. The asterisk means the improvement is statistically significant at p=0.01.

Similarly to what presented for SMT, in this subsection, we analyze the NMT models built with data retrieved with extended TA models. The results can be seen in Table 5.7 for INR and Table 5.8 for FDA.

The outcomes of Section 5.1 showed that NMT models are less sensitive to the variation of the hyperparameters of TA. This is also the case when using alignment entropies. The tables show that the results for NMT are similar to their baselines.

		FDA (default)	FAmean			GZmean			NGRmap		
			DF	DE	DFE	DF	DE	DFE	DF	DE	DFE
BIO											
100K lines	BLEU	33.68	**33.74**	33.68	33.84	**33.70**	33.89	33.60	**33.72**	33.61	33.71
	TER	45.97	45.98	45.71	46.12	46.16	45.64	45.94	**45.91**	45.87	46.07
	METEOR	34.71	**34.91***	34.85	34.80	**34.89**	34.95	34.87	**34.77**	34.81	34.73
	CHRF3	59.24	**59.44**	59.42	59.41	**59.41**	59.54	59.43	**59.46**	59.42	59.27
200K lines	BLEU	33.96	33.74	33.82	33.71	33.79	33.74	33.70	33.57	33.74	34.03
	TER	45.64	45.90	45.73	46.14	45.77	46.10	45.78	46.27	46.03	45.69
	METEOR	35.01	34.80	34.98	34.66	34.91	34.78	34.81	34.65	34.81	34.94
	CHRF3	59.56	59.48	59.56	59.19	59.52	59.27	59.40	59.15	59.30	59.53
500K lines	BLEU	33.75	33.42	33.60	33.35	**33.90**	33.81	33.45	**33.82**	33.48	33.80
	TER	45.92	46.19	45.97	46.10	46.06	46.11	46.39	45.93	45.93	45.86
	METEOR	34.92	34.81	34.68	34.78	**34.94**	34.82	34.80	**34.95**	34.88	34.87
	CHRF3	59.57	59.48	59.36	59.16	59.41	59.47	59.32	**59.66**	59.54	59.33
NEWS											
100K lines	BLEU	26.49	26.36	**26.59**	26.45	26.38	**26.54**	**26.51**	**26.77***	**26.55**	**26.52**
	TER	54.21	54.26	54.26	**54.17**	54.34	54.42	54.31	**54.17**	**54.14**	54.29
	METEOR	30.21	30.12	30.19	30.17	30.14	30.18	30.18	**30.27**	30.20	30.17
	CHRF3	51.80	51.73	**51.82**	51.76	51.74	51.71	**51.83**	**51.95**	**51.92**	**51.81**
200K lines	BLEU	26.55	**26.63**	26.42	26.55	**26.66**	26.53	26.52	**26.57**	26.49	26.54
	TER	54.17	**54.06**	54.38	54.29	**54.08**	54.17	54.19	54.22	54.27	**54.16**
	METEOR	30.24	30.24	30.12	30.21	**30.26**	30.19	30.24	30.20	30.14	30.19
	CHRF3	51.89	**51.92**	51.63	51.84	**51.97**	51.82	51.87	51.84	51.79	51.84
500K lines	BLEU	26.40	**26.52**	**26.69***	**26.72***	**26.54**	**26.42**	**26.67***	**26.74***	**26.66***	**26.55**
	TER	54.47	**54.08***	**54.22***	**54.14***	**54.10***	**54.26**	**54.13***	**54.03***	**54.08***	**54.28**
	METEOR	30.10	**30.16**	**30.24***	**30.24***	**30.20***	**30.17**	**30.28***	**30.32***	**30.21***	**30.18**
	CHRF3	51.71	**51.78**	**51.91**	**51.92**	**51.87**	**51.72**	**51.91**	**52.08***	**51.88**	**51.74**

Table 5.8: Results of NMT models trained with data retrieved by FDA method extended with alignment entropies. The results in bold indicate an improvement over default configuration. The asterisk means the improvement is statistically significant at p=0.01.

Nonetheless, we see slight improvements in some of the experiments.

In INR (Table 5.7) most scores of *FAmean* are better than the default INR, but the only statistically significant improvement is the METEOR score in NEWS test set for 200K lines. *GZmean* tends to perform relatively well on the BIO test set (with two scores statistically significant at p=0.01, when trained with 200K lines) but none of the scores are statistically significant on the NEWS test set. Finally, *NGRmap* performs the worst of the three with none of the scores showing statistically significant improvements. In FDA (Table 5.8), the only experiments in which we find statistically significant (at p=0.01) improvements for more than one score are for the NEWS test set when fine-tuning with 500K lines.

Sentence	dies erschwert es dem Herzen, das sauerstoffreiche Blut in das für die Genesung benötigte Gehirn zu pumpen.
Reference	this makes it harder for the heart to pump the oxygen-rich blood into the brain needed for recovery.
INR	this makes it more difficult to pump the heartbeat blood into the brain needed for recovery.
FAmean	this makes it more difficult to pump the **fat** blood into the brain needed for recovery.
GZmean	this makes it more difficult to pump the **fat** blood into the brain needed for recovery.
NGRmap	this makes it more difficult to pump the **fat** blood into the brain needed for recovery.
FDA	this makes it more difficult to pump the fat blood into the brain needed for recovery.
FAmean DF	this makes it more difficult to pump the fat blood into the brain needed for recovery.
FAmean DE	this makes **it difficult** to pump the **heartbeat** blood into the brain needed for recovery.
FAmean DFE	this makes it more difficult to pump the fat blood into the brain needed for recovery.
GZmean DF	this makes it more difficult to pump the fat blood into the brain needed for recovery.
GZmean DE	this makes it more difficult to pump the fat blood into the brain needed for recovery.
GZmean DFE	this makes it more difficult to pump the fat blood into the brain needed for recovery.
NGRmap DF	this makes **it difficult** to pump the **heartbeat** blood into the brain needed for recovery.
NGRmap DE	this makes it more difficult to pump the fat blood into the brain needed for recovery.
NGRmap DFE	this makes it more difficult to pump the fat blood into the brain needed for recovery.

Table 5.9: Comparison of outputs of the NMT models (100K lines) with data retrieved from INR and FDA. The configurations shown correspond both to default and extended with alignment entropies. In FDA these are applied as decay factor (DF), decay exponent (DE), or decay factor and exponent (DFE).

In Table 5.9 we show a sentence translated with the models built in the experiments. We see that the translations produced by the models are indeed very similar to each other (compared to the variations of generated by SMT models presented in Table 5.6). The differences with translations with the default TA set-up consist of the omission of the word "more" and the translations proposed for the word "sauerstoffreiche".

In the table we see that this word has been incorrectly translated as "world" by INR and FDA, but the extensions that propose different words are "fat" and "heartbeat" which are also erroneous. This reveals a drawback of our proposal, as increasing the variety of the n-grams in the target side, it also causes the models to find the correct translation more difficult.

5.5 Conclusions and Future Work

In this chapter, we have analyzed the performance of models using different configurations of TA.

The outcomes observed in Section 5.1 show that in SMT, the best performance (as demonstrated by steeper coverage curves) is achieved when using those settings that penalize n-grams more heavily. This is the case when smaller decay factors or higher decay exponents are used. This implies that in SMT, it is preferable to select a more diverse set of n-grams rather than obtaining too many occurrences of each. The experiments performed also indicate that changes in the hyperparameters have a lower impact on NMT. This is due to the fact that the experiments involving NMT involve fine-tuning rather than building models from scratch.

Additionally, we have extended the algorithms so each n-gram has its own decay ratio. This extension can be used along with the alignment entropies of the n-grams to build models with better performance. The alignment entropy measure provides an estimation of how difficult an n-gram is to be translated. We proposed three methods to compute it. Using the alignment entropy to alter the decay ratio of each

n-gram causes the selected data to include fewer instances of those n-grams that have a straightforward translation.

The retrieved data have been used to build SMT and fine-tune NMT models. The results reveal that changes in the configuration have a greater impact on SMT models. For example, in SMT approaches, the best results are found when using the *NGRmap* method applied in the decay exponent of FDA. However, in NMT this configuration does not always lead to better translation qualities.

In the future, we want to explore ways to compute the alignment entropies (e.g. values of entropies greater than 1 for use in the decay exponent in FDA) or use other methods to find configurations that further improve the results presented here.

One drawback of the proposal presented in this chapter is that increasing the occurrences of an n-gram in the source language can induce a higher variability in the target side, which may cause the MT system trained with this data to produce unwanted translations. The next chapter (in which we aim to improve these methods not by modifying the selection criteria but artificially augmenting the data available) also addresses this issue, by creating a seed of the TA in the target language. This forces the TA to select target-side n-grams of the new seed, which results in a restriction of the variability in the selected target sentences.

In addition, the results observed in this and the previous chapters clearly show that NMT models produce translations of better quality than SMT. Accordingly, in the next chapters, the experiments will be carried out using NMT approaches only.

Chapter 6

The Use of Synthetic Data to Adapt Models

The experiments in the last chapter aimed to improve the performance of TA by altering the criteria for selecting sentences. The sentences retrieved were extracted from the same candidate pool without any supplementary data.

In this chapter, we want to improve the models by augmenting the amount of candidate sentences. However, as we are operating in a scenario where additional data are not readily available, we investigate the use artificially-generated data (produced by an MT engine) when used in combination with TA. In particular, we explore back-translated parallel sets, i.e. a parallel set where the source side has been artificially created by translating target-language monolingual sentences. Note that in this work we use the terms artificial, synthetic or back-translated dataset (or sentences) interchangeably when referring to sentence pairs that have been built following the back-translation technique.

The questions that we want to answer in this chapter is RQ3: **Can the use of synthetic sentences improve the performance of MT models when used in combination with TAs?**

The use of back-translated parallel sentences offers new candidates to be selected by TA. However, the explored TA performed the selection based on overlaps of

n-grams between the test set and the source-side of the parallel data. As the source-side of the back-translated parallel set has been artificially generated, it hinders TA finding overlaps, and thus preventing relevant sentences from being retrieved because the source side may contain errors produced by the MT such as unnatural word order, repetition or omission of certain words.

For this reason, we want to explore how to use TA to find overlaps of n-grams not only in the source side but also in the target side. However, as the test set, which is in the source language, cannot be used as the seed to search target-side n-grams, we propose to generate a synthetic seed (the approximated target-side, or $test_{trg}$) in the target language. This is achieved by translating the test set with a general-domain MT model.

This technique can be positive on its own to retrieve authentic sentence pairs, as it promotes the selection of the same n-grams (those in $test_{trg}$) in the target side. The experiments in the previous chapter revealed that the increase in occurrence of an n-grams in the target side can induce a larger variety of possible translations, and consequently increase the difficulty of learning the appropriate translation. By using a seed in the target side, the n-grams searched are restricted to those of $test_{trg}$.

Finally, the techniques mentioned above are combined to fine-tune models with TA-selected hybrid data (i.e. an assemblage of authentic and synthetic data). This can be brought about either by merging authentic and synthetic parallel sentences before using TA, or by combining independent executions of TA in authentic and synthetic sets. In this chapter we explore three procedures to construct hybrid training data that when used to fine-tune models, the performance achieved is higher than those using subsets of TA-selected authentic data.

The techniques presented here are inspired by the outcomes of Poncelas et al. (2018c) that revealed that the synthetic data are useful even when used in isolation. The proposals of this chapter are based on the techniques in the works of Poncelas et al. (2018a), Poncelas et al. (2018d), Poncelas et al. (2019a) and Poncelas and Way (2019). The contributions are summarized in the following points:

- We propose two approaches of retrieving artificial parallel sentences using TA (Section 6.3).
- We explore the performance of NMT models fine-tuned with synthetic data extracted with TA (Section 6.5.1).
- We introduce a novel technique of using TA with an artificial seed (Section 6.2) and discuss the benefits of retrieving sentences based on the target side (Section 6.5.2).
- We analyze three techniques to retrieve a mixture of authentic and synthetic sentences using TA that cause the performance of models fine-tuned with this data to be higher than using authentic data (Section 6.5.3).

The experiments in previous chapters have shown that NMT models clearly outperform those SMT. This is true not only for models trained with all data but also with data retrieved from TA (regardless of the configuration). For this reason, the experiments performed in this chapter are restricted to NMT models only.

6.1 The Use of Back-translated Data

The work of Sennrich et al. (2016b) aimed to improve NMT models using monolingual data to boost the decoder and improve the fluency of the translation (playing a similar role to a LM). In their work, they propose to use sentences in the target side to build parallel sets. These parallel sets are created by pairing target-language sentences with either a ⟨NULL⟩ token or artificially-generated translations (*back-translation*) on the source side. Note that the authentic data are always on the target side to prevent the model from learning to produce sentences with mistakes. The results revealed that if the proportion of sentences containing the ⟨NULL⟩ token in the source is too high, then the model learns to ignore the source side. The back-translation method, however, causes the performance of NMT models to increase

and it has become popular in the pipeline of training NMT models (Sennrich et al., 2016a; Di Gangi et al., 2017; Lo et al., 2017).

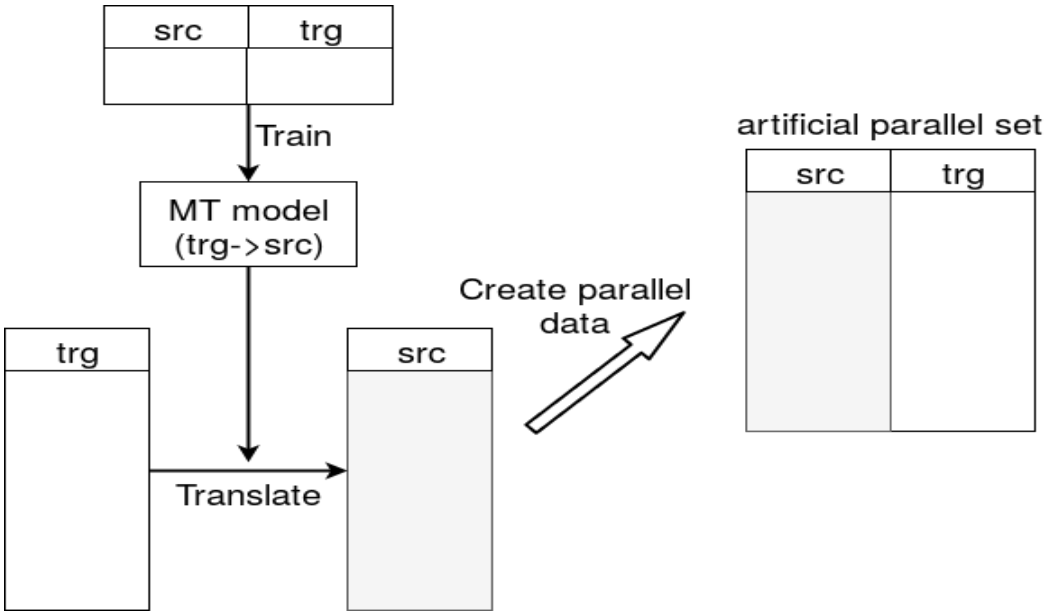

Figure 6.1: Creation of back-translated parallel set.

The procedure for creating back-translated sentences is presented in Figure 6.1: first authentic parallel sentences are used to build an MT in the reverse direction. Then, we use this model to translate a set of sentences into the source language. The created sentences can be paired with the authentic sentences in the target language to create a set of (synthetic) parallel sentences.

6.2 Construction of Approximated Test Set

Another use of synthetic data in the pipeline of selecting sentences with TA is to use artificial sentences as the seed instead of the test set per se. In particular, we use a translation (using an MT) of the test set (*approximated target side* or $test_{trg}$) so it can be used by a TA to retrieve sentences by finding n-gram overlaps with the sentences in the target side instead of the source side (Poncelas et al., 2018a).

In the left-side of Figure 6.2 we show the pipeline of how TAs are generally used to select sentences (the approaches we have followed in previous chapters), and on the right how we propose to use TA with an approximated target side. The first

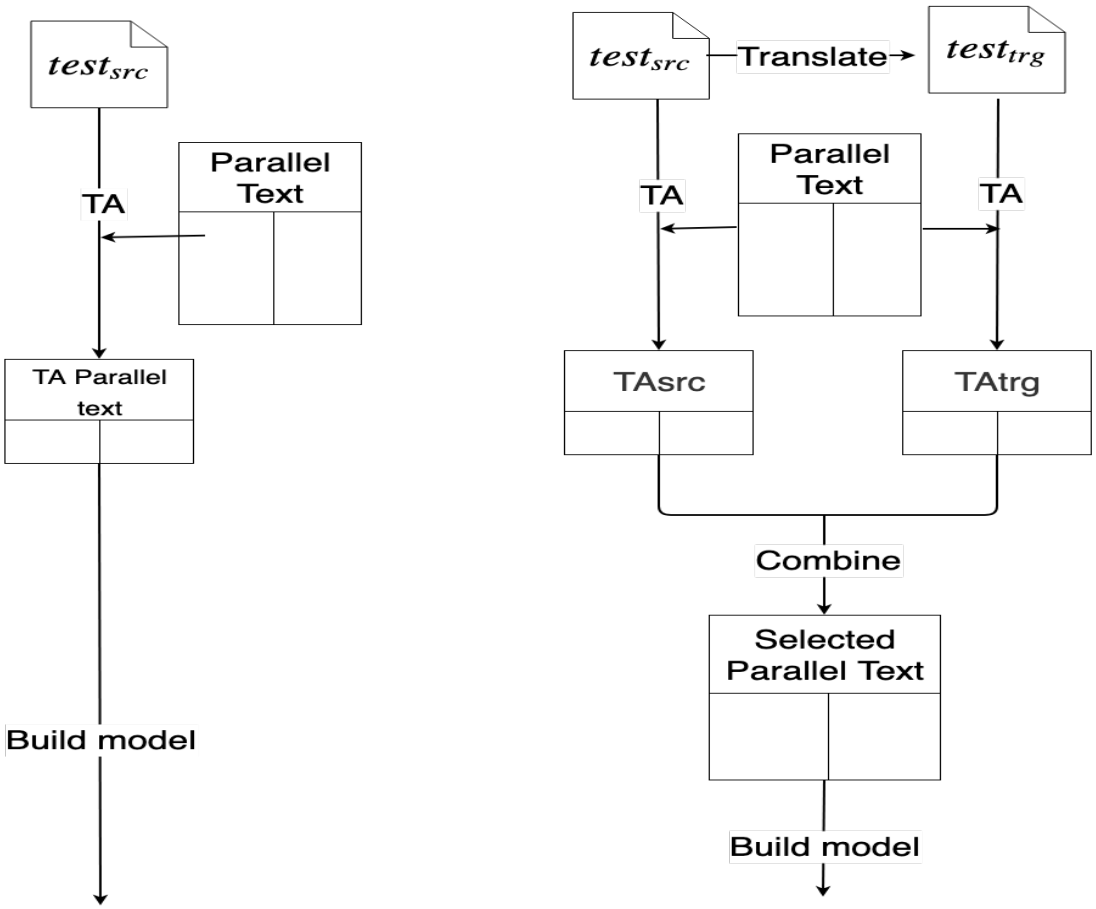

Figure 6.2: Pipeline of the traditional usage of TAs (left) and pipeline of our proposal, using the target-side (right).

step in our approach consists of generating $test_{trg}$ (*translate* step on the right side of Figure 6.2) to use as the seed by a TA to extract parallel sentences. In order to build this approximated target-side, we translate the test set using an MT model which we refer to as the *initial model*.

The output of a TA (using a test set, in the source language, as seed) can be seen as an ordered sequence of sentences as in Equation (6.1):

$$TA_{src} = (s_1^{(src)}, s_2^{(src)}, s_3^{(src)}, ...s_N^{(src)}) \tag{6.1}$$

In a similar way, the output of a TA using the approximated target side as seed (referred to as TA_{trg}) can be formulated as in Equation (6.2):

$$TA_{trg} = (s_1^{(trg)}, s_2^{(trg)}, s_3^{(trg)}, ...s_N^{(trg)}). \tag{6.2}$$

Additionally, both TA_{src} and TA_{trg} can be combined into one training set of N sentences (step *Combine* in the right-hand diagram in Figure 6.2). In this work, we explore the strategy of concatenating both outputs (we propose as future work other methods of merging such as the union or the intersection). This is accomplished by concatenating the top sentences of each subset to obtain a new list of sentences of size N, as in Equation (6.3):

$$TA = (s_1^{(src)}, ...s_{N\cdot\alpha}^{(src)}, s_1^{(trg)}, ...s_{N(1-\alpha)}^{(trg)}) \tag{6.3}$$

where $0 \leq \alpha \leq 1$ indicates the proportion of sentences that are selected from TA_{src} and TA_{trg}.

Note that some of the sentences may be replicated if they have been retrieved by both executions of TA. We decided to keep the duplicates as it may be beneficial to oversample those sentences if there is an agreement of both executions. This approach is an approximation of executing fine-tuning for two iterations: a first iteration on $TA_{src} \cup TA_{trg}$ and a second iteration on $TA_{src} \cap TA_{trg}$.

6.3 Batch and Online Selection

As we have seen, the sentences retrieved by TA are based on n-gram overlaps between the seed and the source side of the parallel data. However, when these methods are used to select back-translated sentences, the n-grams of the test set are searched on MT-generated sentences.

For this reason, we propose two approaches to use the TA for selecting from an artificial parallel set, depending on whether the n-grams are searched for in the source side (artificial) or the target side (authentic) as in Figure 6.3:

- Batch processing: this approach (left-hand side of Figure 6.3) consists of selecting sentences based on the overlap of n-grams between the test set and the synthetic source-side. We designate it as batch processing as it involves back-

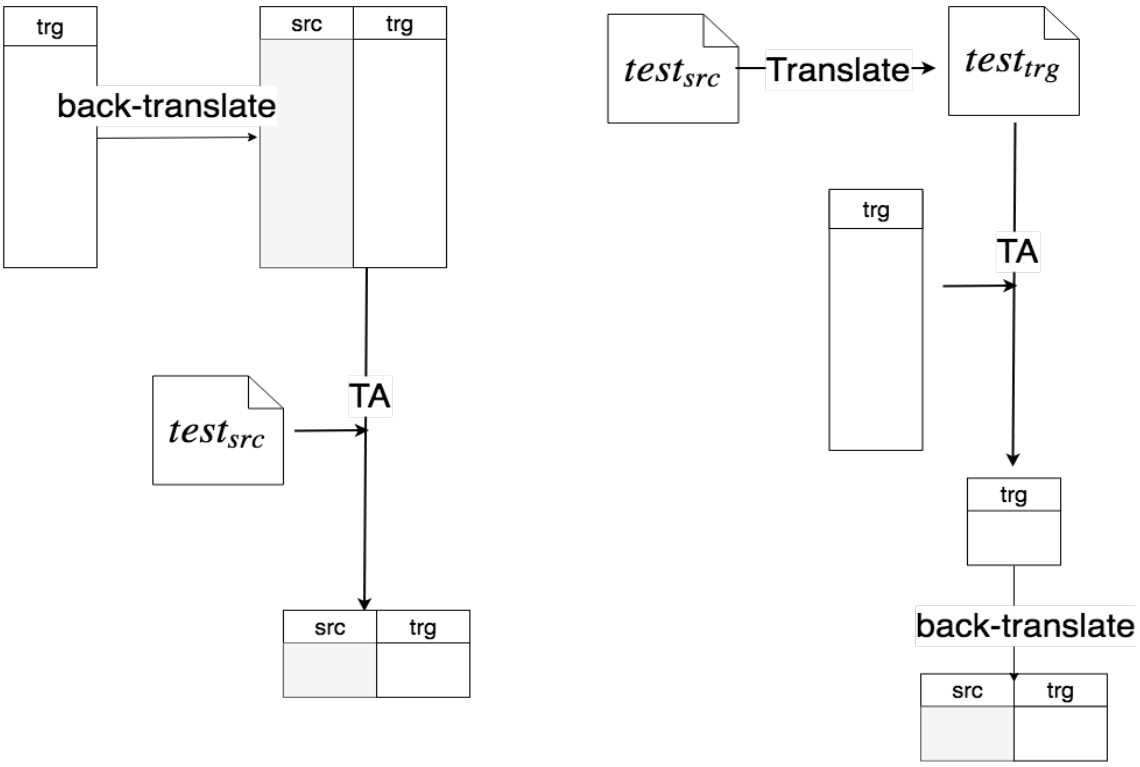

Figure 6.3: Pipeline of the batch (left) and online (right) processing to obtain TA-selected synthetic data.

translating the complete set of monolingual sentences and afterwards selecting sentences from the generated parallel set.

- Online processing: this approach (right-hand side of Figure 6.3) consists of selecting sentences from target-language monolingual data (using an approximated target side as described in Section 6.2) and then back-translating only the selected set (Poncelas et al., 2018d). The characteristic of this approach is that it is not necessary to back-translate the complete data set but only a subset.

6.4 Experiments

In this chapter, we aim to investigate the impact of using artificial data in combination with TA. Similar to our previous experiments, we fine-tune the BASE12 model with 100K, 200K and 500K sentences selected using INR and FDA in which

artificial data are involved.

The experiments carried out are structured in three parts. In the first two parts we examined individually: (i) the performance of TA on back-translated sentences; and (ii), the use of $test_{trg}$, as described in Section 6.2, to retrieve sentences from an authentic parallel set. In the last set of experiments, we gather all data (authentic and synthetic) to fine-tune BASE12 with hybrid subsets.

6.4.1 Models Adapted with Synthetic Data

The first set of experiments consists of models fine-tuned only with synthetic sentences extracted via TA (Poncelas et al., 2019a). These experiments will provide insights on the extent to which using back-translated sentences are useful for improving models.

The back-translated parallel set is created following the work of Poncelas et al. (2018c). We build a model in the reverse direction, English-to-German, with 1M randomly sampled sentences[1] from the *Training set*, with the same configuration described in Section 4.2.1.

We use this model to back-translate all the target sentences of the *Training set* to create a parallel set (S_{synth},T). We are aware that 1M of these sentences have been also used to train the model for back-translation. Despite that, as the model does not overfit, the generated sentences are different from the authentic.

In contrast of experiments in previous chapters, these models are fine-tuned with new data, unseen by the BASE12 model (in previous experiments we have used subsets of the same data used for training).

6.4.2 Use of Approximated Target Side to Select Sentences

We also evaluate the performance of models when fine-tuned with TA-retrieved data using an approximated target side (a synthetic seed) to extract sentences from

[1]This criterion is based on the work of Poncelas et al. (2018c), but we also see that at least in the German-to-English direction in Figure 4.4 in Chapter 4, there is an elbow in the curve is around 1M sentences, and the increases of performance are smaller when using more sentences

the authentic parallel dataset. The purpose is to investigate whether the sentences selected by a TA that uses $test_{trg}$ as the seed can also be used to fine-tune BASE12 and achieve better performance than BASE13. Moreover, in the case of achieving positive results, does it perform better than the model adapted with TA-selected data using the original test set as a seed?

In addition, we explore the performance when both approaches, TA_{src} and TA_{trg}, are combined following the concatenation method presented in Section 6.2. Accordingly, the experiments can be classified as: (i) use TA_{trg} alone for fine-tuning the model, i.e. configuration where $\alpha = 0$ in Equation (6.3); and (ii) concatenate TA_{src} and TA_{trg} (i.e. configuration where different values of α in (6.3) are set).

In the experiments, we use as $test_{trg}$ the output of the BASE12 (we are fine-tuning that model) and explore the values $\alpha = 0.25$, $\alpha = 0.50$ and $\alpha = 0.75$ for concatenation.

6.4.3 Models Adapted with Hybrid Data

Finally, we combine the described techniques to fine-tune models with hybrid data. These experiments can be classified into three approaches depending on how the authentic and synthetic data are combined:

- *hybr* (combine before selection): consisting of selecting from a hybrid set. This involves concatenating both the authentic candidate S_{auth} and artificial S_{synth} sentences. Then the selection is performed on the new set $S_{auth+synth}$.

- *batch* (combine after selection): consisting of performing two executions, TA_{auth} and TA_{synth}, and concatenating them as described in Section 6.3. To avoid confusion, we use γ instead of α to represent the proportions of authentic and synthetic data. We explore the values $\gamma = 0.75$, $\gamma = 0.50$ and $\gamma = 0.25$.

- *online* (combine after selection): consisting on performing two executions of TA, TA_{auth} and $TA_{trg-synth}$. The sentences in $TA_{trg-synth}$ are synthetic sentences retrieved using the approximated target-side as seed (we again use the

translation generated by BASE12). This implies that the overlaps of n-grams are found in the target-side (the side that is authentic) of the parallel data. We concatenate TA_{auth} and $TA_{trg-synth}$ in different proportions of γ. We explore the values $\gamma = 0.75$, $\gamma = 0.50$ and $\gamma = 0.25$.

6.5 Results

6.5.1 Models Adapted with Synthetic Data

		BASE13	INR	synth
BIO				
100K lines	BLEU	33.14	33.52	32.40
	TER	46.79	45.92	47.62
	METEOR	34.57	34.77	**34.86***
	CHRF3	59.08	59.43	**59.69**
200K lines	BLEU	33.14	33.88	32.41
	TER	46.79	45.90	47.82
	METEOR	34.57	34.94	**34.87***
	CHRF3	59.08	59.56	**59.74**
NEWS				
100K lines	BLEU	26.34	26.49	**26.64***
	TER	54.41	54.19	54.92
	METEOR	30.09	30.21	**30.58****
	CHRF3	51.71	51.78	**52.77****
200K lines	BLEU	26.34	26.44	**26.66****
	TER	54.41	54.35	54.85
	METEOR	30.09	30.12	**30.64****
	CHRF3	51.71	51.67	**52.87****

Table 6.1: Results of the models built with different sizes of INR_{src} and INR_{trg} using back-translated data. The results in bold indicate an improvement over BASE13. An asterisk shows that the improvement is statistically significant at p=0.01 when compared to BASE13, and double asterisks when compared to both BASE13 and INR.

In the first set of experiments, we analyze the models that have been fine-tuned with TA-retrieved synthetic data. The results of these models are presented in Tables 6.1 and 6.2. In the tables, we include two baselines in the first columns: the results of BASE13 and the model fine-tuned with an authentic subset. In the third column,

		BASE13	FDA	synth
BIO				
100K lines	BLEU	33.14	33.68	32.28
	TER	46.79	45.97	47.55
	METEOR	34.57	34.71	**34.90***
	CHRF3	59.08	59.24	**59.61**
200K lines	BLEU	33.14	33.96	32.14
	TER	46.79	45.64	47.80
	METEOR	34.57	35.01	**34.87***
	CHRF3	59.08	59.56	**59.74**
500K lines	BLEU	33.14	33.75	32.15
	TER	46.79	45.92	48.42
	METEOR	34.57	34.92	**34.78***
	CHRF3	59.08	59.57	**59.88**
NEWS				
100K lines	BLEU	26.34	26.49	**26.39***
	TER	54.41	54.21	55.25
	METEOR	30.09	30.21	**30.50****
	CHRF3	51.71	51.80	**52.57****
200K lines	BLEU	26.34	26.55	**26.55***
	TER	54.41	54.17	54.97
	METEOR	30.09	30.24	**30.51****
	CHRF3	51.71	51.89	**52.72****
500K lines	BLEU	26.34	26.40	**26.62****
	TER	54.41	54.47	54.83
	METEOR	30.09	30.10	**30.61****
	CHRF3	51.71	51.71	**52.87****

Table 6.2: Results of the models built with different sizes of FDA_{src} and FDA_{trg} using back-translated data. The results in bold indicate an improvement over BASE13. An asterisk shows that the improvement is statistically significant at p=0.01 when compared to BASE13, and double asterisks when compared to both BASE13 and FDA.

we show the results of the model fine-tuned with synthetic data. We mark in bold those scores that are higher than BASE13 and mark with one (or two) asterisk if the improvements are statistically significant at p=0.01. Those scores marked with two asterisks indicate that the improvements are also statistically significant when compared to the model fine-tuned with authentic data.

The results in the tables show that the use of synthetic data for fine-tuning is beneficial (compared to BASE13) as most of the scores are marked in bold (and according to METEOR these improvements are statistically significant at p=0.01). However, we observe a disagreement between BLEU, TER, and METEOR. TER scores indicate worse translation qualities (higher scores) than the baselines (none of them are in bold as higher scores indicate worse translation quality). In contrast, METEOR scores tend to be higher than the baselines which indicates that the output also differs in terms of word choice (METEOR is the only metric that does not penalize different word or phrases if they have a similar meaning).

When comparing the models fine-tuned with synthetic data to those fine-tuned with a subset of authentic data, the improvements are not so evident. The statistically significant improvements are only achieved for the NEWS test set (the numbers marked with two asterisks in NEWS subtable in Tables 6.1 and 6.2) for METEOR scores (and BLEU when using 500K lines), whereas for the BIO test set none of them are statistically significant at p=0.01.

In Table 6.3 we show a few examples of English sentences in the training data, along with the authentic and synthetic counterparts. These examples show positive and negative aspects of using back-translated sentences for training MT models.

In the first row, we see that in the authentic set we have the pair ⟨“10 %!”,“one tenth!”⟩ whereas the synthetic counterpart is ⟨“ein Zehntel!”,“one tenth!”⟩. Although both “10 %!” and “ein Zehntel!” have the same meaning the last one is a literal translation. A different rephrasing may increase the chances of the phrase being retrieved. For example, in these experiments the sentence ⟨“ein Zehntel!”,“one tenth!”⟩ is selected as the test set contains the word “Zehntel”, whereas the authentic

	German (auth)	German (synth)	English
1	10 %!	ein Zehntel!	one tenth!
2	laut dieser Sichtweise sind Ausgaben einfach Ausgaben.	nach Ansicht dieser Ansicht Ausgaben auszugeben.	indeed, according to this view, spending is spending.
3	er ist verheiratet und hat zwei Kinder.	seitdem hat er eine lange Karriere auf der Bühne, im Film und im Fernsehen absolviert und hat sich auch als Sängerin und Autor in den letzten Jahren etabliert	since then, he has had a long career on stage, in film and on television. he has also established himself as a singer and an author in recent years.
4	nach Krankenhausangaben wurden zwei um die 50 Jahre alte Männer durch das Beben schwer verletzt: einer sei von einem herabfallenden Schornstein getroffen worden, der andere habe durch Glas Schnittwunden erlitten. außerdem seien mehrere Menschen durch herabstürzende Gegenstände in ihren Wohnungen leicht verletzt worden.	am Samstag wird es eine weitere Komödie, "La pasone" von Carlo Mazzacurati Italiens, geben, die die fruchtbare Silvio Orlando, die ein washed-up filmin der Toskana ist, in einer Nachbarkapelle aus dem 16. Jh.	Saturday will feature another comedy, "La Passione" by Carlo Mazzacurati of Italy starring the prolific Silvio Orlando, who plays a washed-up filmmaker who is forced to set his last-chance project in Tuscany after a plumbing disaster at his country home damages a 16th-century fresco in a neighbouring chapel.
5	folglich übernimmt Informatic SA keine Gewährleistung für ihre Richtigkeit, ausser sie wurden vom Kunden schriftlich oder per E-Mail ausdrücklich für obligatorisch erklärt.	. . , ...	par conséquent, Informatic SA ne donne donc aucune assurance quant à leur exactitude à moins qu'elles n'aient été expressément déclarées obligatoires par écrit ou par e-mail par le client.

Table 6.3: Examples of back-translated sentences

pair ⟨"10 %!","one tenth!"⟩ is not retrieved.

In the second row, we see an example of a synthetic sentence that is unlikely to have been produced by a native German speaker (observe for example that in the authentic sentence the word repeated, "Ausgaben", corresponds to the English repeated word "spending", but in the MT-generated sentence the repeated word is "Ansicht" which means "point of view") so the number of n-grams that overlap with an authentic seed can be expected to be lower.

An advantage of using back-translation is that it can solve the problem of having noisy parallel sentences. The third example in the table presents the authentic sentence-pair ⟨"er ist verheiratet und hat zwei Kinder.","since then, he has had a long career on stage, in film and on television. he has also established himself as a singer and an author in recent years."⟩ which do not convey the same meaning at all. In the machine-translated sentence, the source-side is "seitdem hat er eine lange Karriere auf der Bühne, im Film und im Fernsehen absolviert und hat sich auch als Sängerin und Autor in den letzten Jahren etabliert". This is closer in

meaning to the target-side sentence, and therefore more useful for training a model (however we see mistakes such as translating the word singer as a female singer, "Sängerin", although the subject of the sentence indicates that he is a male singer (Vanmassenhove et al., 2018)). A similar example is presented in the fourth row. The authentic sentence is not a good translation of the English sentence whereas the synthetic one is more accurate. We observe again that names in the English side such as "Carlo Mazzacurati" or "Silvio Orlando" are present in the synthetic sentence, (*German (synth)* columns) whereas in the authentic sentence they are absent. In this case, again, it is preferable to use the synthetic sentence-pair in the training or fine-tuning of the model.

In the row five there is another positive effect of using TA on back-translated data. As we can see, the target side of that example is not an English sentence but French. Using this pair as training data would cause a negative impact on the performance of the MT. In this case, the synthetic German-side consists of a sequence of dots (the model used for back-translation is not capable of translating from French). Therefore, as TAs search n-grams in the source side, this would cause to discard such unfavourable sentence pair.

6.5.2 Models Adapted Using Approximated Target Side

The second set of experiments comprise executing TA using a $test_{trg}$ as the seed to retrieve authentic sentences. The performance of the models is presented in Table 6.4 (using INR) and Table 6.5 (using FDA). The tables again include two baselines: BASE13 and the model adapted with default TA (i.e. using $\alpha = 1$). The remaining columns show the performance of the models fine-tuned with a combination of TA_{src} and TA_{trg} in different proportions, given by the value of α. As we have two baselines we mark with one asterisk those results that are statistically significant improvements at p=0.01 compared to BASE13 and two if they are also when compared to TA with $\alpha = 1$.

First, we observe that using the $test_{trg}$ alone is useful to adapt the models to

		BASE13	$\alpha = 1$	$\alpha =$ 0.75	$\alpha =$ 0.50	$\alpha =$ 0.25	$\alpha = 0$
BIO							
100K lines	BLEU	33.14	33.52	**33.46**	**33.47**	**33.70***	**33.39**
	TER	46.79	45.92	**46.31**	**46.20**	**45.91***	**46.05***
	METEOR	34.57	34.77	**34.78**	**34.63**	**34.88***	**34.75**
	CHRF3	59.08	59.43	**59.33**	**59.29**	**59.44**	**59.23**
200K lines	BLEU	33.14	33.88	**33.62***	**34.03***	**33.86***	**33.43**
	TER	46.79	45.90	**45.89***	**45.70***	**45.63***	**45.90***
	METEOR	34.57	34.94	**34.77**	**35.02***	**34.89***	**34.95***
	CHRF3	59.08	59.56	**59.25**	**59.55**	**59.53**	**59.39**
NEWS							
100K lines	BLEU	26.34	26.49	**26.59**	**26.64***	**26.55**	**26.59***
	TER	54.41	54.19	**54.08***	**54.17***	**54.13**	**54.30***
	METEOR	30.09	30.21	**30.30***	**30.37****	**30.33****	**30.34****
	CHRF3	51.71	51.78	**51.93**	**52.10***	**52.07**	**52.12****
200K lines	BLEU	26.34	26.44	**26.61****	**26.66****	**26.55**	**26.49**
	TER	54.41	54.35	**54.10****	**54.06****	**54.13***	**54.37***
	METEOR	30.09	30.12	**30.25****	**30.28****	**30.29****	**30.27***
	CHRF3	51.71	51.67	**51.93**	**51.96**	**51.97**	**52.02****

Table 6.4: Results of the models built with different sizes of INR_{src} and INR_{trg} using authentic data. The results in bold indicate an improvement over BASE13. An asterisk shows that the improvement is statistically significant at p=0.01 when compared to BASE13, and double asterisks when compared to both BASE13 and $\alpha = 1$.

the test set. Most of the scores of column $\alpha = 0$ are marked in bold which indicates that the performance is higher than BASE13. In addition, many of these indicate statistically significant improvements (marked with one or two asterisks) at p=0.01. Nonetheless, compared to models fine-tuned with the subset of authentic data ($\alpha = 1$ column), the improvements are smaller and only a few achieve statistically significant improvements for the NEWS test set (e.g. INR using 100K sentences or FDA using 500K sentences in $\alpha = 0$ column).

In general, the best performance is seen when combining TA_{src} and TA_{trg} ($\alpha =$ 0.75, $\alpha = 0.5$ and $\alpha = 0.25$ columns). However, we cannot establish an optimal value of α. In INR the best results are found when using the $\alpha = 0.5$ configuration whereas in FDA the best scores are found with the $\alpha = 0.75$ configuration, although we do still find examples in which $\alpha = 0.25$ achieves better results, e.g. in the 100K

		BASE13	$\alpha = 1$	$\alpha =$ 0.75	$\alpha =$ 0.50	$\alpha =$ 0.25	$\alpha = 0$
BIO							
100K lines	BLEU	33.14	33.68	**33.77***	**33.91***	**33.91***	**33.31**
	TER	46.79	45.97	**46.11***	**45.99***	**45.97***	**46.49**
	METEOR	34.57	34.71	**34.73**	**34.76**	**34.85**	**34.63**
	CHRF3	59.08	59.24	**59.22**	**59.31**	**59.34**	58.86
200K lines	BLEU	33.14	33.96	**34.14***	**33.75***	**33.91***	**33.70***
	TER	46.79	45.64	**45.90***	**45.74***	**45.96***	**45.72***
	METEOR	34.57	35.01	**35.03***	**34.91***	**34.84***	**34.96***
	CHRF3	59.08	59.56	**59.35**	**59.33**	**59.28**	**59.45**
500K lines	BLEU	33.14	33.75	**34.06***	**33.58***	**33.54***	**33.36**
	TER	46.79	45.92	**45.52***	**45.93***	**45.74***	**46.17**
	METEOR	34.57	34.92	**34.96***	**34.85**	**34.94***	**34.85***
	CHRF3	59.08	59.57	**59.49**	**59.46**	**59.42**	**59.36**
NEWS							
100K lines	BLEU	26.34	26.49	**26.65***	**26.42***	**26.43**	26.33
	TER	54.41	54.21	**54.12***	**54.13***	**54.16***	**54.16***
	METEOR	30.09	30.21	**30.27***	**30.22***	**30.19**	**30.20**
	CHRF3	51.71	51.80	**51.99**	**51.84**	**51.86**	**51.81**
200K lines	BLEU	26.34	26.55	**26.65****	**26.51**	**26.52**	**26.54***
	TER	54.41	54.17	**54.12***	**54.13***	**54.21***	**54.04***
	METEOR	30.09	30.24	**30.27****	**30.25***	**30.25***	**30.27***
	CHRF3	51.71	51.89	**51.99***	**51.90**	**51.95**	**51.98**
500K lines	BLEU	26.34	26.40	**26.58****	**26.71****	**26.54**	**26.50**
	TER	54.41	54.47	**54.14****	**54.12****	**54.15****	**54.04****
	METEOR	30.09	30.10	**30.28****	**30.28****	**30.24****	**30.28****
	CHRF3	51.71	51.71	**51.98**	**51.95****	**51.94**	**51.94**

Table 6.5: Results of the models built with different sizes of FDA_{src} and FDA_{trg} using authentic data. The results in bold indicate an improvement over BASE13. An asterisk shows that the improvement is statistically significant at p=0.01 when compared to BASE13, and double asterisks when compared to both BASE13 and $\alpha = 1$.

row of BIO subtable (in either Table 6.4 or Table 6.5). We believe that the language pair is another factor to be considered when choosing the best value of α. In these experiments, finding overlapping n-grams in English is easier than in German, as the latter language has more complex inflection and compounding. We propose as future work to evaluate this technique on other language pairs.

Intuitively, models built using the data selected using the approximated target side as seed alone should have worse performance than using the original test set. As

$test_{trg}$ is artificially generated, it may contain errors. Therefore, a TA that bases the decision to extract sentences on that seed may not select the best ones. However, basing the decision of the selection solely on the test set solely also has limitations. Although it guarantees that the selected source sentences are similar to the test set, it does not provide any information about the target side of the selected sentences. Therefore, as we have seen in the previous section, it may still select sentences with target-side translations that are wrong or not suitable given the domain of the test set, and so hurt the final translation accuracy.

German	English		
INR-retrieved data		pos INR_{src}	pos INR_{trg}
die 40er -Jahre sind das "goldene Zeitalte" des Kinos, viele Millionen Menschen strömen wöchentlich in die Lichtspielhäuser.	family Plot (1976) was his last film.	44	-
FDA-retrieved data		pos FDA_{src}	pos FDA_{trg}
diese Zahl ist mehr als doppelt so viel, als vor 10 Jahren.	famous pieces from the 19th century include those by Delacroix, Gauguin, Monet, Renoir and Corot.	50	-
verglichen mit dem Vorjahr entspricht dies einer Umsatzsteigerung von 6 %.	this is an increase of 6 % compared to the previous year.	-	1
Herr Präsident, ich sage Ihnen, das habe ich getan.	yes, I did.	-	30

Table 6.6: Examples of sentences retrieved by TA_{src} and TA_{trg}.

In Table 6.6 we show examples of sentence pairs retrieved by INR_{src}, INR_{trg}, FDA_{src} and FDA_{trg}. In the table, we also indicate the position of the sentence pair in the retrieved set. For example, the value of 44 in the column $posINR_{src}$ indicates that it corresponds to the sentence pair $INR_{44}^{(src)}$. In the columns with positions, we indicate with a "-" if the sentence pair is not found in the top-200K sentences retrieved by INR, or the top-500K sentences retrieved by FDA.

In the first row, we see that the sentence "die 40er -Jahre sind das 'goldene Zeitalte' des Kinos, viele Millionen Menschen strömen wöchentlich in die Lichtspielhäuser." has been selected by INR_{src}. However, this is a noisy sentence as the source side and target side are not an accurate translation of each other (we can see for example that the year "1976" in the target side is not present in the German

side). A similar example of a selected noisy sentence happens for FDA with the sentence "diese Zahl ist mehr als doppelt so viel, als vor 10 Jahren." (this is again easily noticeable as the names "Delacroix, Gauguin, Monet, Renoir and Corot" are only present in the English-side sentence). Including these sentences in the set data will cause the model to learn incorrect translations.

The previous examples have successfully been excluded from the output when using the target side as seed. In the table we can also find examples of noisy sentences retrieved by TA_{trg} such as "Herr Präsident, ich sage Ihnen, das habe ich getan." which in the target side has the incorrect translation of "yes, I did.". In spite of that, the negative impact of including this sentence is smaller; in this case, it is only present in TA_{trg}. This means that the n-grams of the source side of this sentence pair are not likely to be present in the test set (we cannot guarantee that they are not included in the test set as we are only analyzing the top-200K and the top-500K sentences of selected data).

In the table, we also find sentence pairs exclusively in TA_{trg} such as ("verglichen mit dem Vorjahr entspricht dies einer Umsatzsteigerung von 6%.","this is an increase of 6% compared to the previous year.") which is not a noisy sentence. This sentence is not included in TA_{src} as the n-grams in the German side do not match those in the test set, but including the back-translated sentence may be useful in NMT as it is similar to the authentic counterpart in the vector space.

Combining the outputs of TA_{src} and TA_{trg} causes the training data to be reinforced with sentences with relevant translations. Note that mixing the outputs of the two executions of TA cause some sentence pairs to be replicated, as there is an overlap of the outputs. Nonetheless, the number of unique sentence pairs of each TA-retrieved data set is above 90%, regardless of the value of α.

6.5.3 Models Adapted with Hybrid Data

The last set of experiments consists of combining authentic and synthetic parallel data to adapt models. We present in Table 6.7 the results using INR, and Table

			hybr	batch			online		
		INR $_{\gamma=1}$	hybr	$\gamma =$ 0.75	$\gamma =$ 0.50	$\gamma =$ 0.25	$\gamma =$ 0.75	$\gamma =$ 0.50	$\gamma =$ 0.25
BIO									
100K lines	BLEU	33.52	**33.87**	**33.88**	33.50	**33.67**	33.46	33.62	33.13
	TER	45.92	46.17	46.23	46.63	46.60	46.40	46.39	46.64
	METEOR	34.77	**35.01**	**35.03**	**35.13**	**35.13**	**34.90**	**34.96**	**35.12**
	CHRF3	59.43	**59.53**	**59.62**	**59.72**	**59.95**	**59.48**	**59.73**	59.92
200K lines	BLEU	33.88	33.70	33.85	33.70	33.51	33.69	33.52	33.39
	TER	45.90	46.33	46.43	46.14	46.80	46.40	46.46	46.62
	METEOR	34.94	**35.23**	**35.00**	**35.20**	**35.06**	**35.04**	**34.96**	**35.08**
	CHRF3	59.56	**60.03**	59.51	**59.95**	**59.93**	**59.61**	59.49	**59.91**
NEWS									
100K lines	BLEU	26.49	**26.76**	**26.81***	**26.59**	**26.75**	**26.77***	**26.73**	**26.75**
	TER	54.19	54.36	54.39	54.46	54.49	54.40	54.71	54.84
	METEOR	30.21	**30.48***	**30.45***	**30.51***	**30.65***	**30.54***	**30.51***	**30.64***
	CHRF3	51.78	**52.35***	**52.30***	**52.46***	**52.69***	**52.34***	**52.46***	**52.70***
200K lines	BLEU	26.44	**26.80***	**26.85***	**26.77***	**26.61**	**26.79***	**26.81***	**26.66**
	TER	54.35	**54.34**	54.37	54.43	54.70	54.40	54.67	54.67
	METEOR	30.12	**30.51***	**30.49***	**30.58***	**30.60***	**30.48***	**30.55***	**30.63***
	CHRF3	51.67	**52.39***	**52.35***	**52.55***	**52.60***	**52.25***	**52.57***	**52.69***

Table 6.7: Results of the models built with different sizes of INR-selected hybrid data. The results in bold indicate an improvement over INR. The asterisk means the improvement is statistically significant at p=0.01.

			hybr	batch			online		
		FDA $_{\gamma=1}$	hybr	$\gamma =$ 0.75	$\gamma =$ 0.50	$\gamma =$ 0.25	$\gamma =$ 0.75	$\gamma =$ 0.50	$\gamma =$ 0.25
BIO									
100K lines	BLEU	33.68	**33.86**	33.56	33.48	33.30	33.39	33.41	33.45
	TER	45.97	46.20	46.51	46.49	46.93	46.33	46.58	46.70
	METEOR	34.71	**35.22***	**34.97**	**35.09***	**34.89**	**34.97**	**35.00**	**35.11***
	CHRF3	59.24	**59.89***	**59.48**	**59.62**	**59.59**	**59.45**	**59.55**	**59.83***
200K lines	BLEU	33.96	33.94	33.75	33.86	32.96	33.93	33.54	33.33
	TER	45.64	46.09	46.17	46.14	47.04	45.99	46.26	46.77
	METEOR	35.01	**35.29**	**35.08**	**35.08**	34.96	35.01	**35.09**	34.98
	CHRF3	59.56	**59.89**	**59.58**	**59.79**	**59.77**	**59.63**	**59.85**	**59.65**
500K lines	BLEU	33.75	**33.90**	33.72	33.74	33.13	33.75	**33.95**	33.46
	TER	45.92	46.34	46.32	46.13	46.90	46.12	46.14	46.85
	METEOR	34.92	**35.12**	**35.15**	**35.16**	34.87	**35.14**	**35.11**	**34.93**
	CHRF3	59.57	**59.80**	59.84	59.83	59.61	**59.83**	**60.00***	**59.77**
NEWS									
100K lines	BLEU	26.49	**26.71**	**26.58**	**26.61**	**26.60**	**26.51**	**26.59**	**26.54**
	TER	54.21	54.31	54.34	54.69	54.94	54.26	54.38	54.80
	METEOR	30.21	**30.43***	**30.36***	**30.43***	**30.51***	**30.33***	**30.49***	**30.49***
	CHRF3	51.80	**52.30***	**52.09***	**52.37***	**52.52***	**52.06***	**52.41***	**52.49***
200K lines	BLEU	26.55	**26.78***	**26.62**	**26.66**	**26.65**	**26.77***	**26.72**	**26.80***
	TER	54.17	54.42	54.36	54.52	54.70	54.29	54.48	54.55
	METEOR	30.24	**30.51***	**30.37***	**30.49***	**30.55***	**30.45***	**30.54***	**30.61***
	CHRF3	51.89	**52.41***	**52.14***	**52.49***	**52.55***	**52.35***	**52.56***	**52.75***
500K lines	BLEU	26.40	**26.78***	**26.70***	**26.92***	**26.73**	**26.68***	**26.94***	**26.98***
	TER	54.47	**54.38**	**54.29**	**54.37**	**54.63**	**54.13***	54.35	54.56
	METEOR	30.10	**30.51***	**30.44***	**30.59***	**30.59***	**30.42***	**30.58***	**30.68***
	CHRF3	51.71	**52.38***	**52.29***	**52.57***	**52.66***	**52.19***	**52.58***	**52.81***

Table 6.8: Results of the models built with different sizes of FDA-selected hybrid data. The results in bold indicate an improvement over FDA. The asterisk means the improvement is statistically significant at p=0.01.

6.8 using FDA. As in this set of experiments we use one baseline (i.e. the results of models fine-tuned with authentic data only), the numbers in bold indicate improvements when compared to this baseline, and statistically significant improvements at p=0.01 are marked with one asterisk.

The results show that increasing the size of the candidate pool is beneficial. We see that most of the scores are better than the model fine-tuned with only authentic data. The tables are structured in three subtables showing the results of *hybr*, *batch* and *online* approaches (the last two are also split into different columns according to the values of γ).

The *hybr* approach has the advantage of using a single selected pool, whereas *batch* or *online* approaches consist of two independent executions (one on the authentic and one synthetic parallel sets). This causes the final results to be an approximation as the *n*-grams of the test set that are present in both authentic and synthetic sets would follow a distorted process. For example, in INR this approach causes each *n*-gram to have a different threshold to be considered frequent in INR (more occurrences of *n*-grams can be found if they do not exceed the threshold in each execution) or not to decay as much as they should in FDA (a similar problem of ParFDA). Nonetheless, the three approaches achieve comparable results. In addition, in the *batch* or *online* approaches, we do not observe an optimal value of γ. Although expecting that higher values of γ should achieve better results (higher presence of authentic sentence) we also see experiments in which using $\gamma = 0.25$ achieve the best results. For example, using INR, in the 100K row in the NEWS subtable in Table 6.8, both for *batch* and *online*, the best results are found in the $\gamma = 0.25$ columns.

We are aware that the combinations of authentic and synthetic data may include duplicated sentences in the target side. For this reason, in Table 6.9 we present the percentage of unique target-side sentences in the training data of each model. For example, a 90% selection of unique sentences in a set of 100K would indicate that for 10K authentic selected-sentences their synthetic counterparts are present (or vice

		hybr	batch $\gamma =$ 0.75	batch $\gamma =$ 0.50	batch $\gamma =$ 0.25	online $\gamma =$ 0.75	online $\gamma =$ 0.50	online $\gamma =$ 0.25
INR								
BIO	100K	90% (56% auth)	93%	90%	93%	94%	92%	93%
	200K	90% (55% auth)	92%	90%	92%	93%	92%	93%
	500K	-	-	-	-	-	-	-
NEWS	100K	92% (64% auth)	94%	92%	94%	96%	95%	96%
	200K	91% (62% auth)	93%	91%	94%	95%	93%	95%
	500K	-	-	-	-	-	-	-
FDA								
BIO	100K	90% (53% auth)	94%	90%	93%	95%	94%	95%
	200K	90% (52% auth)	93%	89%	93%	95%	93%	94%
	500K	87% (51% auth)	91%	86%	90%	92%	90%	92%
NEWS	100K	93% (58% auth)	95%	93%	95%	97%	96%	96%
	200K	91% (58% auth)	93%	94%	91%	94%	94%	95%
	500K	88% (57% auth)	90%	92%	88%	92%	92%	93%

Table 6.9: Number of unique sentences in the target-side of the training data.

versa). In the table, we observe that all the percentages are above 88% (regardless of TA, domain, and the technique used to combine the authentic and synthetic data), so the number of duplicates is not too high.

In addition, in the *hybr* column we include in bracket, the percentage of sentences that are authentic (in the other columns the ratio of authentic is given by the value of γ). We see that the number of authentic sentences is slightly above half (between 50% and 65%) implying that there is no preference towards a particular type.

We also present in Table 6.10 a sentence translated by the different models explored in this section to show the effect of adding synthetic sentences in the training data. First, we observe that models trained with hybrid data alter the order of the *n*-gram "a policeman was injured" and "hospital information".

Additionally, in the German sentence, the word "nach" (literally "after") has the meaning of "according to" in this context (as seen in the reference). Despite that, the output generated by INR and FDA is the incorrect word (in this context) "after" whereas hybrid models correctly produce "according to" (we indicate in bold these *n*-grams as they are different when compared to default executions of TA). The

Sentence	nach Krankenhausangaben wurde ein Polizist verletzt.
Reference	according to statements released by the hospital, a police officer was injured.
BASE13	a police officer was injured after hospital information.
INR (auth)	a policeman was injured after hospital information.
INR hybr	**according to** hospital information, a policeman was injured.
INR batch $\gamma = 0.75$	**according to** hospital information, a policeman was injured.
INR batch $\gamma = 0.5$	**according to** hospital information, a policeman was injured.
INR batch $\gamma = 0.25$	**according to** hospital information, a policeman was injured.
INR online $\gamma = 0.75$	**according to** hospital information, a policeman was injured.
INR online $\gamma = 0.50$	**according to** hospital information, a policeman was injured.
INR online $\gamma = 0.25$	after hospital information, a policeman was injured .
FDA (auth)	a policeman was injured after hospital information.
FDA hybr	**according to** hospital information, a policeman was injured.
FDA batch $\gamma = 0.75$	**according to** hospital information, a policeman was injured.
FDA batch $\gamma = 0.5$	**according to** hospital information, a policeman was injured.
FDA batch $\gamma = 0.25$	**according to** hospital information, a policeman was injured.
FDA online $\gamma = 0.75$	**according to** hospital information, a policeman was injured.
FDA online $\gamma = 0.50$	**according to** hospital information, a policeman was injured.
FDA online $\gamma = 0.25$	**according to** hospital information, a policeman was injured.

Table 6.10: Comparison of outputs of the NMT models (100K lines) with hybrid data retrieved from INR and FDA following different approaches.

only exception is INR $\gamma = 0.25$, more influenced by the seed $test_{trg}$ (i.e. sentence "a police officer was injured after hospital information.") which does not contain the n-gram "according to".

In the training data, the only sentence containing the n-gram "nach Krankenhausangaben" is the authentic sentence presented in the fourth row of Table 6.3 (selected by every execution of TA), and as we have said the authentic pair does not correspond to an accurate translation. We also searched for other sentences starting with the word "nach" which are translated as "according to" in the target side, and we find that the majority of these sentences are synthetic. For example, one of the selected synthetic pairs is ("nach Ansicht dieser Ansicht Ausgaben auszugeben.", "indeed, according to this view, spending is spending."), which is the example included in the second row of Table 6.3). This again shows how having a different rephrasing in the source side increases the chances of selecting new sentences that improve the models.

The example presented in Table 6.10 is not the only one in which models containing synthetic sentences in the training data produce correct translations, on which the BASE13 model or those fine-tuned with data from TA do not. Another example is the sentence "Tanzfreudiger Nachwuchs gesucht" (in the reference: "Dance-crazy youths wanted"). This sentence is translated as "looking for a Happy New Year's Eve" by BASE13 and the models fine-tuned with INR or FDA. However, when using the hybrid training set, the translation is closer to the reference "looking for Dancing Dance".

6.6 Conclusions and Future Work

In this chapter, we have explored different uses of synthetic data to improve the models when fine-tuned with data selected from TA. The experiments carried out revealed that despite the artificially-generated data being imperfect, it is useful to improve the models.

First, we have explored models fine-tuned with synthetic data only. Although manifesting smaller improvements than those fine-tuned with authentic data, we discovered that the use of back-translated data has several benefits. As the source-side constitutes a different rephrasing, it increases the chances of the sentence pair being selected by TA. Moreover, we found examples of sentence pairs in which the artificial source-side represents a more accurate translation of the target side than that found in the authentic sentence pair.

Furthermore, we have explored a different pipeline in which TA can be used. In particular, we have incorporated the approximated target-side (a translation of the test set using a general-domain MT engine) as the seed of TA. Although using this $test_{trg}$ in isolation to retrieve sentences is not always beneficial, the combination of executions using both the test set and the $test_{trg}$ can lead to improvements.

Finally, we combined all the techniques proposed in the chapter to fine-tune models with both authentic and back-translated sentences. This has been achieved

by following three different approaches: (i) combining the data sets before the selection; (ii) concatenating the TA-selected authentic and synthetic data; and (iii), concatenating the TA-selected authentic data and the synthetic set (retrieved by finding overlaps of n-grams in the authentic target-side).

As in these experiments we have only investigated with one authentic set and one synthetic set, in the future we want to explore whether the results improve if diverse synthetic sets are used, i.e. multiple back-translated sentences, using different models, for each target-side sentence. These models can be built using a different set of data, different configurations, or different paradigms. For example, Poncelas et al. (2019c) showed that combining back-translated data generated form an SMT and an NMT model can be more beneficial to train MT models than using only one approach. Moreover, we want to consider other procedures for combining the outputs of TA. This is also applicable to the creation of the approximated target side. By building many of them using different MT engines we can retrieve several TA_{trg} variants. This expands the number of alternatives to explore as there would be more methods to concatenate the outputs of TA using different seeds.

Chapter 7

Conclusions and Future Work

In this thesis we have explored the impact of using different amounts of data on the two leading MT approaches. We have seen that in SMT, adding more data does not necessarily lead to better results. In contrast, in NMT, the inclusion of high-quality data tends to be beneficial and the best performance is seen when a larger set of sentences is used.

Nonetheless, both SMT and NMT approaches have also been shown to perform better when using less, but more relevant data to the test set. We have used two TAs INR and FDA to identify relevant sentences. The experiments showed that in SMT, using a small subset retrieved by the TAs was enough to build models that perform better than those built with the complete training data (or a randomly-sampled subset).

As INR and FDA were designed to work in SMT, the performance on NMT approaches, which require more data to perform well, was hitherto unexplored. In this work we demonstrated that these data-selection algorithms can also be beneficial in NMT.

In addition, we have proposed different methods to improve the TAs by: (i) modifying the selection criterion (with alignment entropies); (ii) augmenting the number of candidates that the TAs can select (using synthetic sentences); and (iii) also considering the target side to select sentences (using an artificial target-side

seed).

7.1 General Recommendations

We recommend the use of context-dependent TA such as INR and FDA over the techniques such as TFIDF that retrieve data by comparing each sentence individually without promoting the variability of n-grams.

In terms of the values of the hyperparameters of TAs, we have seen that those configuration that have a steeper coverage curve (e.g. smaller decay factor in FDA) are preferable. Note that some parameters can cause the models to perform better, but it also has an impact on the training time. For example, higher orders of n-grams tend to achieve better results, but this would also increase the execution time. In general, both the number of selected sentences and order of n-grams are the most relevant factors that affect the execution time in FDA.

In the case of INR, the value of the threshold t also has an impact on the execution time as it influences the number of sentences selected. The higher the value of t, the more sentences that will be retrieved (see Table 3.4), but it also causes the execution time to increase. On the other hand if the value of t is too small, it may not retrieve enough sentences. The optimal value of t highly depends on the data used for initialization (in terms of size and n-gram variety), for this reason we suggest to execute preliminary experiments using a *dev set*.

In this work we have not determined how many sentences should be retrieved so the MT model achieves the best performance. In our experience, just a small subset of data is enough for achieving improvements. However, the selected data should at least achieve the plateau when the highest coverage possible is reached.

In general, what we find to have the highest positive impact is the inclusion of additional sentence pairs in the candidate pool. This augmentation is always favourable as the TAs can select those that are truly relevant. On the top of that, even when parallel sentences are not available, the use of artificially-created sen-

tences are also helpful.

7.2 Research Questions Revisited

As a conclusion of this work, we summarize the outcomes of the experiments performed to answer each RQ individually:

1. **RQ1: How can we tailor data-selection algorithms to be most effective in combination with NMT?**

   In Chapter 4 we have used the data selected by the TA both to build NMT models from scratch and to fine-tune a general-domain model. The experiments performed showed that: (i) when BPE is not applied, TA-extracted data are more useful to build models from scratch as the vocabulary of the test set is included in the most frequent words of the selected data; and (ii) when BPE is applied, the vocabulary is not a limitation and so by just executing a single extra epoch with selected data, the models can be improved.

2. **RQ2: Can word-alignment information be useful for improving state-of-the-art TAs?** The word-alignment entropies of the n-grams proposed in Chapter 5 are indicators of how difficult it is to generate their translations. The entropies were used as an input of the TAs (as values of their parameters) so each n-gram is penalized differently, depending on how difficult they are to be translated. N-grams with higher entropies indicate that they are more difficult to be translated, and so the decay should be smaller. In our experiments, we showed that this extension can benefit SMT models but it has only a small influence on NMT. A disadvantage of this method is that increasing the number of sentences containing n-grams with multiple (evenly-distributed) possible translations can also increase the chance that a model translates the word incorrectly.

3. **RQ3: Can the use of synthetic sentences improve the performance of MT models when used in combination with TAs?**

   In Chapter 6 we explored the performance of NMT models fine-tuned with back-translated sentences retrieved from TAs. We showed that the models fine-tuned with these data (synthetic sentences only) can also boost the general-domain models and achieve similar performance to those adapted with authentic sentences. Accordingly, we proposed three ways of using TAs with authentic and synthetic datasets: (i) combining both datasets before the selection; (ii) combining the TA-selected authentic sentences with sentences extracted from the synthetic set; and (iii) combining the TA-selected authentic sentences with sentences extracted from monolingual target-side data (which are back-translated afterwards). The sentences obtained via these procedures were used to fine-tune NMT models, and the results revealed that all three of them (which have a similar performance) enabled the model to increase the quality of the translation.

   Another use of artificial sentences in TAs is to use them as a seed. In Chapter 6 we used this approach to select sentences by considering the target side of the training sentences. This implies searching for the n-grams of an MT-translated test set (instead of the original). The experiments carried out showed that selecting sentences from the target side can also boost NMT models. The results when TAs extract sentences finding n-grams in the source side (German) were similar to those when finding n-grams in the target side (English).

7.3 Future Work

There are several ways in which this work can be expanded. One limitation of this work is that, although the techniques explored are language-independent, the experiments we have run are only in the German-to-English direction. In future, we want to investigate these methods using different language pairs and different

directions.

Furthermore, there are other specific lines of research that can be contemplated that would constitute useful extensions of this thesis.

7.3.1 Generalisation Capabilities of TAs

		BASE	CED	TFIDF	INR	FDA
BIO						
100K lines	BLEU	22.62	18.91	21.54	23.77	24.15
	TER	57.98	60.71	58.06	56.93	55.35
	METEOR	28.26	24.86	27.68	29.39	29.24
	CHRF3	51.40	48.00	50.44	52.67	52.59
NEWS						
100K lines	BLEU	18.21	13.79	15.95	16.74	16.43
	TER	66.88	67.55	65.41	64.98	65.1
	METEOR	26.01	21.78	23.99	24.79	24.19
	CHRF3	47.15	41.51	43.72	44.63	44.18

Table 7.1: Results of the models built with selected data using an in-domain set.

Although the primary aim of this thesis is to adapt MTs models to a particular test *document*, a future line of research would be to explore whether TAs can be used to adapt models towards a particular *domain* instead, using an in-domain set as seed. This would demonstrate that TAs have generalization capabilities beyond what has been shown in this thesis.

In Table 7.1, we present some preliminary research in this direction. Assuming we do not have the test set to hand, we (i) executed CED, TFIDF, INR and FDA techniques using an in-domain set as seed to extract sentences (100K lines), and (ii) built SMT models using those extracted sentences. As in-domain data we used the European Medicines Agency (EMEA)[1] (Tiedemann, 2009) dataset (361K sentence pairs) for the BIO test set and the *rapid2016*[2] data set (1.3M sentence pairs) for the NEWS test set.

In the table we see that INR and FDA are able to outperform other data-selection

[1] `http://opus.nlpl.eu/EMEA.php`
[2] `https://tilde.com/`

techniques for domain adaptation such as the more commonly used CED. Note too that when compared to the results of the BASE model, they can achieve better performance depending on the seed used (such as for the BIO test set).

This provides preliminary evidence that INR and FDA have generalization capabilities beyond specific test documents, although of course we have only performed experiments on one language pair and on two domains. For a more definitive answer, more experiments on further language pairs and domains would need to be conducted.

Further work that can be done in this direction includes exploring which characteristics of the seed (such as the size, number of n-grams, etc.) would have the best impact when used with TAs. Of course, given its standing as the state-of-the-art paradigm in MT today, all these issues ought to be explored in NMT, too.

7.3.2 Exploration of Configuration of TAs

Additionally, the configurations of both TAs have been the same across the whole thesis. Section 5.1 showed that different configurations have an impact in SMT. However, the configurations explored involve changing only one parameter at a time. Another issue that was raised in Section 4.2.2 is whether these methods work better using BPE in the seed and candidates before selection is performed.

Similarly, different configurations of the models could be explored in NMT. For example, should the general-domain model (BASE) be close convergence before fine-tuned with TA-selected data? How many iterations of fine-tuning should be executed with TA-selected data for optimal performance?

Regarding the alignment probabilities proposed in Chapter 5, we have seen that they can have a positive impact in SMT when used in the decay exponent of FDA. However, the entropies computed are in the $(0, 1)$ range whereas the decay exponent can have any positive value. We would like to know whether using higher values of entropies could be beneficial. In addition we propose to explore alternative methods of computing the alignment entropies.

7.3.3 Augmentation of Candidate Pool

In the last chapter, which explores the use additional artificial sentences, there are also directions for further research. First of all, the generated sentences are obtained from a single NMT model. In the future we want to explore the creation of synthetic data using MT models with different configurations, different sets of data or even following other approaches such as SMT. These models can be explored independently or also can be combined to build different artificial source sides from each target-side sentence. This can also be applied when the artificial test set is built: will the use of multiple approximated target side as seed be beneficial? Finally we want to consider other procedures for combining the outputs of TA_{src} and TA_{trg}, or TA_{auth} and TA_{synth}, such as union or intersection.

Bibliography

Ambati, V., Vogel, S., and Carbonell, J. G. (2011). Multi-strategy approaches to active learning for statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*, page 122–129, Xiamen, China.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, San Diego, USA.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, Ann Arbor, Michigan.

Biçici, E. (2013). Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 78–84, Sofia, Bulgaria.

Biçici, E. (2016). Parfda for instance selection for statistical machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 252–258, Berlin, Germany.

Biçici, E., Liu, Q., and Way, A. (2014). Parallel FDA5 for fast deployment of accurate statistical machine translation systems. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 59–65, Baltimore, Maryland, USA.

Biçici, E., Liu, Q., and Way, A. (2015). ParFDA for fast deployment of accurate statistical machine translation systems, benchmarks, and statistics. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 74–78, Lisbon, Portugal.

Biçici, E. and Yuret, D. (2011). Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland.

Biçici, E. and Yuret, D. (2015). Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):339–350.

Biçici, M. E. (2011). *The Regression Model of Machine Translation*. PhD thesis, Koç University.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisboa, Portugal.

Britz, D., Le, Q., and Pryzant, R. (2017). Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 118–126, Copenhagen, Denmark.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Cho, K., van Merrienboer, 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., Dabre, R., and Kurohashi, S. (2017). An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 385–391, Vancouver, Canada.

Chu, C. and Wang, R. (2018). A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, USA.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Deep Learning and Representation Learning Workshop, NIPS 2014*, Montreal, Canada.

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 (Volume 2: Short Papers)*, page 176–181, Portland, Oregon.

Di Gangi, M. A., Bertoldi, N., and Federico, M. (2017). FBK's participation to the English-to-German News Translation Task of WMT 2017. In *Proceedings of the Second Conference on Machine Translation*, pages 271–275, Copenhagen, Denmark.

Dowling, M., Lynn, T., Poncelas, A., and Way, A. (2018). SMT versus NMT: Preliminary comparisons for Irish. In *Technologies for MT of Low Resource Languages (LoResMT 2018)*, page 12, Boston, USA.

Dyer, C., Chahuneau, V., and Smith, N. (2013). A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pages 644–648, Atlanta, Georgia, USA.

Dzendzik, D., Poncelas, A., Vogel, C., and Liu, Q. (2017). ADAPT centre cone team at IJCNLP-2017 task 5: A similarity-based logistic regression approach to multi-choice question answering in an examinations shared task. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 67–72, Taipei, Taiwan.

Eck, M., Vogel, S., and Waibel, A. (2005a). Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MT Summit X*, pages 227–234, Phuket, Thailand.

Eck, M., Vogel, S., and Waibel, A. (2005b). Low cost portability for statistical machine translation based on n-gram frequency and TF-IDF. In *2005 International Workshop on Spoken Language Translation, IWSLT*, pages 61–67, Pittsburgh, PA, USA.

Eetemadi, S., Lewis, W., Toutanova, K., and Radha, H. (2015). Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

Freitag, M. and Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., and Casacuberta, F. (2012). Does more data always yield better translations? In *Proceedings of the*

13th Conference of the European Chapter of the Association for Computational Linguistics, pages 152–161, Avignon, France.

Gülçehre, Ç., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Haffari, G., Roy, M., and Sarkar, A. (2009). Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423, Boulder, Colorado.

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.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 133–142, Budapest, Hungary.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–1780.

Khadivi, S. and Ney, H. (2005). Automatic filtering of bilingual corpora for statistical machine translation. In *International Conference on Application of Natural Language to Information Systems*, pages 263–274.

Khayrallah, H., Kumar, G., Duh, K., Post, M., and Koehn, P. (2017). Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 20–25, Taipei, Taiwan.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 67–72, Vancouver, Canada.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 181–184, Detroit, MI, USA.

Kobus, C., Crego, J., and Senellart, J. (2017). Domain control for neural machine translation. In *Proceedings of Recent Advances in Natural Language Processing*, page 372–378, Varna, Bulgaria.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, USA.

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 SMT. In *Proceedings of 45th annual meeting of the ACL on interactive poster & demonstration sessions*, 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., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of Conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series*, pages 48–54, Edmonton, Canada.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710.

Li, X., Zhang, J., and Zong, C. (2018). One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 910–917, Miyazaki, Japan.

Liu, C.-H., Groves, D., Hayakawa, A., Poncelas, A., and Liu, Q. (2017a). Understanding meanings in multilingual customer feedback. In *Proceedings of First Workshop on Social Media and User Generated Content Machine Translation (Social MT 2017)*, Prague, Czech Republic.

Liu, C.-H., Moriya, Y., Poncelas, A., and Groves, D. (2017b). IJCNLP-2017 Task 4: Customer Feedback Analysis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 26–33, Taipei, Taiwan.

Lo, C.-k., Chen, B., Cherry, C., Foster, G., Larkin, S., Stewart, D., and Kuhn, R. (2017). NRC Machine Translation System for WMT 2017. In *Proceedings of the Second Conference on Machine Translation*, pages 330–337, Copenhagen, Denmark.

Lopez, A. D. (2008). *Machine Translation by Pattern Matching*. PhD thesis, University of Maryland, College Park, MD, USA.

Lu, Y., Huang, J., and Liu, Q. (2007). Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350, Prague, Czech Republic.

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79, Da Nang, Vietnam.

Mandal, A., Vergyri, D., Wang, W., Zheng, J., Stolcke, A., Tur, G., Hakkani-Tur, D., and Ayan, N. F. (2008). Efficient data selection for machine translation. In *Spoken Language Technology Workshop, 2008. SLT 2008*, pages 261–264, Goa, India.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mohit, B. and Hwa, R. (2007). Localization of difficult-to-translate phrases. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 248–255, Prague, Czech Republic.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224, Uppsala, Sweden.

Och, F. (2003). Minimum error rate training in statistical machine translation. In *ACL-2003: 41st Annual Meeting of the Association for Computational Linguistics, Proceedings*, pages 160–167, Sapporo, Japan.

Och, F. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ozdowska, S. and Way, A. (2009). Optimal Bilingual Data for French-English PB-SMT. In *Proceedings of the 13th Annual Meeting of the European Association for Machine Translation*, pages 96–103, Barcelona, Spain.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Parcheta, Z., Sanchis-Trilles, G., and Casacuberta, F. (2018). Data selection for NMT using infrequent N-gram recovery. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, page 219–227, Alacant, Spain.

Passban, P. (2017). *Machine translation of morphologically rich languages using deep neural networks*. PhD thesis, Dublin City University.

Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2018a). Data selection with feature decay algorithms using an approximated target side. In *15th International Workshop on Spoken Language Translation (IWSLT 2018)*, pages 173–180, Bruges, Belgium.

Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2019a). Adaptation of machine translation models with back-translated data using transductive data selection methods. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France.

Poncelas, A., de Buy Wenniger, G. M., and Way, A. (2019b). Transductive data-selection algorithms for fine-tuning neural machine translation. In *The 8th Workshop on Patent and Scientific Literature Translation (PSLT 2019)*, Dublin, Ireland.

Poncelas, A., Maillette de Buy Wenniger, G., and Way, A. (2017). Applying n-gram alignment entropy to improve feature decay algorithms. *The Prague Bulletin of Mathematical Linguistics*, 108(1):245–256.

Poncelas, A., Maillette de Buy Wenniger, G., and Way, A. (2018b). Feature decay algorithms for neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 239–248, Alacant, Spain.

Poncelas, A., Popovic, M., Shterionov, D., de Buy Wenniger, G. M., and Way, A. (2019c). Combining SMT and NMT back-translated data for efficient NMT. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 922–931, Varna, Bulgaria.

Poncelas, A., Sarasola, K., Dowling, M., Way, A., Labaka, G., and Alegria, I. (2019d). Adapting NMT to caption translation in Wikimedia Commons for low-resource languages. In *35th International Conference of the Spanish Society for Natural Language Processing (SEPLN 2019)*, Bilbao, Spain.

Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G. M., and Passban, P. (2018c). Investigating backtranslation in neural machine translation. In *21st Annual Conference of the European Association for Machine Translation*, pages 249–258, Alacant, Spain.

Poncelas, A. and Way, A. (2019). Selecting Artificially-Generated Sentences for Fine-Tuning Neural Machine Translation. In *Proceedings of the 12th International Conference on Natural Language Generation*, Tokyo, Japan.

Poncelas, A., Way, A., and Sarasola, K. (2018d). The ADAPT System Description for the IWSLT 2018 Basque to English Translation Task. In *International Workshop on Spoken Language Translation*, pages 72–82, Bruges, Belgium.

Poncelas, A., Way, A., and Toral, A. (2016). Extending feature decay algorithms using alignment entropy. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 170–182, Seville, Spain. Springer.

Popovic, 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.

Salton, G. and Yang, C.-S. (1973). On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016c). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725, Berlin, Germany.

Silva, C. C., Liu, C.-H., Poncelas, A., and Way, A. (2018). Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231, Brussels, Belgium.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, Montreal, Canada.

Taghipour, K., Afhami, N., Khadivi, S., and Shiry, S. (2010). A discriminative approach to filter out noisy sentence pairs from bilingual corpora. In *Proceedings of 5th International Symposium on Telecommunications (IST 2010)*, pages 537–541, Tehran, Iran.

Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Tillmann, C. (2004). A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, USA.

van der Wees, M., Bisazza, A., and Monz, C. (2017). Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark.

Vanmassenhove, E., Hardmeier, C., and Way, A. (2018). Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium.

Vanmassenhove, E., Moryossef, A., Poncelas, A., Way, A., and Shterionov, D. (2019). ABI Neural Ensemble Model for Gender Prediction Adapt Bar-Ilan Submission for the CLIN29 Shared Task on Gender Prediction. In *Computational Linguistics of the Netherlands CLIN29*, Groningen, The Netherlands.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Wang, L., Wong, D. F., Chao, L. S., Lu, Y., and Xing, J. (2014). A systematic comparison of data selection criteria for SMT domain adaptation. *The Scientific World Journal*.

Wang, R., Utiyama, M., Finch, A., Liu, L., Chen, K., and Sumita, E. (2018). Sentence selection and weighting for neural machine translation domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1727–1741.

Wang, R., Utiyama, M., Liu, L., Chen, K., and Sumita, E. (2017). Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark.

Yepes, A. J., Névéol, A., Neves, M., Verspoor, K., Bojar, O., Boyer, A., Grozea, C., Haddow, B., Kittner, M., Lichtblau, Y., et al. (2017). Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247, Copenhagen, Denmark.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas.