Universität des Saarlandes
Universidad del País Vasco/Euskal Herriko Unibertsitatea

Domain Adaptation for Multilingual Neural Machine Translation

Master's Thesis

submitted in fulfillment of the degree requirements of the
MSc in Language Science and Technology at **Saarland University**
as part of the
European Masters Program in Language and Communication Technologies

Author
Ádám Csaba Varga

Advisors
Dr. Cristina España-Bonet
Prof. Dr. Josef van Genabith
Dr. Gorka Labaka

August 24, 2017

Contents

EIDESTATTLICHE ERKLÄRUNG

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

DECLARATION

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Saarbrücken, August 24, 2017

Ádám Csaba Varga

Abstract

Neural machine translation (NMT) is currently considered the state-of-the-art for language pairs with vast amounts of parallel data. In this thesis project, we utilize such systems to provide translations between four languages in the psychology domain, where the biggest challenge is posed by in-domain data scarcity. Therefore, the emphasis of the research is laid on exploring domain adaptation methods in this scenario. We first propose a system for automatically building in-domain adaptation corpora by extracting parallel sentence pairs from comparable articles of *Wikipedia*. To this end, we use supervised classification and regression methods trained on NMT context vector similarities and complementary textual similarity features. We find that the best method for our purposes is a regression model trained on continuous similarity labels. We rerank the extracted candidates by their similarity feature averages and use the top-N partitions as adaptation corpora. In the second part of the thesis we thoroughly examine multilingual domain adaptation by transfer learning with respect to the adaptation data quality, size, and domain. With clean parallel in-domain adaptation data we achieve significant improvements for most translation directions, including ones with no adaptation data, while the automatically extracted corpora prove beneficial mostly for language pairs with no clean in-domain adaptation set. Particularly in these latter cases, the combination of the two adaptation corpora yields further improvements. We also explore the possibilities of reranking N-best translation lists with in-domain language models and similarity features. We conclude that adapted systems produce candidates that can result in a higher improvement in translation performance than the ones of unadapted models, and that remarkable improvements can be achieved by similarity-based reranking methods.

Introduction

Data-driven machine translation (MT) systems rely on the existence and availability of large-scale parallel corpora for language pairs between which the system should be able to provide automatic translations. Recently, the focus of MT has shifted from statistical machine translation (SMT) approaches to neural machine translation (NMT), and currently such systems are considered to be the state-of-the-art for well-resourced language pairs due to their superior translation quality compared to previous architectures [Bojar et al., 2016].

NMT systems, however, require more parallel data in order to achieve a significant improvement in translation performance compared to classical SMT setups [Zoph et al., 2016]. This usually does not pose a problem for resource-rich language pairs for which large amounts of parallel text is often available. On the other hand, NMT translation quality between language pairs that have less such data (usually referred to as low-resourced or under-resourced language pairs) is still below that of the SMT systems.

The CLUBS project[1] aims at machine translation between language pair combinations of German (*de*), English (*en*), Spanish (*es*), and French (fr). Due to the nature of the project, the candidate texts lie in the domain of psychology. Depending on the language pair, the availability of parallel corpora and the different domains covered by them varies largely. Some language pairs, such as *en–es* have large amounts of data available including texts in the psychology domain as well as out-of-domain parallel corpora. Other language pairs have significant amounts of parallel text, but only for domains that do not include psychology (e. g. *en–*fr). While for language pairs excluding *en* (e. g. *de–es*) there exists out-of-domain parallel data, in our research we choose not to include such corpora for training purposes in order to study these

[1] `https://www.CLUBS-project.eu/en/`

zero-shot translation directions, which pose the biggest challenge in MT scenarios.

The focus of the thesis project is to explore the possibilities of using NMT within the CLUBS project framework to address two main problems: (in-domain) data scarcity and multilinguality. In particular, we lay the emphasis on investigating the topic of domain adaptation for such systems. To this end, the effect of applying transfer learning to pre-trained general-domain models is studied in different scenarios. While this technique has been researched previously [Chu et al., 2017, Freitag & Al-Onaizan, 2016], in this thesis project we aim to further explore questions about the effects of adaptation data quality and the effect of adaptation on under- and zero-resourced translation directions. In addition to the transfer learning method, we also conduct experiments on selecting the best translation candidates by means of language model (LM) and similarity feature reranking.

As there are in-domain parallel corpora available within the CLUBS project, we investigate the effect of clean, strictly domain-specific adaptation data on our system. However, since such parallel data is not available for all language pairs involved in the project, we propose a method for automatically extracting additional in-domain parallel corpora from *Wikipedia*[2] for any of the six language pairs. We run our domain adaptation experiments using these automatically created corpora, and check how a relatively less clean data set affects the translation quality in the NMT framework. We also investigate whether and how the system can benefit from the combination of this and the high-quality parallel data. During our investigations, we lay special attention to examining zero-resourced translation scenarios.

Outline

This thesis is organized as follows. Chapter 1 summarizes the background of NMT and discusses domain adaptation techniques for such systems, while shortly describing similar approaches within SMT frameworks. Chapter 2 describes the resources used in this thesis project and introduces our proposed method for in-domain parallel sentence extraction from *Wikipedia*. Chapter 3 discusses the experiments in domain adaptation by transfer learning in NMT systems using the available parallel and automatically extracted adaptation corpora, as well as the combination of the two. This chapter

[2] `https://www.wikipedia.org/`

also describes the experiments conducted on reranking of N-best lists of translation candidates in order to select the best-performing output sentences. Finally, we summarize the work carried out, and we draw conclusions and lie out possibilities for future research.

Chapter 1

Related Work

In this chapter we provide a brief introduction to NMT, laying the emphasis on the architecture used in this thesis project. We discuss the possibilities of extending NMT to the multilingual space, and introduce approaches of domain adaptation within this framework. The possibilities of solving the same problems in SMT systems are also summarized briefly, along with some further problems related to this thesis project, most importantly parallel sentence identification. The structure of Section 1.1 is loosely based on [Cho, 2015].

1.1 Machine Translation

1.1.1 Rule-Based, Statistical, and Neural Approaches

The goal of machine translation is to build systems that are capable of translating the sentences from one natural language to another. While RBMT systems make an attempt on describing the underlying rules of transforming sentences in one language to another (e. g. [Forcada et al., 2011, Mayor et al., 2011]), in the case of SMT the task is finding an appropriate mapping function between the two languages by statistical methods [Koehn, 2009]. This calls for a collection of translated sentence pairs, referred to as the parallel corpus. Phrase-based SMT systems attempt to learn the mapping function using log-linear models, operating on a set of weighted feature functions. These can include (bidirectional) phrase alignment probabilities trained

on the parallel corpus, reordering penalties and language model(s), as well as various additional features. While such systems can be expanded with neural networks to perform reranking of translation candidates [Devlin et al., 2014] or by using neural language models [Schwenk, 2007], NMT refers to learning this function with a single end-to-end neural network architecture.

In this thesis project, we focus on using NMT systems. While for well-resourced language pairs their translation quality tends to be superior to that of SMT models, they have additional properties that make them an appropriate choice for addressing our problems. Namely, it is possible to train one NMT system on multiple language pairs at the same time. Not only does this property allow for translating multilingual documents in a compact way, but it also enables translation between zero-resourced language pairs without any additional effort. Furthermore, the internal representations learned by such systems can be used to obtain language-independent embeddings of sentences in a multilingual space. As shown in Chapter 2, these can be utilized to identify parallel sentences in comparable corpora, thus acquiring additional data for adaptation purposes.

1.1.2 Neural Machine Translation Architecture

While deep neural networks are known to eliminate the need for extensive feature engineering as they are capable of learning multiple levels of abstraction via their hidden layers, the design choice for the exact network architecture is a key step and it highly depends on the task at hand. In the case of NMT, recurrent neural networks (RNN) are a common choice, as they are capable of maintaining their hidden vectors while processing sentences in a sequential fashion. This functions as a memory state that is able to model long-term dependencies that arise when processing natural language inputs. In addition to RNN-based NMT models, approaches based on convolutional neural network (CNN) architectures have proven to deliver results of a similar quality with a significant speedup in computational time [Gehring et al., 2017]. Our choice for this thesis project is the RNN-based architecture based on [Bahdanau et al., 2015] as at the beginning of this thesis project this approach was considered the state-of-the-art, supported by readily available open-source implementations.

Although simple RNNs [Elman, 1991] are theoretically adequate for learning tasks on sequential data, certain practical issues (such as the vanishing gradient problem) call

for more sophisticated solutions with more complex recurrent units. Two widely used choices are long short-term memory (LSTM) units [Hochreiter & Schmidhuber, 1997] and gated recurrent units (GRU) [Cho et al., 2014b]. Using recurrent architectures, it is possible to predict the output words given the history of the input words represented by the hidden vector.

For deep learning models, learning adequate feature representations of the data is a key element. NMT systems utilize an encoder-decoder architecture to this end. The encoder's task is converting the input words (usually represented as simple one-hot encodings of the given tokens) into continuous representations by using a weight matrix (that is to be trained to maximize the translation performance). More formally, if $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is a sentence of length n consisting of one-hot word vectors $\mathbf{x}_i$, the the continuous word representations are given by $\mathbf{u}_i = \mathbf{W}_x \cdot \mathbf{x}_i$, where $\mathbf{W}_x$ is the input embedding weight matrix. This continuous vector sequence is fed to a RNN. Its hidden state $\mathbf{h}_i$ at the ith word is characterized by $\mathbf{h}_i = \Phi_\theta(\mathbf{h}_{i-1}, \mathbf{u}_i)$, where Φ_θ is a gated unit. This way, $\mathbf{h}_n$ represents a summary of the whole sentence.

After obtaining such representations of the target sentences, the final hidden vector $\mathbf{h}_n$ is passed to the decoder stage of the NMT system. It is fed to another RNN where the hidden states $\mathbf{z}_i$ are conditioned on this vector, the decoder's previous hidden state, as well as the the previously generated output word vector $\mathbf{y}_{i-1}$ ($\Phi_{\theta'}$ is again a gated unit similar to the one used in the encoder):

$$\mathbf{z}_i = \Phi_{\theta'}(\mathbf{h}_n, \mathbf{z}_{i-1}, \mathbf{y}_{i-1}) \tag{1.1}$$

The hidden states of the decoder thus allow for assigning scores to candidate output words $\mathbf{y}_k$, depending on both the input sequence and words translated so far. This is done by a scoring function that takes the dot product of the decoder's hidden state and the candidate word vector as inputs, resulting in higher scores for more similar vectors: $score_i(k) = \mathbf{y}_k^T \mathbf{z}_i + b_k$ (b_k is a bias term). The scores are then converted into probabilities using the *softmax* function [Bridle, 1990]. The obtained probability distributions can be used for sampling output words one after another, until a certain stop sign indicating the end of the sentence is reached.

One problem of the simple encoder-decoder architecture arises when long sentences

are encountered. Since the source sentences are summarized in fixed-length context vectors, the model is usually not capable of adequately representing long sentences due to dimensionality issues. Although increasing the number of dimensions in the model could theoretically overcome this problem, the limited memory available for computation sets an upper limit to the extent this can be done. This way, alternative solutions need to be proposed in order to enable the translation of longer sentences without dramatically decreasing the system's performance.

In order to represent the source sentence as a vector where each source word has a corresponding slot, RNNs are replaced by bidirectional recurrent neural networks (BRNN). Such architectures consist of two separate RNNs that read the input in two different directions, i. e. from forward to backward and vice versa. This results in *forward* and *backward* hidden states $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$. Using pairs of hidden states from the two networks at a given position then can be viewed as the summaries of the source sentence up until that position from the beginning and the end respectively:

$$\mathbf{h}_i = \left[\overleftarrow{\mathbf{h}}_i, \overrightarrow{\mathbf{h}}_i\right] = \left[\Phi_{\theta,bw}(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{u}_i), \Phi_{\theta,fw}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{u}_i)\right] \tag{1.2}$$

This way, each hidden state pair represents the summary of the complete source sentence. We can use the concatenation of these pairs to obtain a *context vector* $\mathbf{c}$ for each sentence and feed this to the decoder:

$$\mathbf{c} = \{\mathbf{h_1}, \ldots, \mathbf{h_n}\} \tag{1.3}$$

This summary, however, is influenced by proximity effects at each given position, as RNNs' memory decays with time, and more recent words have a higher effect on the hidden states making them context dependent. Due to this property, the decoder stage needs to weigh up these state pairs, since some of them need higher attention than others.

There are various ways for tackling this problem, generally referred to as *attention mechanisms*. The two most common architectures are the additive [Bahdanau et al., 2015] and the multiplicative [Luong et al., 2015a] approaches. While the latter method

is computationally less expensive, according to [Britz et al., 2017] the additive variant delivers better results and we choose to use this approach for our thesis. An additive attention mechanism consists of a single feed-forward neural network (FFNN) that can learn to weigh the context vectors accordingly using the decoder's previous hidden state and the hidden state pair at each given position. Similarly to the output word probability distributions, the scores are converted into probabilities by computing the *softmax* function (it has to be noted here that this addition slows down the computation significantly, as the *softmax* computation is generally the slowest component of training neural networks). The attention weights α_{ij} for the ith source word at the jth decoder state are computed as shown in Equations 1.4 and 1.5. Here, $\mathbf{W}_a$ and $\mathbf{U}_a$ are the corresponding weight matrices to the previous decoder state and the current hidden vector at step i.

$$a(\mathbf{z}_{j-1}, \mathbf{h}_i) = \mathbf{v}_a \cdot \tanh(\mathbf{W}_a \cdot \mathbf{z}_{j-1} + \mathbf{U}_a \cdot \mathbf{h}_i) \tag{1.4}$$

$$\alpha_{ij} = \frac{\exp(a(\mathbf{z}_{j-1}, \mathbf{h}_i))}{\sum_{i'} \exp(a(\mathbf{z}_{j-1}, h_{i'}))}, \quad \mathbf{c}_j^{'} = \sum_i \alpha_{ij} \mathbf{h}_i \tag{1.5}$$

This way, the weighted context vector $\mathbf{c'}_j$ can be fed to the decoder at the jth time step. Using an attention mechanism overcomes the problems with longer sentences and additionally serves as a soft alignment model between source and target sentences [Bahdanau et al., 2015].

1.1.3 Solving the Out-of-Vocabulary Problem

As NMT systems use pre-defined word vectors, they are only able to opearate on a closed vocabulary (typically containing only 30 K–100 K words due to their extensive memory usage). It is possible to split the input into units smaller than words, and even to translate character sequences to word sequences reading the input side character-by-character [Ling et al., 2016]. Using subword units can overcome the issues of handling out-of-vocabulary (OOV) words during translation. Initially, these words had been handled by simple dictionary lookups [Jean et al., 2015, Luong et al., 2015b], but as it is pointed out in [Sennrich et al., 2016b], this has proven problematic for certain cases

such as compounds between languages with varying levels of morphological richness. Their suggested approach is to apply *byte pair encoding* (BPE) on unseen or rare words in order to merge certain frequent character n-grams, thus solving the problem in a more sophisticated way.

1.1.4 Multilingual Neural Machine Translation

According to [Ha et al., 2016], since the encoder architecture creates a representation of the source text in an embedding space, it is possible to include sentences from different source languages on the source side in order to obtain an embedding space that shares the common semantic traits of the involved source languages. Then the decoder can theoretically be used to translate from this shared space to any target language (although certain constraints might have to be introduced in order to facilitate this). To this end, the authors apply language specific coding on the source side and target forcing on the target side. The former step is simply done by appending the language code to source words and the latter involves appending the target language tags on the sentence level.

This observation allows for various multilingual neural machine translation (ML-NMT) scenarios that can be used to overcome the scarcity problem for under-resourced language pairs. One possibility described in the above-mentioned publication is enriching the system with monolingual data (e. g. adding *de–de* sentence pairs to an *en–de* parallel corpus), while another approach might be using additional parallel data where the source sides are the same (e. g. expanding an *en–de* system with data for fr–*de*). Both attempts result in better translation performance compared to baseline systems operating on small-scale parallel data.

An extreme case of under-resourced scenarios is when no parallel data is available for the given language pair, but the two languages occur in other available language pairs, referred to as zero-resourced translation. A simple solution can be applying an intermediate *pivot* language to overcome this issue, i. e. building two NMT systems that first translate the source language to the pivot language (e. g. *en*) for which there is available parallel data, then the pivot language gets further translated into the source language (using parallel data between the pivot and target languages). The authors of [Ha et al., 2016] attempt to build single many-to-many ML-NMT systems in order to simplify the process by using parallel data between the source and the pivot

language and the pivot and target language in an ML-NMT scenario enriching it by monolingual data for the pivot language and optionally the target language. These systems, however perform worse than the two-stage approach.

According to [Johnson et al., 2016], the attempt from *Google* for enabling zero-shot translation overcomes this problem by utilizing an incremental training method. This is achieved by adding small amounts of actual parallel data in the last few epochs of the training procedure that significantly improves the translation performance. It has to be noted, however, that in this case one cannot talk about a completely zero-resourced scenario, but the method can be a good example of refining under-resourced setups.

The proposed model in [Cheng et al., 2016] improves the two-stage pivot approach by encouraging the two networks to learn the same vector representations for words that lie in the intersection of the pivot vocabularies of source-to-pivot and pivot-to-target models through introducing a connection term to the objective function during training. If the translation scenario is not zero-resourced, a small available source-to-target corpus can be also used for the joint optimization.

In [Firat et al., 2016], an approach is proposed for ML-NMT that uses separate encoders for each source language (ultimately projecting embeddings into a common-dimensional space). The models they describe include a shared attention mechanism between languages, and they are capable of producing better results for the translation of under-resourced language pairs than single NMT systems or ML-NMT models enriched with monolingual data.

Zero-Shot Translation Directions in Statistical Machine Translation

Enabling zero-shot directions in SMT systems is only possible via pivot languages. In this case, the trivial approach is to perform the translation pipeline $source \rightarrow pivot \rightarrow target$, assuming there is parallel *source–pivot* and *pivot–parallel* data. A more sophisticated way of achieving the same goal with higher efficiency is to perform phrase table combination by triangulation [Cohn & Lapata, 2007] or by co-occurence counts [Zhu et al., 2014]. The former method merges phrases that are have identical *pivot* phrases and multiplies the posterior probabilities of such instances in order to acquire the final probability. The latter work approaches the problem by estimat-

ing the co-occurrences of *source-pivot-target* phrase pairs and computing translation probabilities from these by standard SMT training.

1.1.5 Internal Representations of the Encoder

It has been shown in [Sutskever et al., 2014] that the context vectors of NMT systems preserve the underlying semantic and syntactic structure of sentences. This research has demonstrated that context vectors corresponding to sentences with similar meaning and/or structure lie close to each other when projected to a two-dimensional space. In the meanwhile, context vectors of unrelated sentences do not showcase this behavior. While a similar phenomenon is observed when using simple bag-of-words representations, NMT context vectors preserve differences in word order; e. g. *"John admires Mary"* and its paraphrases do not tend to lie in the same cluster together with the sentence and paraphrases of *"Mary admires John"*.

ML-NMT systems showcase a similar behavior, according to [Johnson et al., 2016]. Sentences coming from different languages tend to belong to the same cluster when they are grouped together by such unsupervised methods. The only exceptions are zero-shot directions: if a language pair does not have parallel training data during the training phase of the system, the context vectors belonging to these examples occupy different regions of the embedding space than their semantically similar counterparts. A probable explanation for this is that the authors use context vectors that have already been weighted by the attention mechanism that introduces language-dependency as it serves as a soft alignment between source and target sentences.

It has been shown in [España-Bonet et al., 2017] that similarity measures between internal representations can be successfully used for discriminating sentence pairs that are either translations of each other, similar in meaning or semantically unrelated.

1.2 Domain Adaptation

As the amount of available in-domain data for certain MT tasks is often scarce, it is unrealistic to build a well-performing system using only such parallel texts. This problem calls for the utilization of larger-scale out-of-domain corpora.

1.2.1 Acquiring In-Domain Corpora

Additional in-domain data can be acquired in different ways. Similarly to any training corpus, domain-specific parallel data can be created manually. This approach results in high-quality data, but requires vast amounts of human labor and adequate funds.

If we are to automatically create more in-domain parallel data, there are two main approaches to follow. First, it is possible to translate monolingual corpora by an MT system (cf. [Schwenk, 2008, Lambert et al., 2011, Sennrich et al., 2016a]). Alternatively, one can use language models trained on in-domain data to select domain-specific sentence pairs from general parallel corpora [Axelrod et al., 2011]. Another possibility is to identify parallel sentences in *comparable corpora* by defining various measures of sentence similarity operating on syntactic and/or semantic features [Rauf & Schwenk, 2011, Skadiņa et al., 2012, Barrón-Cedeño et al., 2015]. We follow the latter method in this thesis project, focusing on using NMT encoder embeddings to such ends. This allows for getting parallel data for under- and zero-resourced language pairs (as the translation approach will be unlikely to produce good-quality results even in a ML-NMT setting). For the detailed description of the system cf. Chapter 2. This way, we can create adaptation sets without extensive efforts from translators; however, in this case we have to make sacrifices regarding the quality of the automatically created/extracted sentence pairs.

1.2.2 Domain Adaptation for Statistical Machine Translation

The possibilities of adapting SMT systems to certain domains has been thoroughly studied. The main approaches include selecting in-domain sentences from larger out-of-domain corpora (e. g. by using in-domain language models) [Yasuda et al., 2008, Moore & Lewis, 2010, Duh et al., 2013, Lu et al., 2007] and interpolating in- and out-of-domain translation models [Koehn & Schroeder, 2007, Bisazza et al., 2011, Finch & Sumita, 2008, Sennrich, 2011].

1.2.3 Domain Adaptation for Neural Machine Translation

Domain adaptation is a relatively new research line within the NMT framework. The different approaches revolve around the possible ways of combining in-domain data

with larger quantities of available out-of-domain text. In a simple scenario, parallel texts of both types can be mixed together for training the NMT system. Since the training data distribution is unbalanced with regard to the domains due to the difference in available amounts, in-domain data needs to be up-sampled [Chu et al., 2017].

One more refined attempt is fist training a NMT system on larger amounts of out-of-domain data. Then a *transfer learning* method can be applied, meaning additional epochs of training are performed on smaller-scale in-domain data. As [Freitag & Al-Onaizan, 2016] point out, this might lead to overfitting, and to overcome this issue, they propose using an ensemble of the out-of-domain and the domain-adapted model at translation time. In [Chu et al., 2017] an additional method is discussed. Inspired by ML-NMT settings, where target forcing tags are appended to source sentences, their proposed model mixes in- and out-of-domain data in a single system where each sentence has the additional information represented in a domain tag. This way, the system can be forced to learn to generate sentences for the specified domains.

Another method described in [Watanabe et al., 2016] builds on the idea of parameter augmentation. While this research focuses on caption generation, it can probably applied to NMT domain adaptation as well. Here, the output parameters of a neural system are dependent on the domain of the input data, and both can be decomposed into two terms, one of which is shared for all domains, while the other is domain-specific. During training time, only the respective terms are optimized for each minibatch, depending on their domain.

Chapter 2

Resources and In-Domain Corpora Generation

This chapter is organized as follows. First, in Section 2.1 we describe the available parallel corpora used for the purposes of the thesis project. These have been collected and created within the CLUBS project prior to this work, and made available for the experiments we conducted; the statistics and the description of the contents of the various corpora are based on the internal documentation of the project. Second, in Section 2.2 we introduce in-domain comparable corpora extracted from *Wikipedia*. Section 2.3 is dedicated to describing methods for automatically extracting parallel sentences from such comparable corpora.

2.1 Parallel Corpora

2.1.1 In-Domain Corpora

pubPsych Corpora

The CLUBS project aims at improving multilingual document retrieval from the *pubPsych* database[1] by using MT. The database contains articles in the psychology do-

[1] `https://www.pubPsych.eu/`

Table 2.1. *Number of records in different languages in the pubPsych database.*

	de	*en*	*es*	*fr*
Titles	324,005	895,982	53,065	47,707
Abstracts	250,263	513,000	34,815	33,206

Table 2.2. *Availability of resources in number of records between various language combinations in the pubPsych database.*

	en–de	*en–es*	*en–fr*	*de–es*	*de–fr*	*es–fr*	*en–es–fr*	*de–en–fr*
Titles	307,37	25,680	45,324	7	50	2	2	6
Abstracts	47,218	16,934	189	0	0	105	105	0

main in several languages. The fact that these articles are translations of each other, allows for building parallel corpora by applying sentence-level alignment.

The database consists of 958,726 articles, from which the titles and abstracts are used for training, adaptation and testing purposes (the actual content of the articles is not available due to copyright reasons). It has to be pointed out that titles and abstracts are not available for every instance and/or for every language, which leads to uneven amounts of parallel data among different language pairs when creating parallel corpora. The exact numbers can be observed in Table 2.1. The most data is available for *en*, while *de* has the second most amount of text. *fr* and *es* have roughly the same amount of documents, that is considerably less than the previous two languages. When looking at available parallel data in Table 2.2, one has to notice that while *en–de* and *en–es* are sufficiently resourced, language pairs not involving *en*, as well as the *en–fr* pair do not have sufficient parallel data. Furthermore, the number of documents that are present all in *en*, *es* and *fr* is scarce (it overlaps with the *en–fr* pair), while there are even less records involving *en*, *de* and *fr*. There are no titles or abstracts that exist in all four languages. This fact justifies the need for exploring ML-NMT approaches, with a special focus on under-resourced directions that is a core element of this thesis project.

The corpus is divided into training, development and test sets; the distribution of these three partitions by language pairs is displayed in Table 2.3.

Table 2.3. *Statistics of the pubPsych parallel corpora by language pair, partition and titles/abstracts.*

Abstracts

	en–de			*en–es*			*en–fr*		
	snt.	*en* tok.	*de* tok.	snt.	*en* tok.	*es* tok.	snt.	*en* tok.	*fr* tok.
Train	241,749	6,584,364	6,135,612	88,848	2,640,441	2,909,559	0	0	0
Dev.	1,500	39,968	37,557	1,500	45,611	50,831	0	0	0
Test	2,162	60,219	55,610	2,486	74,382	81,575	823	25,884	29,226

Titles

	en–de			*en–es*			*en–fr*		
	snt.	*en* tok.	*de* tok.	snt.	*en* tok.	*es* tok.	snt.	*en* tok.	*fr* tok.
Train	306,640	3,480,727	3,059,048	25,105	293,164	340,203	45,137	463,610	567,618
Dev.	0	0	0	0	0	0	0	0	0
Test	737	9,691	8,202	575	6,935	8,002	187	2,589	3,012

EMEA and *Scielo* Corpora

As the strictly in-domain *pubPsych* data is not sufficient for training NMT systems from start, we only consider it using for domain adaptation experiments. In the light of this, the general systems are first trained on parallel data from similar domains described here, in combination with out-of-domain data introduced in Subsection 2.1.2.

The EMEA parallel corpus [Tiedemann, 2009] contains sentence-aligned documents of the *European Medicines Agency*[2] and covers 22 languages. For training NMT systems, only language pairs involving *en* are used. This fact also allows for studying zero-shot translation in our experiments.

The *Scielo* corpus contains documents of the *Scientific Electronic Library Online*[3] covering the domains of health and psychology. It has been prepared by the organizers of the *Biomedical Translation Task in the First Conference on Machine Translation*[4] (WMT16) and covers *en–es* and *en–fr* language pairs.

[2] `http://www.emea.europa.eu`
[3] `http://www.scielo.org`
[4] `http://www.statmt.org/wmt16/biomedical-translation-task.html`

Table 2.4. *Size of the general, EMEA and Scielo parallel corpora.*

	en–de			*en–es*			*en–fr*		
	snt.	*en* tok.	*de* tok.	snt.	*en* tok.	*es* tok.	snt.	*en* tok.	*fr* tok.
UN	162,981	6,098,083	5,617,876	11,196,913	320,064,682	366,072,923	12,886,831	361,877,676	421,687,471
EP	1,920,209	53,091,548	50,548,739	1,965,734	54,505,707	57,047,216	2,007,723	55,730,752	61,888,789
ComCrawl	2,399,123	58,864,439	54,570,779	1,845,286	46,855,705	49,557,537	3,244,152	81,084,856	91,281,890
Subtotal	**4,482,313**	**118,054,070**	**110,737,394**	**15,007,933**	**421,426,094**	**472,677,676**	**18,138,706**	**498,693,284**	**574,858,150**
EMEA	1,108,752	14,477,119	13,197,725	1,098,333	14,334,648	15,975,506	1,092,568	14,317,365	17,046,979
ScieloBio	–	–	–	117,862	3,252,183	3,382,511	–	–	–
ScieloHealth	–	–	–	558,714	14,382,853	15,031,533	9,129	244,486	308,055
Subtotal	**1,108,752**	**14,477,119**	**13,197,725**	**1,774,909**	**31,969,684**	**34,389,550**	**1,101,697**	**14,561,851**	**17,355,034**
pubPsych	241,749	6,584,364	6,135,612	88,848	2,640,441	2,909,559	–	–	–
Total	**5,832,814**	**139,115,553**	**130,070,731**	**16,871,690**	**456,036,219**	**509,976,785**	**19,240,403**	**513,255,135**	**592,213,184**

Table 2.5. *Size of the development and test sets available in the project.*

	en–de			*en–es*			*en–fr*		
	snt.	*en* tok.	*de* tok.	snt.	*en* tok.	*es* tok.	snt.	*en* tok-	*fr* tok.
news-test2012	3,003	72,988	72,603	3,003	72,988	78,887	3,003	72,988	81,797
news-test2013	3,000	64,809	63,411	3,000	64,809	70,540	3,000	64,809	73,658
EMEA dev	2,000	38,658	37,945	2,000	36,676	39,959	2,000	34,554	41,026
EMEA test	2,000	36,864	35,773	2,000	34,359	38,615	2,000	33,316	39,674
pubPsych dev	1,500	39,968	37,557	1,500	45,611	50,831	–	–	–
pubPsych test	2,162	60,219	55,610	2,486	74,382	81,575	823	25,884	29,226

2.1.2 General Corpora

In order to capture general non-domain-specific phrases by the NMT systems, entirely out-of-domain corpora are also used for training purposes. The include the *Europarl Corpus* [Koehn, 2005], the *United Nations Corpus* [Chen & Eisele, 2012] (both contain political documents) and web crawls, namely the *Common Crawl Corpus* made available within the *Shared Task on Machine Translation* (WMT).

Table 2.4 summarizes the available data among the different corpora broken down by language pairs involving *en*. Contrary to the in-domain *pubPsych* data, *en–fr* has the most available parallel sentence pairs in the out-of-domain case. The language pairs *en–es* and *en–de* both have significant amounts of in- and out-of-domain data, with a high bias towards general parallel corpora.

2.1.3 Development and Test Sets

There are six sets available for development and testing purposes; two in the general, two in the medical domain, and two strictly in-domain. The general ones are *news-test2012* and *news-test2013*. These are made available by the WMT workshop organizers. The medical-domain corpora are subsets of the EMEA corpus. The strictly in-domain development and test sets are subsets of the *pubPsych* corpora. Table 2.5 displays the statistics for all the development and test sets available within the project.

2.2 Comparable Corpora

As one of the goals of the thesis project is to test the applicability of using parallel data extracted from comparable corpora for domain adaptation, we design a system that is capable of automatically identifying such sentence pairs for all languages. Section 2.3 describes the method in detail; in this section we briefly summarize the resources used for this task.

2.2.1 In-Domain Comparable Corpus from *Wikipedia*

We use *Wikipedia* as a source for automatically extracting parallel sentences lying in the appropriate domain. The main idea is that certain articles exist in several languages, and while the content is rarely an exact translation between given language pairs, their content is comparable with respect to the available information, structure, etc. Thus, it can be assumed that within these articles certain sentence pairs are indeed matching translations of each other.

Using the *WikiTailor*[5] tool, we extract parallel articles of the psychology and health domains from Wikipedia. The extraction of articles in different languages is facilitated by the fact that such articles are connected by inter-language links. Since *Wikipedia* is represented as a graph, the domains can be restricted by searching it to a given depth taking the corresponding category (e. g. psychology or health) as the starting node. The search terminates when such a depth is reached that titles are not included in the vocabulary in the domain anymore.

Since the graphs are different for every language, the extracted articles will differ as well. Extracting linked articles can be done by taking either the intersection or the union of the nodes coming from different languages. The former approach leads to results with high precision and low recall, while the latter approach has low precision and high recall. Since the goal is obtaining large amounts of data, the union method is preferred for the purposes of our experiments, extracting articles that exist at least for two of the four languages. The amount of data obtained this way is shown in Table 2.6.

[5] `http://cristinae.github.io/WikiTailor/`

Table 2.6. *Number of extracted comparable in-domain Wikipedia articles and sentences.*

Articles

	es	*de*	*fr*
en	87,306	92,378	89,560
es	–	80,383	80,384
de	–	–	81,367

Sentences

	es	*de*	*fr*
en	5,867,565/2,794,176	6,063,864/3,822,984	5,986,395/3,102,534
es	–	2,640,159/3,467,394	2,644,061/2,907,173
de	–	–	3,513,156/2,934,459

For the purpose of extracting parallel sentences from this comparable data, we train supervised classification and regression models. As *Wikipedia* itself does not contain parallel sentence-level alignments, we need additional data for training purposes, which we describe in the following two Subsections.

2.2.2 BUCC Corpus

One corpus for training the parallel sentence identification system is the one provided for the *10th Workshop on Building and Using Comparable Corpora*[6] (BUCC 2017). This includes comparable data for the language pairs *en–de* and *en–fr*, indicating matching sentences within texts. The data consists of *Wikipedia* and *News Commentary* data. These two language pairs contribute to a total of 18,666 translated sentence pairs.

Due to the characteristics of the training corpora's design, negative examples (i. e. sentence pairs that do not match) are not indexed. In order to overcome this issue, we select a random subset from all possible sentence pairs. To obtain a balanced distribution of positive and negative samples, we sample the same number of random non-matching pairs. Thus, the final *BUCC* dataset contains 37,332 instances (cf. Table 2.7 for the figures).

[6] `https://comparable.limsi.fr/bucc2017/`

Table 2.7. *Statistics of the BUCC and SemEval corpora.*

Corpus	Language pair	Snt. pairs
BUCC	*en–fr*	18,172
	en–de	19,160
	Subtotal	**37,332**
SemEval	*en–es*	1,595
	en–en	20,278
	es–es	1,555
	Subtotal	**23,428**
	Total	**60,760**

2.2.3 *SemEval* STS Corpus

The second corpus is the one provided for the *SemEval 2017 Semantic Textual Similarity Task*.[7] It consists of sentence pairs and corresponding human-annotated continuous similarity scores ranging between 0 and 5. This data set includes parallel sentences that can be either bilingual or monolingual (i. e. paraphrases to varying degrees). The exact figures are displayed in Table 2.7. While the BUCC set is balanced with regard to available data for the two language pairs, this corpus has only little parallel data, and only for *en–es*. The majority of the instances is made up from monolingual *en* paraphrase pairs, while a less significant partition consists of *es* monolingual data.

2.2.4 Preprocessing

We preprocess the available data according to the general MT pipeline in order to make it usable for building and adapting systems for our purposes. These steps are:

1. Text normalization
2. Text tokenization
3. Truecasing

Text normalization and tokenization are carried out as implemented in the *Moses* toolkit [Koehn et al., 2007], with the normalizer having been previously adapted to cover some irregularities of the *pubPsych* data. The truecaser has been trained

[7] `http://alt.qcri.org/semeval2017/task1/`

on *Wikipedia* and *Europarl* V7 monolingual data and made available for this thesis project.

2.3 In-Domain Parallel Sentence Extraction

This section describes a way of automatically extracting parallel adaptation sets from comparable corpora using *Wikipedia* articles. As the thesis is centered around NMT, we lay a strong focus on using the internal representations of such systems for measuring similarity between cross-lingual sentence pairs.

2.3.1 Candidate Sentence Pairs in the *Wikipedia* Corpus

The details about building the comparable corpus are described in Subsection 2.2.1. After the linked in-domain articles have been extracted, the assumption can be made that these article pairs contain sentence pairs that are matching translations of each other. Since only some articles are translations of each other and others are constructed from scratch, each sentence has to be compared with each sentence in order to find instances where an exact matching can be observed.

If we consider all possible source-target sentence matchings within the given articles, the resulting number of candidate pairs is $n \times m$ for each article (where n stands for the number of sentences in the source article and m for the target article sentence count). Table 2.8 displays the number of candidate sentence pairs for each language pair. As it can be seen, there are almost 30 million candidate sentence pairs considering only language pairs that include *en*. Because of the polynomial complexity, we filter out obvious negative candidates, defined by sentence length ratio. Namely, if one sentence is twice or more as long as the other (measured by token count), we do not consider that candidate pair. Furthermore, we ignore sentences that are likely to be scientific formulae by the simple heuristic of checking for certain special mathematical characters (equals sign, arrows, backslashes etc.). "Sentence pairs" such as these are not considered useful for translation engines, even if they are indeed parallel. While these steps slightly reduce computational cost, the later classification steps still require heavy parallelization to keep running times at a tolerable level.

Table 2.8. *Number of candidate sentence pairs for language pairs.*

	es	*de*	*fr*
en	361,103,606	470,887,694	413,752,230
es		226,577,446	205,713,405
de			274,479,174

2.3.2 Feature Extraction and Similarities

Our goal is to build a supervised predictive model that is capable of identifying matching sentence pairs. For this purpose, several metrics are to be calculated that are capable of measuring similarity between source and target side sentences. Some of them capture semantic similarity while others capture similarities in length and vocabulary.

Context Vectors (*ctx*)

As it was discussed in Subsection 1.1.5 of Chapter 1, the context vectors of a NMT system lie in a shared embedding space. This property enables using the encoder stage of such an architecture in order to obtain these vectors for any given sentence. For this purpose, we use the general-domain ML-NMT system available in the CLUBS project. The system has been trained on the general, EMEA, and *Scielo* corpora described 2.1.1 and 2.1.2. We use a model that has been trained with the *Nematus* system[8] using 512-dimensional word embeddings, no dropout, and *Adadelta* optimization with a learning rate of 0.0001. As it has been shown in [España-Bonet et al., 2017], the hidden layer size does not effect the internal representations with regard to their discriminatory power for semantic similarity/unrelatedness, the model has an 512-dimensional recurrent layer in the encoder, since this gives a fast performance. Furthermore, the same study shows that the difference for representational similarities between translations and unrelated sentence pairs stays constant after as early as 0.5 epoch of the training (although the context vector representations keep evolving). In the light of this, we use a model at this point of the complete training procedure (as opposed to a fully trained one) in order to save time. It has to be pointed out, however, that for the best translation quality we use models that have been fully trained (i. e. the performance converges on a validation set).

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

The candidate sentence pairs are fed to the system's encoder stage separately in order to extract the corresponding context vector pairs. This is performed by a modified version of the *Nematus* toolkit that allows for the extraction of the context vectors. For the purposes of this thesis project, we use the context vectors *before* applying the attention mechanism. As it serves as a soft alignment between source and target sentences, it would eliminate the language-independent nature of the embeddings to some extent, while our goal is to design a system that can operate between any language pair. We then take the sum of each hidden state pair in the encoder to get a representation for the sentence of length n:

$$\mathbf{C} = \sum_{i=1}^{n} c_i \tag{2.1}$$

We can then measure the similarity between two context vectors ($\mathbf{C}_{src}$ and $\mathbf{C}_{tgt}$) using the *cosine similarity* measure:

$$\cos \vartheta = \frac{\mathbf{C}_{src}\mathbf{C}_{tgt}}{||\mathbf{C}_{src}||||\mathbf{C}_{tgt}||} = \frac{\sum_{i=1}^{n} \mathbf{C}_{src,i} \cdot \mathbf{C}_{tgt,i}}{\sqrt{\sum_{i=1}^{n} \mathbf{C}_{src,i}^2}\sqrt{\sum_{i=1}^{n} \mathbf{C}_{tgt,i}^2}} \tag{2.2}$$

The value varies between [-1,1]; the more similar two vectors are, the more close this value gets to 1.

Context vectors are used not only in terms of similarity features, but also as inputs to (neural) classifiers. This approach is discussed in 2.3.3.

Complementary Features (*comp*)

In addition to context vectors that mostly depend on semantics, we extract certain complementary features that account for textual ("syntactic") similarities between sentence pairs.

Character n-Grams We take advantage of the fact that parallel sentences share vocabulary, and there are orthographical similarities among matching words (as long as the same alphabets are used). As described in [McNamee & Mayfield, 2004], if candidate sentence pairs are represented as sequences of characters, the sentences can be split into character n-grams. Namely, two vectors can be constructed, where each dimension stands for different character n-grams occurring in any of the two sentences. The values of one vector are the actual counts of the given character n-grams in the corresponding sentence. Since the languages include *de* that is known for containing long compound words, whitespace characters are ignored during computation. The cosine similarity can then be calculated between these two vectors in a similar fashion to context vector similarities. For the purposes of this project, the value of n is varied between [2,5], calculating similarity features for all four n-gram vectors. An example demonstrating 2-gram character vectors for the words *aquatic* (*en*) and *acuático* (*es*) can be seen below:

$$aquatic \rightarrow (aq = 1, qu = 1, ua = 1, at = 1, ti = 1, ic = 1, ac = 0, cu = 0, co = 0)$$
$$acuático \rightarrow (aq = 0, qu = 0, ua = 1, at = 1, ti = 1, ic = 1, ac = 1, cu = 1, co = 1)$$

Pseudo-Cognates In natural language, cognates are *"words that are similar across languages"* [Manning & Schütze, 1999] that can also serve as indicators of connection between multilingual sentences. Here we use a less restrictive definition referred to as *pseudo-cognates* [Simard et al., 1993] that include the following three cases:

1. Punctuation marks
2. Tokens that only contain digits
3. The first four characters of tokens that are at least four characters long

Two vectors can be constructed in a similar fashion to character n-gram vectors. In this case, each dimension stands for a given pseudo-cognate candidate occurring in any of the two sentences the value indicating the number of occurrences in the given sentence. As before, the cosine similarity measure between these two vectors is used as a feature. For example, the *en* sequence *embarrassed casualty* and the *es* sequence

Table 2.9. *Average length factor values (μ) and their standard deviation (σ) for each language pair.*

Language pair	μ	σ
de–en	0.95	0.63
en–de	1.17	0.77
de–es	1.00	0.31
es–de	1.11	1.32
fr–de	1.02	0.54
de–fr	1.03	0.30
es–fr	1.04	0.38
fr–es	1.02	0.33
es–en	0.93	0.44
en–es	1.13	0.42
fr–en	0.91	0.31
en–fr	1.16	0.41

embarazada casualidad would both have pseudo-cognate vectors of (1, 1) and a cosine similarity of 1.

Length Factor Intuitively, similar sentences have similar lengths. The differences between languages (average word length and count), however, call for a metric that account for these variations. We use the *length factor* parameter as defined in [Pouliquen et al., 2006]:

$$\rho(s,t) = \exp\left[-0.5\left(\frac{\frac{|t|}{|s|} - \mu}{\sigma}\right)^2\right] \tag{2.3}$$

where $|t|$ and $|s|$ stand for target and source sentence length in characters, μ and σ are the average length factor values an their standard deviation for the given language pair respectively; cf. Table 2.9 for details.

Token and Character Count Character and token counts on the source and target sides are also included as features in addition to the length factor that captures the same idea in a different manner.

2.3.3 Parallel Sentence Identification

In this section we describe various methods for identifying parallel sentence pairs using different combinations of the features introduced in Subsection 2.3.2. We discuss simple greedy search approaches for determining optimal threshold values as a classification boundary. The focus, however, is laid on machine learning methods, treating the problem as a supervised classification/regression task.

Classification Task

We use the BUCC training corpus as described in Subsection 2.2.2 for training classifiers. In the following experiments we use the following partitioning for all results reported:

1. 87.5 % for training with cross-validation (32,666 instances)
2. 10 % for training an ensemble of the best performing models (3,733 instances)
3. 2.5 % for held-out testing purposes (933 instances)

Regarding the features, the following scenarios are examined:

1. *ctx*: as the main focus of the thesis is NMT, context vectors obtained from the NMT model
2. *comp*: considering the fact the NMT context vectors mostly account for semantic similarities, the complementary features described in Subsection 2.3.2, as they mostly capture sentence similarity in terms of syntax (a line of investigation being whether this performs better and worse than the semantic information provided by the embedding space)
3. *all*: the combination of context vectors and the complementary features, examining whether the semantic and syntactic information is complementary

Classification with Context Vectors

Table 2.10. *Threshold values, corresponding training accuracies (Tr. Acc) and test results for context vector similarities on the BUCC corpora.*

	de–en	*fr–en*	Joint
Threshold	0.43	0.41	0.43
Tr. *Acc* (%)	97.2	97.4	97.3
Test P (%)	95.5	95.4	98.3
Test R (%)	97.1	100.0	98.1
Test F_1 (%)	96.3	97.7	98.2

Context Vector Similarities For the *ctx* method, a simple greedy search method can be applied as only one feature is available. This is done incrementally searching for suitable similarity thresholds between the lowest score among positive training examples and the highest one among negative instances. The resolution of the search (i. e. the step size) is set to 0.005 and the algorithm is optimized for accuracy on the training set. Using this approach, the optimal threshold for *de–en* lies at 0.43 that corresponds to a training set accuracy of 97.2 %. In the case of *fr–en* this value is 0.41 corresponding to 97.4 % accuracy on the training set. There are some differences between these values depending on the language pair in question, but since we are working with multilingual representations that are language-independent in nature, these are negligible. Based on this fact, a joint threshold is determined following the same approach on the joint data set involving both language pairs. This leads to a threshold value of 0.43 with a training accuracy of 97.2 % that aligns with the values obtained for *de–en*. For our goals, this joint approach is the preferred one, as the system is intended for extracting sentence pairs from multilingual comparable corpora. Table 2.10 shows the accuracies and similarity thresholds for both the separate language pairs as well as the for joint training set. The evaluation of the classification results on the held-out test set is also shown in terms of precision (P), recall (R) and F_1 score. The greedy method achieves $F_1 = 98.2$ on the joint dataset; it has to be noted that on *fr–en* subset it performs with a 100 % recall.

Raw Context Vectors While most of the models discussed in this section use context vector similarities as features, it is also possible to build classifiers on raw context vector outputs of the NMT system. In this case, the context vectors can be fed to one or more fully connected feed-forward layers of a deep neural network (DNN). In the simple classification scenario, a two-dimensional softmax layer can produce the class probabilities. Combining the feature vectors corresponding to each

Table 2.11. *Classification results (F_1 %) with DNN-based classification.*

Hidden dim.	Hidden layers	Concat.	Multi.	Sub.	All
512	1	88.9	97.4	82.4	97.1
	2	88.6	97.3	82.9	97.1
1024	1	88.3	**97.5**	83.8	96.9
	2	89.0	97.4	85.5	97.2
2048	1	89.4	97.4	85.6	97.0
	2	89.6	97.4	88.8	**97.3**
4096	1	**89.8**	97.0	88.1	97.2
	2	**89.8**	97.4	**91.0**	**97.3**

of the candidate sentences at the input layer can be done in several ways. The most straightforward solution is a simple concatenation of the vectors ($[\mathbf{C}_{src}, \mathbf{C}_{tgt}]$). In addition to this, they can be combined with element-wise multiplication of each of the components ($\mathbf{C}_{src} \odot \mathbf{C}_{tgt}$), or by subtracting one candidate vector from the other ($\mathbf{C}_{src} - \mathbf{C}_{tgt}$). We test all of these scenarios. Finally, we concatenate the results of all three methods resulting in a feature with four times as many dimensions as that of one context vector: $[\mathbf{C}_{src}, \mathbf{C}_{tgt}, \mathbf{C}_{src} \odot \mathbf{C}_{tgt}, \mathbf{C}_{src} - \mathbf{C}_{tgt}]$. The training is implemented in *TensorFlow*[9], using the *AdaGrad* optimizer with a learning rate of 0.0001.

Table 2.11 summarizes the results achieved with the DNN-based classification using the four approaches described above: concatenation (Concat.), element-wise multiplication (Multi.), subtraction (Sub.), and the concatenation of the three methods (All). We display F_1 scores on the held-out test set with eight different architectures; the hidden layer dimension (Hidden dim.) varies between 512 and 4096, and the number of hidden layers between 1 and 2. For each method, the best results are typeset in bold.

The main differences in the results stem from the different input representations, while the DNN architectures influence the performance to a smaller extent. The concatenation method's performance slightly increases as the architectures become more complex. The best results are delivered when using 4096 neurons in the hidden layers, however, they are well below the performance of the greedy search method on context vector similarities. The trends are similar for the subtraction method. The best result is somewhat higher in this case, and there is a noticeable improvement in F_1 scores depending on the number of hidden layers, especially in the networks using

[9] `https://www.tensorflow.org/`

Table 2.12. *Thresholds values of similarity averages on the BUCC training copora using the all feature set.*

Thrs.	P	R	F_1
0.84	96.83	94.89	95.85

2048 and 4096 hidden units. The multiplication and combined approaches manage to reach results higher than $F_1 = 97.0$. With respect to the neural network architectures, while the combined method's performance saturates at more complex architectures, the element-wise multiplication of the context vectors performs best when using 1 hidden layer with 1024 units.

The best result we achieve with this set of experiments is $F_1 = 97.5$ on the held-out test set. While the value is high, it does not overperform the greedy search method on context vector similarities. It has to be pointed out that building classifiers on raw context vectors is a more complex problem due to the inputs' higher dimensionality, which can explain the worse performance of these approaches. Therefore, in the rest of the experiments we use similarity measures for our purposes due to their simpler nature.

Complementary and Combined Features

Threshold Search In the scenarios *comp* and *all* we employ 7 and 8 features respectively. One approach to build a simple classification model is to systematically find threshold values for each feature that maximizes the F_1 score of the classification [Barrón-Cedeño et al., 2015] and finding appropriate ways of combining these thresholds for making the classification decision. Here, we first experiment with a simple approach where we combine all six similarity measures (context vector, n-gram and cognate vector) by taking their average. On the joint BUCC dataset we achieve an F_1 score of 95.85 by this approach, the threshold lying at 0.84 (the details are summarized in Table 2.12).

Supervised Classification While we have seen that a simple greedy threshold search methods can already produce satisfactory results for parallel sentence identification, we explore the possibilities of building supervised classifiers on the available

Table 2.13. *Precision (P), Recall (R) and F_1 scores (%) obtained on the binary classification of sentence pairs on the held-out test set.*

		de–en			*fr–en*			Joint		
		P	R	F_1	P	R	F_1	P	R	F_1
ctx	Thrs.	95.5	**97.1**	96.3	95.4	**100.0**	97.7	98.3	98.1	98.2
	SVM	96.2	96.2	96.2	95.6	99.1	97.3	97.1	98.0	97.6
	GB	97.0	95.7	96.4	95.6	99.6	97.6	97.0	97.3	97.2
	Ens.	98.2	95.7	97.0	95.6	99.1	97.3	96.9	97.8	97.3
comp	SVM	72.3	85.5	78.4	76.7	85.1	80.7	73.4	80.9	77.0
	GB	93.5	85.1	89.1	97.2	93.2	95.1	96.9	90.7	93.7
	Ens.	84.0	89.4	86.6	95.5	95.5	95.5	93.4	91.6	92.5
all	SVM	74.6	86.4	80.1	81.8	87.3	84.5	86.1	85.6	85.8
	GB	98.7	96.6	**97.6**	**99.1**	99.6	**99.3**	**98.9**	98.9	**98.9**
	Ens.	**99.1**	96.6	**97.8**	**99.1**	99.6	**99.3**	98.7	**99.1**	**98.9**

features. To this end, we choose to train support vector machines (SVM) with radial basis function kernel, and gradient boosting (GB) using the deviance objective function. The models are trained on the training set with 10-fold cross-validation. Finally an ensemble of these two models using soft voting is trained on the separate training subset selected for this purpose. For the course of these experiments we use implementations of the *scikit-learn*[10] toolkit in *Python*. We train and test our models in all three scenarios, i. e. *ctx* (using only context vector similarities), *comp*, and the combination of all 8 features (*all*).

Precision (P), Recall (R) and F_1 scores are displayed in Table 2.13 for all three scenarios. It has to be pointed out that when using context vector similarities only (*ctx*), the greedy search yields in better results than any machine learning approach within this scenario, although the differences seem to be negligible. A bigger difference in performance is observed when comparing this scenario to any approach using the *comp* features only: the greedy search for *ctx* significantly overperforms any of the machine learning methods trained on the feature set consisting of the 7 syntactic features. From this, it can be concluded that while having no knowledge of semantics whatsoever and relying only textual similarities can result in a high performance (F_1=93.7 on the joint dataset), sentence embeddings by themselves are clearly better choices for this task as they bear valuable interlingual semantic information about the candidate sentences.

As we have previously seen in 2.3.3, the performance in the *ctx* scenario is indepen-

[10] `http://scikit-learn.org/stable/index.html`

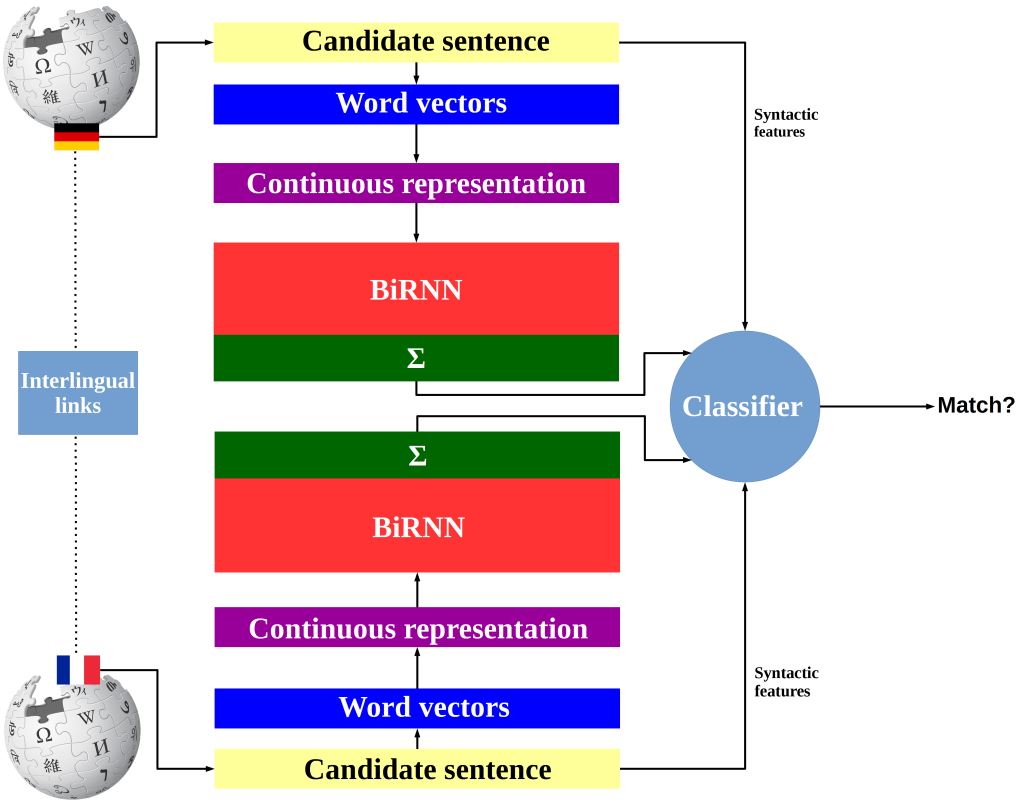


Figure 2.1. *System architecture*

dent of the language pair in question due to the language-independent nature of the approach. Contrary to this observation, the complementary feature set *comp* does showcase differences depending on the language pair involved, and has a noteworthy performance drop in the case of *de–en*. This also results in a worsening performance in the joint dataset when comparing results to the *fr–en* partition. When we combine both context vector and complementary features in the *all* scenario, the same differences are observable due to the incorporation of the *comp* feature set. However, we achieve $F_1 = 98.9$ % on the joint set with the full feature set, even if this result is worse than on the *fr–en* partition ($F_1 = 99.3\%$). This result is the best one on the full dataset, which leads to the conclusion that the inclusion of complementary features is indeed beneficial for our purposes and syntactic-textual characteristics do complement the mainly semantic information carried in the context vector similarities well.

Table 2.14. *Size in number of parallel sentences of the extracted in-domain corpus from Wikipedia.*

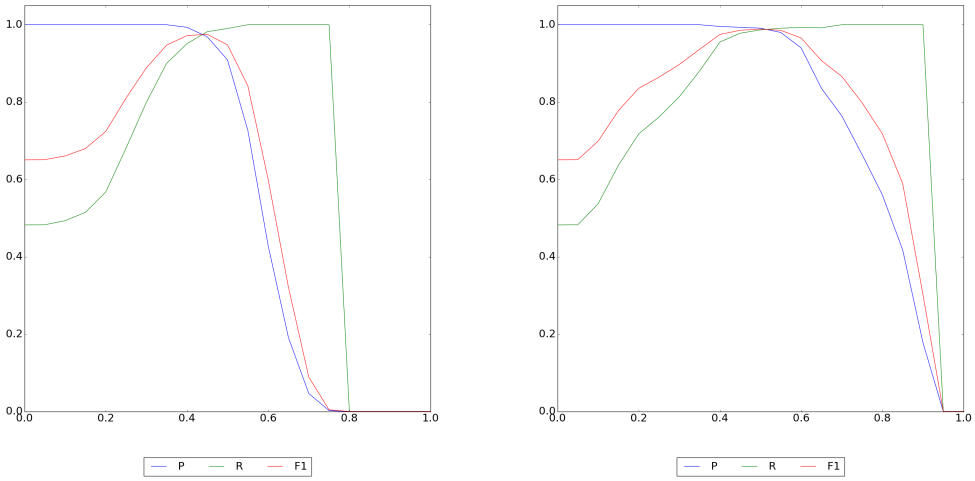
	de–es	*en–es*	Joint
ctx	24,491,723	42,116,484	66,608,207
set	13,415,885	9,070,685	22,486,570
all	25,551,816	46,194,714	71,746,530
union	36,479,058	57,758,832	94,237,890

Extraction from *Wikipedia* Articles

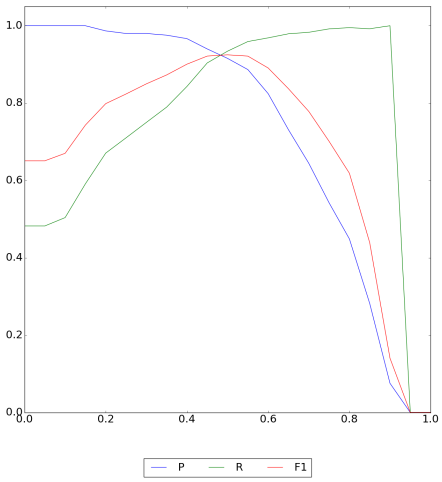
After running the experiments described above, we use the best-performing models to extract matching sentence pairs from the *Wikipedia* articles extracted previously (cf. Subsection 2.2.1). To this end, we use the classification output label of the model to filter out the exact translations, running the classifier on candidate sentence pairs within articles, discarding those that do not meet the criteria detailed at the beginning of this section (length constraints and mathematical equations). The extraction pipeline is illustrated on Figure 2.1.

We run preliminary experiments on the *en–es* and *de–es* subsets; we choose these two pairs in order to study one high-resourced and one zero-shot scenario. Table 2.14 displays the number of extracted sentence pairs from articles covering these language pairs. The extracted sentence pairs, however, contain too much noise for our purposes with any of the classifiers. Examining the obtained data reveals that the classifier that has a high F_1-score on the held-out subsection of the training set is highly biased towards recall on the *Wikipedia* data. In the following paragraphs we discuss several alternatives for extracting higher-quality parallel sentence pairs using a classifier obtained on the BUCC training data. Unfortunately, as we do not have an available gold standard, we need to manually examine the extractions and heuristically decide whether it is a viable approach.

Using Confidence Scores In this approach, the default 0.5 classification threshold is shifted towards higher values in order to overcome the high number of false positives. This leads to a lower recall, but at the same time it increases the precision. The held-out test set is used for determining a suitable tradeoff between these two values while keeping the F_1-score reasonably high. The precision-recall curves and the corresponding F_1 curves are displayed on Figure 2.2.



(a) *ctx* **(b)** *comp*



(c) *all*

Figure 2.2. *Precision (P), recall (R), and F_1 scores vs. decision threshold.*

In the light of these results, a decision threshold of 0.8 is a suitable value, as in the case of the *comp* and *all* the F_1 score starts to radically decrease after this point. Similarly, in the case of the pure context vector classifier, a threshold of 0.6 seems appropriate (using the simple threshold search classifier).

In an alternative related approach, all extracted pairs are ranked according to their confidence scores. This allows for only the top N candidates being considered for the domain adaptation experiments.

Quality of Extractions The original BUCC training data is problematic regarding two aspects. Firstly, the classification task on this dataset is easier than on *Wikipedia*; in the latter case sentences being compared come from the same articles, thus they share vocabulary and semantics even if they are not matching translations. The BUCC training set only contains binary examples, i. e. sentences that are either exact translations of each other and pairs that are completely unrelated. This binary nature of the training data does not account for paraphrases and partial translations that are often to be found among the comparable *Wikipedia* articles. Ideally, we would use training corpora that better mirror the properties of the *Wikipedia* extractions, however, currently there is no such data available.

The second issue arises in the approach of using confidence scores. Most of these values lie in a very narrow interval for the extracted sentence pairs (the length of the interval is below 0.1 for *en–es*), which makes the confidence ranking approach problematic. To overcome this problem, we propose using regression models instead of classifiers, which we discuss in the next paragraphs.

Training Regression Models

In our first regression approach the classifiers are replaced with regression models while still using the BUCC data, and instead of using the confidence of the classification, the predicted regression scores are used for ranking.

In the second method, we train our regression models on the combination of the BUCC corpus and the *SemEval STS* corpora introduced in Subsection 2.2.3. This dataset, contrary to the BUCC data, contains sentence pairs with varying degrees of similarity.

As the available bilingual data is scarce (cf. Table 2.7), further steps are necessary for obtaining an adequate training set. As the context vectors of NMT systems showcase a language-independent nature, the *en–en* and *es–es* subsets can be used alongside with the *en–es* embedding similarities for training models that only operate on these features. On the other hand, the complementary syntactic features can only be calculated on the *en–es* data. This issue can be solved by using the monolingual *en–en* subset to translate the target side sentences into either *es* or *fr* with the same pre-trained general-domain ML-NMT system used for obtaining the context vectors.

This way, two different sets are obtained that can be used for training different systems:

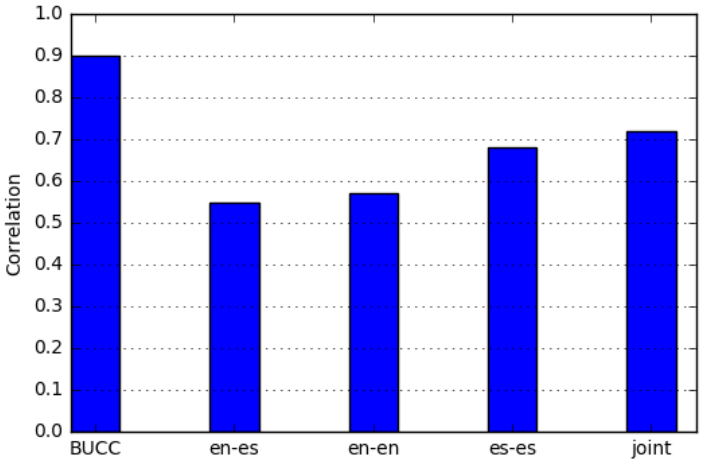


Figure 2.3. *Pearson correlation between the context vector similarities and labels on the different training subsets.*

one can be used for building regression models only on context vector features, while the other can operate in all three scenarios (*ctx*, *comp*, and *all*).

In order to examine the quality of the training corpora acquired this way and to compare it to that of the BUCC training set we examine the correlation of the different features with the label score by using the Pearson-ρ metric. First, the test is performed on the data set using context vectors only. The results are shown on Figure 2.3, broken down into different subsets of the data along with the joint correlation. It can be seen from these figures that while context vector similarities of the BUCC data set have a correlation with the training as high as 0.9, the newly added subsets have significantly lower ρ-values. This results in a lower correlation for the joint dataset (slightly above 0.7).

Figure 2.4 reveals similar trends on the other dataset, where the complementary similarity measures can be calculated in addition to context vector similarities. The correlation of syntactic features tends to change proportionally with that of the context vector similarity measures. The only exception is the length factor parameter that stays almost the same as in the case of the BUCC dataset in the case of the *en–es* complementary data. The portions created by translating monolingual data, however do not showcase this behavior.

Table 2.15 displays the performance of these new models. We run the tests on two

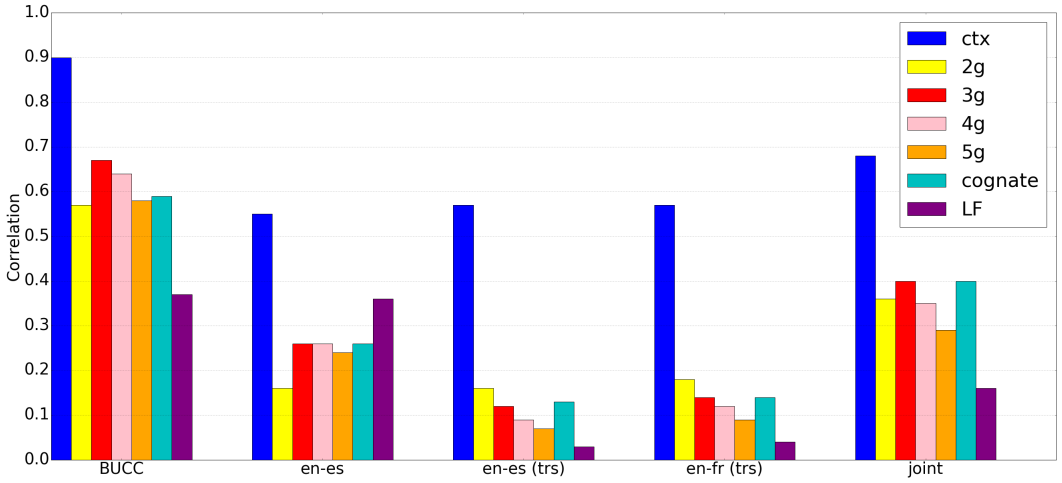


Figure 2.4. *Pearson correlation between different features and labels on the different training subsets.*

Table 2.15. *Held-out test set results with the additional dataset. Results are shown for BUCC-only (BUCC) and joint (BUCC+SemEval) training copora (Tr.) and test sets (Test)*

		BUCC test				BUCC+SemEval Test			
		P	R	F_1	MSE	P	R	F_1	MSE
BUCC+SemEval Tr.	ctx	97.8	96.9	97.3	0.77	80.9	97.7	88.5	1.66
	all	98.5	98.7	98.6	0.37	85.4	93.0	89.1	1.02
BUCC Tr.	ctx	97.0	97.3	97.2	0.78	80.7	98.1	88.5	2.25
	all	99.3	98.9	98.9	1.63	76.9	92.4	83.9	3.30

different test sets: the original BUCC held-out test data and a randomly sampled 2.5 % portion of the full training set excluded from training (including both BUCC[11] and *SemEval* data). As in our previous experiments the gradient boosting approach's performance was very close to that of the ensemble methods, this time we omit training regression ensembles and use only gradient boosting regressors. The BUCC regression models' performance on both test sets is also shown in comparison. For the precision, recall and F_1 metrics we use a decision threshold of 2.5 on the regression outputs. On the test set covering the full data, we achieve the lowest mean squared error (MSE) and best F_1 score (after applying the decision threshold) with the *all* feature set; on this test set this performs significantly better than models trained on only the BUCC

[11]In order to comply with the rest of the dataset, we convert the BUCC binary labels to scores of 0 and 5.

data.

Reranking with Feature Averages

We use the best performing model from the above scenarios to repeat the extraction process. Although the manual examination of the data sets leads to the observation that it contains more and higher quality true positives than in the case of previous systems for pairs with high regression scores, the extracted set still suffers from a high number of false positive sentence pairs.

One approach to this can be utilizing the pre-trained NMT system to filter extracted sentences. The architecture can be used as a multilingual language model that allows for scoring target sentences corresponding to given source sentences. This indeed leads to a ranking that separates positive and negative examples. However, we decide not to use this method as it can "erase" the effect of other language-independent features, and does not allow for the extraction between zero-resourced language pairs anymore.

Instead, we overcome this issue by applying a reranking method on the extractions; in this case our reranking score is simply defined by the average of the similarity features as described in 2.3.3. This approach eventually results in sentence pairs we consider good quality for automatically created parallel corpora. We can then use the top N pairs for each language pair and observe how it affects translation performance when used for domain adaptation. To make our extractions as clean as possible, we discard sentence pairs that are

1. not from the corresponding language (we use the *langdetect*[12] tool for this purpose)
2. have a relative token edit distance lower than 35 % of the average token length of the two sentences after function word removal

We perform the second step in order to filter out sentence pairs that are too similar, mainly instances that only contain proper names and connectives such as *James Bond and Michael Jackson – James Bond y Michael Jackson*. We hypothesize that such examples would not prove useful for our purposes due to their high redundancy. The

[12] `https://pypi.python.org/pypi/langdetect?`

token edit distance is calculated by the Levenshtein distance [Levenshtein, 1966], and the relative value is based on the average length of the two sentences. The above filtering steps are performed during the extraction of the top N candidates.

Chapter 3

Adaptation with In-Domain Corpora

This chapter is dedicated to examining the possibilities of domain adaptation via transfer learning and reranking, with a strong focus on the effect of the data quality, size, and domain. To this end, we explore three scenarios and make observations on how adaptation data quality (extracted automatically from parallel and comparable corpora) can influence NMT systems. Our investigation has several purposes: first, to see the achievable performance boost in terms of translation quality. Second, to check if using automatically extracted in-domain data is a viable option for adapting NMT models. Finally, we examine how the inherently noisier automatically extracted data affects NMT systems during domain adaptation in the presence of high-quality parallel corpora. We report our results using Bilingual Evaluation Understudy (BLEU) scores [Papineni et al., 2002], a metric based on n-gram matching precision between translation hypotheses and gold standards.

3.1 Domain Adaptation via Transfer Learning

As discussed in 1.2.3 of Chapter 1, there are various possibilities available for this purpose, including but not limited to transfer learning using a general out-of-domain system with in-domain data, ensembling adapted and unadapted systems, as well as performing target-forcing within the NMT system with respect to the training sen-

tences' domains. The goal of this section is to execute the transfer learning approach in various scenarios regarding the available adaptation data, namely:

1. using only parallel, clean data (*pubPsych*, PP)
2. using only the extracted data from comparable corpora (*Wikipedia*, WP)
3. using the combination of the two data sets (Merge)

For our adaptation purposes we use a fully trained NMT system as the general-domain baseline model, trained on EMEA, *Scielo*, and general data (cf. the first two blocks of Table 2.4 in Chapter 2). This model was made available in the CLUBS project framework and was trained with the following parameters:

Word embedding dimension:	512
Hidden layer dimension:	2,048
Vocabulary size:	80,000 + 2,000 BPE units
Optimizer:	AdaDelta
Learning rate:	0.0001
Batch size:	80

The adaptation procedure consists of continuing the training of the system for a given number of additional epochs. While during the training of the general system no dropout is applied, for adaptation purposes we utilize a dropout rate of 0.2 to avoid overfitting. It has to be pointed out that since the vocabulary is fixed (built from out-of-domain and *Wikipedia* data), the adapted systems' translations are limited to these words and subword units. This fact can affect the performance on in-domain data after adaptation, as what is not included in the vocabulary cannot be learned.

3.1.1 Data Preprocessing

In addition to the general steps that are necessary to carry out for any MT systems (cf. Chapter 2, Subsection 2.2.4), ML-NMT architectures require additional preprocessing steps, namely:

1. Applying BPE

2. Appending target forcing tags

The BPE conversion is carried out by the system implemented in the *Subword Neural Machine Translation* system[1] using a BPE model that has been trained on all language pair combinations of the CLUBS project. For ML-NMT scenarios, we need to indicate the desired target language on the source side, which is carried out by appending the tags `<2L2>` to the start of each source sentence, where `L2` stands for the target language (`en`, `es`, `de`, or `fr`).

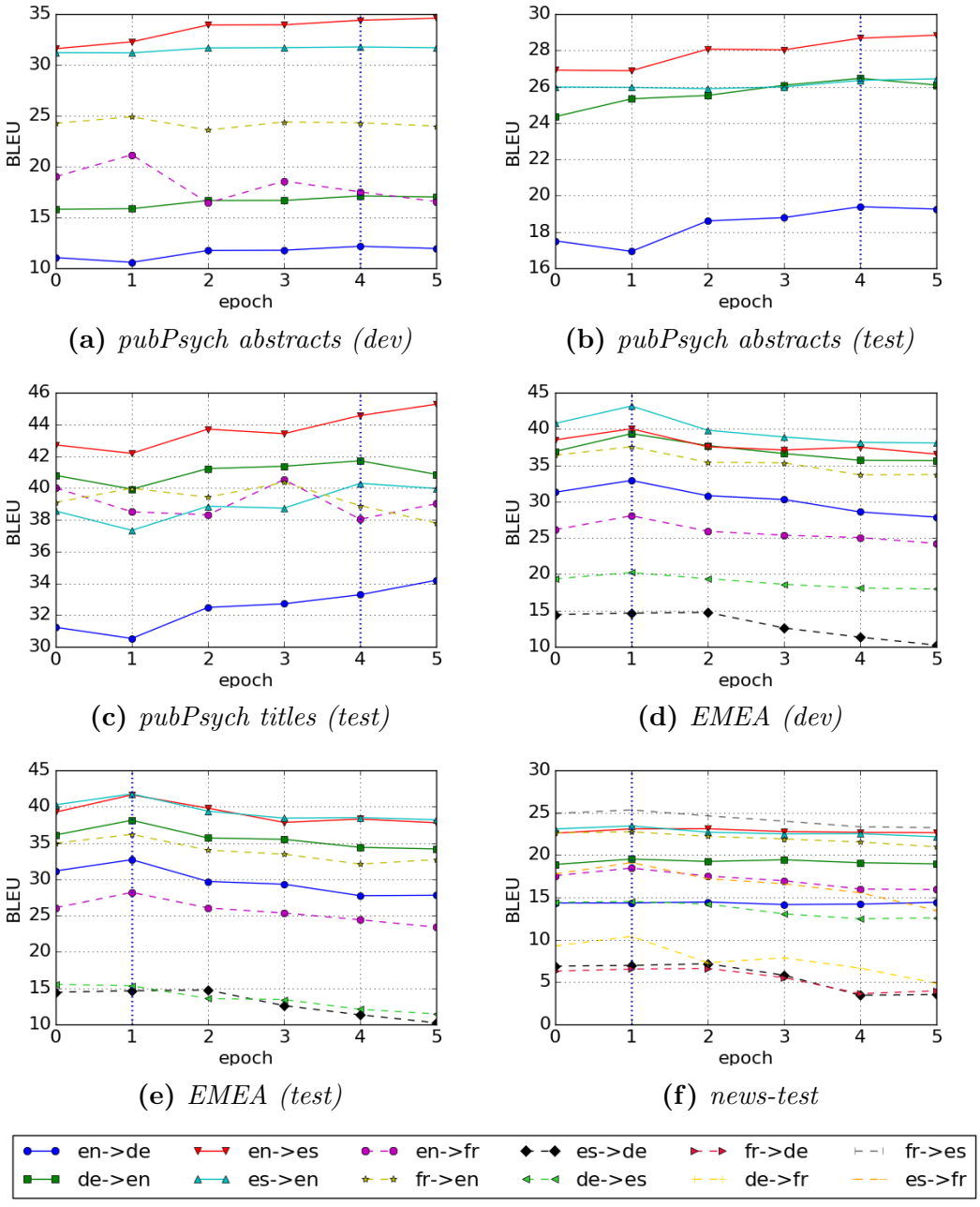
3.1.2 Transfer Learning with In-Domain Parallel Corpora

First, we run experiments using the high-quality parallel data only (PP), in order to see its effect on different language pairs and test sets in ML-NMT adaptation setting. Besides the language target forcing tags, we additionally introduce a category tag that specifies the origin of the sentence ("title" or "abstract") as this would further improve translation quality on the test data where the same information is available (*pubPsych titles* and *abstracts* test sets).

We run the training process for an additional 5 epochs and we examine the evolution of the translation performance in terms of BLEU score after each of these epochs. The adapted systems are first tested on the in-domain *pubPsych* test sets. We also run the evaluation on the close-domain test sets (EMEA) and out-of-domain ones (*newstest2013*). Apart from the advantage of being able to see the performance evolution for these missing language pairs, this is also beneficial for observing the effect domain adaptation has on out-of-domain datasets.

Figure 3.1 displays the evolution of the NMT system throughout 5 adaptation epochs. The figure is broken down by dataset, and different lines correspond to source-language pair combinations. Regarding the change in translation quality, several observations can be made. First, there are two different trends to be observed that depend on the type of the test data. It can be said in general that on in-domain datasets, the system's performance keeps improving until the fourth or fifth epoch. In most cases, the performance starts decreasing at the fourth epoch (a phenomenon that can be explained by overfitting). In the case of the three close- and out-of-domain datasets,

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



(a) *pubPsych abstracts (dev)*

(b) *pubPsych abstracts (test)*

(c) *pubPsych titles (test)*

(d) *EMEA (dev)*

(e) *EMEA (test)*

(f) *news-test*

Figure 3.1. *Evolution of domain adaptation through five epochs using the pubPsych adaptation corpus. Dashed lines indicate missing language pairs from adaptation data.*

however, the results saturate after one epoch of adaptation, and worsen after this point. The explanation for this behavior is that the overfitting takes effect much earlier due to the mismatch between the domain of the adaptation and test sets. Furthermore, this trend is repeated in the case of in-domain test sets for the language pair that is underrepresented in the adaptation set (*en–fr*, only titles are available). This is caused by the fact that even though the new data is beneficial, in this case its quality is inferior to the original training data, as that set contains sufficient parallel data for this language pair. Because of this, while one epoch of adaptation has positive effects, further training leads to decreasing quality.

From this experiment we can draw several preliminary conclusions to determine the direction of further steps and more detailed analysis. Firstly, it is clear that the transfer learning approach to domain adaptation is a viable one in the case of this particular system and available adaptation dataset. Secondly, it helps us determine the number of adaptation epochs where the performance of the adapted systems is expected to be the best. On in-domain data, for the majority of language pairs this lies at four additional epochs. In the light of this, we choose to test our systems at this point on the *pubPsych* datasets. Furthermore, we can draw the conclusion that domain adaptation can even be beneficial for translating data that is not strictly in-domain or out-of-domain. In this case, however, we use systems that have only been adapted for one additional epoch of training. Similarly, this seems to be the best point in training for translating in-domain texts between language pairs that do not have any or sufficient parallel data in the adaptation set.

Regarding the achievable performance improvement, the results are displayed in Table 3.3 to 3.7, in the blocks marked as NMT PP. Arrows represent significant changes with respect to the baseline performance (NMT BL) as calculated by the bootstrap resampling method [Koehn, 2004] implemented in Moses[2]. Statistical significance is indicated at $p = 0.005$. The evaluations are performed after 1 and 4 epochs of adaptation respectively, depending on the test set as described above. On the in-domain data we obtain significant improvement in all cases, that generally lies between 1 and 2 BLEU scores. In certain cases the improvements are not significant, most notably on the *en–fr* language pair where there is no adaptation data available (other cases are *es* → *en* on the *pubPsych abstracts* sets, and *de* → *en* on *pubPsych titles*). On

[2] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl`

Table 3.1. *Relative improvements on in-domain and general test sets using the manual adaptation data.*

Test set	Avg. (%)	SD (%)
pubPsych abs. (dev)	+7	4
pubPsych abs. (test)	+4	3
pubPsych tit.	+7	4
In-Domain full	+6	4
EMEA (dev)	+5	2
EMEA (test)	+4	3
news-test	+4	3
General & Close-Domain full	+4	3

the *pubPsych titles* test set, the performance is significantly worse in the case of the $en \rightarrow fr$ direction. The average relative improvements (Avg.) and their standard deviations (SD) are displayed in Table 3.1, broken down by test set.

The effect of domain adaptation with the parallel corpora on close- and general-domain test sets showcases similar trends (cf. Table 3.1 for the general overview, and Tables 3.3 to 3.7 for the full details). The improvements, however are significant in less cases. Furthermore, for the $es \rightarrow de$ direction on the EMEA *test* set results worsen significantly. Nevertheless, the domain adaptation process demonstrates some positive effect on non-present language pairs for both types of test data, meaning that "zero-shot domain adaptation" is a possibility for NMT systems.

3.1.3 Transfer Learning with In-Domain Comparable Corpora

In the next step we examine the extent the automatically extracted parallel sentences from *Wikipedia* can be used for domain adaptation purposes. In order to get a clear picture of how the amount and quality of the data affects NMT systems, we experiment with different partitions of the data. Namely, we take the top N extracted sentence pairs for each language pair according to the score produced by the ranking approach as described in 2.3.3 of Chapter 2. We set N to 5,000 (WP5), 10,000 (WP10) and 30,000 (WP30) resulting in datasets with 60,000, 120,000 and 360,000 sentence pairs respectively. These partitions are significantly smaller than the PP adaptation data (1,414,958 sentence pairs in total), and their covered domain is less homogeneous as they are extracted automatically from articles about health and psychology. We

Table 3.2. *Distribution of target forcing tags in the PP adaptation dataset.*

Tag	Count	%
<2en>	707,479	50.0
<2de>	548,389	38.8
<2es>	113,953	8.1
<2fr>	45,137	3.2

perform the partitioning in order to investigate the tradeoff between the amount of data and its quality, as in our ranking approach it is assumed that more related sentence pairs receive a higher score. This way, the data sets will contain more noise by increasing N values. As discussed in the previous section, we run the in-domain tests after 4 epochs of adaptation, while the translations on the test sets that are not strictly in-domain are done with systems adapted only for one epoch.

Initially, we run our experiments with the previously used batch size of 80. In this case, however, we observe that for certain translation directions the multilingual system does not perform the target forcing successfully, and translates source sentences to the same language. These instances actually undergo translation, as they do not remain exactly the same on the target side, but the system fails to translate them to the appropriate target language. We overcome this issue by increasing the batch size to 160. In this case, the target forcing works appropriately. A possible explanation could be that this way the system is presented with more instances per batch and thus it is to able to learn the appropriate translation directions more efficiently since the error computation is averaged over more examples. However, this phenomenon does not occur in the case of adapting the system with PP data, even though the batch size was not increased in that case. This can be explained by the larger amount of available data: as there are over 1 million sentence pairs in that case, for the automatic extraction we use partitions with sizes varying between 60,000 and 360,000. Furthermore, the distribution of target forcing tags is even with the WP data, while this does not hold for the PP adaptation set (cf. Table 3.2), making the learning task "less easy". While further investigation would be necessary to study this matter in detail, at this point we can conclude that when the adaptation data is scarce and evenly distributed, ML-NMT systems need to be adapted with larger batch sizes in order to keep the target forcing mechanism functioning appropriately.

The results on in-domain datasets are displayed in Tables 3.3 to 3.7 (arrows represent changes with respect to the baseline performance as long as they are significant at

$p = 0.005$). Generally speaking, for most translation directions and test sets the results are 1–2 BLEU points worse compared to the baseline system's performance, often reaching significant levels. Certain scenarios perform slightly better than the unadapted system, the improvement, however, is usually not significant compared to the baseline. The general trend in case of worsening performances is proportionate to the adaptation datasets' size: as it increases, the results get worse, reaching significant levels of difference for the WP30 in the case of $en \rightarrow es$ and $de \rightarrow en$ on certain test sets (for $de \rightarrow en$ it already happens at WP10 for the *pubPsych titles* test set). The noteworthy exception is the $en \rightarrow fr$ direction on the *pubPsych titles* test set, where the top-5,000 set already drastically worsens the translation quality (more than 3.5 points of drop), and one can observe a decreasing trend in terms of BLEU score as the size of the dataset increases. Conversely, the reverse direction $fr \rightarrow en$ showcases significant improvement on the same test set for WP5.

There is one case, however, when the adaptation with the automatic extractions yields results that are significant improvements with regard to the baseline performance. Interestingly enough, this also happens with the $en \rightarrow fr$ direction, on the abstract test set. In this case, we can observe a steady improvement as the size of the dataset increases. All results overperform the adaptation results with the PP adaptation set at the same number of epochs (that has no data for abstracts for this language pair). On the other hand, the reverse direction $fr \rightarrow en$ does not demonstrate the same behavior on the same test set, as results get worse using this adaptation data, and in this case adaptation with the parallel PP data yields a somewhat better performance (even though it is still worse than the baseline's). These differences, however, are not significant in any of these cases.

The difference between the behavior of titles and abstracts on the $en \rightarrow fr$ direction are explained by their differences: abstracts contain "common" full sentences, while titles are inherently of a different structure. Adapting the system with the PP data allows for using the `<abstracts>` and `<title>` tags, that account for these differences. As these target forcing tags are not available in the WP data, and since its content is more similar to that of the abstracts (full sentences), it improves the results on the corresponding test set, while on the titles set it worsens the performance. Additionally, since the PP set has no data for abstracts in the en–fr language pair, adaptation with WP10 and WP30 overperform the results achieved with PP.

Regarding this set of experiments, we can draw several conclusions. Firstly, as in most

Table 3.3. *Adaptation results (BLEU) for en–es. Best results on each test set are shown in bold (in case of significant improvements).*

en → es

	PP abs. (dev)	*PP abs. (test)*	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	26.91	31.60	42.71	38.50	39.25	22.58
SMT BL	–	32.11	35.22↓	–	–	–
NMT PP	↑28.67	↑34.39	↑44.56	↑**40.02**	↑**41.62**	↑23.09
SMT PP	–	32.83	36.58↓	–	–	–
WP5	↓26.23	31.25	41.05	↓35.89	↓36.49	↓21.74
WP10	26.79	32.09	40.91	↓35.78	↓36.17	↓21.92
WP30	26.72	31.73	↓40.26	↓32.25	↓36.11	22.49
Merge5	↑28.13↓	↑34.02	↑**45.31**	↓36.55↓	↓36.79↓	↑23.49
Merge10	↑28.19	↑33.44	↑44.72	↓37.61	↓38.58↓	↑**23.68**↑
Merge30	↑**28.79**	↑**34.60**	↑44.98	↓36.33↓	↓36.42↓	↑23.43

es → en

	PP abs. (dev)	*PP abs. (test)*	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	25.98	31.20	38.55	40.77	40.25	23.08
SMT BL	–	29.94↓	33.48↓	–	–	–
NMT PP	26.35	31.77	↑40.29	↑**43.16**	41.75	23.43
SMT PP	–	↑31.07	↑35.51↓	–	–	–
WP5	25.52	↓30.12	38.16	↓37.24	↓36.91	↑23.85
WP10	25.45	↓30.35	37.20	↓37.82	↓37.56	↑23.73
WP30	↓24.61	↓30.10	↓35.47	↓37.04	↓36.45	↑**23.94**
Merge5	↑26.57	32.22	↑40.25	↓38.43↓	↓38.05↓	23.49↑
Merge10	↑**26.75**	32.21	↑**40.37**	↓39.45	↓38.67↓	23.36↓
Merge30	26.32	31.91	40.19	↓37.86↓	↓37.52↓	↑23.78↑

cases we cannot achieve significant improvement on the in-domain test sets, it can be concluded that using automatically extracted adaptation sets from *Wikipedia* without any additional clean data is generally not useful for NMT systems, as its quality is not high enough to significantly improve the in-domain performance. It has to noted that in addition to the lower quality, the actual domain coverage of the data does not necessarily align completely with that of the test sets. This is backed up by the fact that adapting with the WP data leads to significant improvements on the out-of-domain *news-test2013* dataset (except for *en* → *es* and *en* → *de*), while on the close-domain EMEA test sets we observe a behavior similar to the strictly in-domain *pubPsych* sets. As we aimed for covering a large number of articles when extracting from the health and psychology domains, the automatic adaptation set contains many parallel sentences that are only slightly related to these topics. Narrowing the search

Table 3.4. *Adaptation results (BLEU) for en–de. Best results on each test set are shown in bold (in case of significant improvements).*

en → de

	PP abs. (dev)	*PP abs. (test)*	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	17.51	11.05	31.23	31.29	31.15	14.35
SMT BL	–	8.33↓	15.92↓	–	–	–
NMT PP	↑19.39	↑**12.16**	↑33.28	↑**32.91**	32.69	14.35
SMT PP	–	↑10.94↓	↑24.12↓	–	–	–
WP5	↓16.42	↓10.04	30.44	↓27.01	↓26.70	↓13.17
WP10	↓16.34	↓9.80	↓28.59	↓27.23	↓26.67	↓13.25
WP30	↓13.09	↓7.76	↓23.12	↓26.37	↓25.95	↓13.12
Merge5	↑18.81↓	↑11.74↓	↑33.60	↓29.17↓	↓28.32↓	14.34
Merge10	↑19.01	↑11.69↓	↑33.89	↓29.60↓	↓29.54↓	14.62
Merge30	↑**19.79**	↑11.96	↑**33.97**	↓26.63↓	↓27.33↓	13.46↓

de → en

	PP abs. (dev)	*PP abs. (test)*	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	24.35	15.80	40.79	36.91	36.10	18.89
SMT BL	–	12.77↓	23.64↓	–	–	–
NMT PP	↑**26.46**	↑**17.11**	41.70	↑**39.36**	↑**38.12**	19.52
SMT PP	–	↑15.61↓	↑32.79↓	–	–	–
WP5	↓22.76	↓15.01	40.59	↓35.48	↓34.28	19.26
WP10	↓23.49	15.12	↓39.24	↓34.78	↓33.66	↑19.37
WP30	↓23.41	15.34	↓37.86	↓34.92	↓33.18	18.39
Merge5	↑25.73↓	↑16.49↓	41.34	↓35.78↓	↓34.84↓	19.26
Merge10	↑25.38↓	↑16.66↓	↑**42.02**	↓36.70↓	↓34.81	↑**19.46**
Merge30	↑26.38	↑16.95	41.44	↓33.85↓	↓33.03↓	19.41

criteria could be a future possibility, however, this could lead to losing a great number of high-quality parallel sentences, and therefore would not necessarily lead to an improvement.

In the case of the close-domain and general test sets additional conclusions can be drawn, since we have available data for language pairs that are under- or zero-resourced in either the PP data set, or even the general NMT system. The *Wikipedia* extractions have a clear positive effect for these language pairs as the adapted systems often overperform the one that has been adapted on the *pubPsych* data. For *es → de*, WP10 yields a significant improvement on the EMEA *dev* set, and all WP sets result in significantly better performance on the *news-test2013* test data. In the case of EMEA *test*, the performance improvements are not significant, but all results are above the

Table 3.5. *Adaptation results (BLEU) for en–fr. Best results on each test set are shown in bold (in case of significant improvements).*

en → fr

	PP abs. (test)	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	19.01	40.00	26.12	26.08	17.56
SMT BL	20.14	22.79↓	–	–	–
NMT PP	↑21.16	↓38.35	↑**28.05**	↑**28.19**	18.46
WP5	↑20.96	↓36.41	↑26.91	25.33	↑18.94
WP10	↑22.41	↓35.39	↑27.17	25.36	↑19.20
WP30	↑**22.87**	↓35.03	↑27.33	25.83	↑**19.28**
Merge5	19.41↓	39.36	↑26.45↓	25.15↓	↑18.38↓
Merge10	18.71↓	38.87	↑27.10	↑26.22↓	↑18.89↑
Merge30	↑21.40↑	38.89	↑27.12↓	↓25.60↓	↑18.94↑

fr → en

	PP abs. (test)	*PP tit.*	*EMEA (dev)*	*EMEA (test)*	*news-test*
NMT BL	24.60	39.09	36.39	34.97	22.62
SMT BL	23.80	36.00	–	–	–
NMT PP	24.89	39.98	37.55	36.18	↑22.74
WP5	23.39	↑**40.84**	↓34.75	33.67	↑23.12
WP10	22.41	39.22	↓34.88	↓32.85	22.85
WP30	23.51	38.44	↓34.78	32.03	↑**23.38**
Merge5	24.6	39.33	34.52	↓32.91↓	22.78↑
Merge10	24.92	38.16↓	↓34.91	33.05	22.51
Merge30	24.25	40.99	↓33.94↓	↓32.87	22.19↓

baseline and the performance with the PP adaptation (which is significantly worse than that of the unadapted system). For the reverse *de → es* direction, however, results on the close-domain test sets are slightly worse than with the PP adaptation, while on the general-domain set WP5 produces a result that is both better than the performance of the system adapted with PP data, and a significant improvement over the baseline.

Improvements are clear and significant for the *fr–es* language pair on the general-domain data, and they are higher than for the systems adapted with PP. The same can be observed for *de–fr*, except for the WP30 set in case of *de → fr*.

In addition to the zero-resourced adaptation directions, the *en–fr* language pair also undergoes significant improvements, similarly to the in-domain test sets. However, as

Table 3.6. *Adaptation results (BLEU) for es–de. Best results on each test set are shown in bold (in case of significant improvements).*

es → de

	EMEA (dev)	*EMEA (test)*	*news-test*
NMT BL	14.44	15.52	6.89
NMT PP	14.65	↓15.34	6.97
WP5	14.86	16.58	↑9.78
WP10	↑15.89	16.84	↑9.73
WP30	14.94	16.39	↑9.94
Merge5	↑**16.38**↑	**17.46**↑	↑10.71↑
Merge10	16.55	17.71	↑10.68↑
Merge30	16.34	17.12	↑**10.76**↑

de → es

	EMEA (dev)	*EMEA (test)*	*news-test*
NMT BL	19.37	20.57	14.42
NMT PP	↑**20.23**	20.90	14.52
WP5	19.83	↓20.55	↑**15.03**
WP10	↑20.02	↑20.65	14.62
WP30	19.28	↓19.45	↓13.98
Merge5	↑20.00↓	↓20.43↓	14.75↑
Merge10	↑19.58↓	↓20.05	14.90↑
Merge30	↑19.65↓	↓20.21↓	14.96↑

Table 3.7. *Adaptation results (BLEU) for es–fr and de–fr. Best results on each test set are shown in bold (in case of significant improvements).*

es → fr		*fr → es*	*de → fr*	*fr → de*
	news-test	*news-test*	*news-test*	*news-test*
NMT BL	17.77	24.92	9.27	6.30
NMT PP	↑19.11	25.33	↑10.40	6.54
WP5	↑**21.30**	↑25.77	↑11.29	↑10.14
WP10	↑**21.30**	↑25.92	↑**10.98**	↑10.01
WP30	↑21.28	↑26.15	8.23	↑9.01
Merge5	↑20.21↑	↑25.48↑	↑10.53↑	↑**10.41**↑
Merge10	↑20.88↑	↑25.82↑	↑10.97↑	↑10.03↑
Merge30	↑20.80↑	↑**26.18**↑	9.78↓	↑9.44↑

in that case, the improvement depends on the particular test set, as it does not occur with the EMEA *test* data. Furthermore, as observed previously, for the $fr \rightarrow en$ direction we cannot talk about such improvements; conversely, most results get significantly worse for close-domain test sets, while there are still significant improvements on the general-domain *news-test2013* data. For other well-resourced language pairs, results significantly worsen on the close-domain EMEA sets, while the out-of-domain test results tend to depend on the exact translation direction.

As for the tendencies regarding the size of the extraction partition, there are again some general trends to be observed, although they do not hold for all cases. It can be said that when results worsen (mostly for language pairs including *en*, except for $en \rightarrow es$), the BLEU score decreases with the increasing dataset size. For language pairs where the adaptation is beneficial, this trend is reversed. In some cases, however, the performance saturates at smaller datasets, and increasing the adaptation set size worsens the results (e. g. $de \rightarrow es$ on *news-test2013*). These trends should align with the quality of the extractions, that can be evaluated via crowdsourcing in future work.

Taking the above observations into consideration, the most important conclusion is that automatic extractions can be beneficial for under- or zero-resourced language pairs. This fact allows for adapting ML-NMT systems on these language pairs by automatically acquiring the adaptation sets and without having parallel data between these languages, that is often one of the main bottlenecks of such scenarios. It has to be pointed out that due to the unavailability of the in-domain test sets for low-resourced pairs we could only test our systems on out-of-domain data. We hypothesize that the same positive trend would hold on in-domain sets as well, and the improvement could be even higher due to the better match between domains of the adaptation and test sets. This should be researched in the future, as zero-resourced in-domain test sets are currently being created.

In addition to this, adapting with automatic extractions can prove useful on general-domain test sets for well-resourced language pairs as long as the quality is high enough (although different translation directions can behave differently within the same language pair). As for the size of the automatically extracted adaptation set, the best approach is using smaller but higher-quality partitions. Even though using more data can further improve the performance, it is not always the case, and when it is, the additional improvement is often negligible. As long as the quality of the data is high, a smaller corpus should be sufficient for adaptation purposes.

3.1.4 Transfer Learning with Combined Parallel and Comparable In-Domain Data

In the final iteration of the domain adaptation experiments the parallel *pubPsych* adaptation set is merged with the *Wikipedia* extractions (Merge). Similar to the previous approach, we adapt the systems with three different partitions of the automatic extractions, which are mixed with the full PP adaptation set (Merge5, Merge10, Merge30).

Results are also shown on Tables 3.3 to 3.7. Here, the arrows after the BLEU scores indicate significant differences ($p = 0.005$) when compared to the adapted results achieved in the PP adaptation scenario, while the ones before the numbers stand for significant differences with respect to the baseline system. For the in-domain test sets in most cases the addition of the *Wikipedia* extractions does not lead to significant differences compared to the results achieved with only the PP adaptation data. In these cases the results are somewhat different, but the differences stay within the significance level. There are exceptions where results with certain partitions are significantly worse than the ones achieved with the PP adaptation data, mostly on the *pubPsych abstracts* test sets ($en \rightarrow es$, *en–de*, $en \rightarrow fr$), as well as on the *pubPsych titles* in the case of $fr \rightarrow en$. However, for the same translation direction, the Merge30 adaptation scenario significantly overperforms the results with PP on the *pubPsych abstracts test* set. This is explained by the fact that in this case, the sole WP adaptations have already yielded superior results to the PP adaptation. As we increase the amount of WP data in the Merge approaches, the benefits of the WP extractions re-emerge (even though they do not reach the same quality as adaptations without the PP data). Increasing the partition size of WP in the Merge scenarios generally gets the results closer to the ones achievable in the PP scenario for well-resourced language pairs on in-domain test sets. In addition to $en \rightarrow fr$, Merge30 also overperforms the PP results for $en \rightarrow es$ (*pubPsych abstracts dev*), although the difference is not significant in this case.

As for the close- and general-domain test sets, it can again be observed that the addition of the automatic extractions is mainly beneficial for zero-shot translation. The most noteworthy translation direction is $es \rightarrow de$. In this case we observe significant improvements on all test sets (around 2 BLEU scores on the *EMEA* sets, and almost 4 points on the *news-test2013* data). These results are higher than that of achieved

by the PP adaptation set, which does not contain this language pair. Furthermore, the translation performance is higher compared to the results achieved with the automatic extractions. This leads to the conclusion that even though the manually created in-domain dataset does not contain all language pairs, mixing it with the *Wikipedia* corpora that has parallel data for every language combination boosts the achievable performance. This phenomenon can serve as another proof for additional data improving the translation performance of other translation directions during adaptation.

The reverse direction $de \rightarrow es$, however, does not behave the same way. While performance improves compared to the unadapted system, they are mostly inferior with respect to the adaptation results using the manual dataset. The only exception is the *news-test* set, where the results are significantly higher compared to the manual adaptation's BLEU score, but still cannot get better than the translation of quality of the system adapted on the top-5,000 *Wikipedia* extractions only, and fail to produce significant improvements with respect to the baseline system.

There are a few additional translation directions and test sets where using the merged data proves to be the best possible configurations regarding the achievable results. Namely, for *en–es*, on the *news-test2013* set we obtain a significant improvement compared to the PP adaptation for a certain partition (WP10). On $fr \rightarrow es$, the best results are obtained in the Merge30 scenario, and all Merge systems perform significantly better than the one adapted only with PP. Still on the same test set, the $fr \rightarrow de$ direction delivers the best performance in the Merge5 configuration, while all Merge configurations significantly overperform PP again. Since the *pubPsych* dataset is a domain-specific one, we achieve better results on general-domain data when combining it with the WP partitions covering broader topics. However, since some of the Merge systems perform better than the ones adapted only with WP data proves that the addition of the in-domain PP adaptation set can still be beneficial on out-of-domain test sets as well.

For *en–es* and *en–de*, results change in a negative direction significantly on the EMEA sets. In the case of *en–fr*, certain scenarios in the $en \rightarrow fr$ direction significantly overperform the baseline system, but they fail to achieve the same improvement as the PP data does, often staying significantly below those results. The pairs are well-represented in the training data for the general NMT system, and noisy automatically extracted data is unable to further improve these results even with the help of manual *pubPsych* in-domain parallel corpora. On the *news-test2013* test set, the best results

are delivered on *en–es* with the Merge10 and Merge30 approaches respectively, that are both significantly better than the results with the PP adaptation set. For the *en–de* pair, the Merge30 result is significantly worse than the one in the PP approach in the $en \rightarrow de$ direction. Other systems for this language pair do not showcase significant differences compared to the PP adaptation scenario (the significant improvement with respect to the baseline is inherited with the Merge10 adaptation for $de \rightarrow en$).

3.1.5 Comparison to State-of-the-Art Results

While the topic of domain adaptation for ML-NMT systems has not been studied in much detail previously, results have been reported for non-multilingual systems. In [Chu et al., 2017], the transfer learning approach similar to ours improves BLEU scores between 1 and 2 points, while the domain target forcing method yields similar results with differences usually being only a few decimals between the two techniques. The results reported in [Freitag & Al-Onaizan, 2016] are approximately 4 points better compared to the baseline after domain adapting NMT systems via transfer learning. Both of these studies achieve the results on the $en \rightarrow de$ translation direction. Our adapted systems perform approximately 2 points better relative to the baseline for this scenario, which is in line with [Chu et al., 2017]. It has to be noted that results are highly dependent on both the data available for training and domain adaptation and the test sets, thus it is not straightforward to draw such comparisons.

3.1.6 Comparison to SMT Systems

As there are general-domain SMT systems available within the CLUBS project for language pairs involving *en*, as well as models trained on in-domain data, we test and compare their performance on in-domain test sets (*pubPsych abstracts test* and *titles*). The general systems have been trained on close- and out-of-domain data (cf. top two blocks of Table 2.4), and the in-domain ones on the PP dataset (bottom block of Table 2.4). The results are shown in Tables 3.3 to 3.5 in the rows SMT BL (baseline) and SMT PP (in-domain), and are repeated in Table 3.8. For SMT systems, arrows in front of the numbers of the SMT PP scenario represent significant changes with respect to the SMT baseline, while the ones after the results indicate significant differences compared to NMT systems within the same scenario ($p = 0.005$).

In case of the abstracts, the SMT baseline systems' performance is generally significantly inferior to the NMT baseline's (-1–3 points of BLEU), except for $en \rightarrow es$ and en–fr. This shows that the NMT baseline is a strong starting system, and that a ML-NMT system has superior performance even with a relatively small vocabulary (80+2 K for all four languages). For the titles test set, the general-domain SMT models provide translations that are of a remarkably worse quality than those of the NMT system. A possible explanation for the higher difference is that NMT architectures are known to be especially good at translating short sentences [Cho et al., 2014a]. The difference for *en–es* is -5–7 BLEU scores, and an even higher -16–17 for *en–de*. In the case of en–fr, there is a noteworthy difference between the two translation directions. The SMT baseline for $en \rightarrow fr$ performs approximately 18 BLEU scores worse than the NMT system, while for the reverse direction $fr \rightarrow en$, this difference is only 3 points, which is the only non-significant case for this test set.

The in-domain SMT systems never achieve better performance than the NMT models after domain adaptation, and their results are significantly worse than that of the NMT approaches in most cases. Moreover, their results lie even below the baseline NMT system's in all but one cases ($en \rightarrow es$, *pubPsych abstracts test*). The achievable performance boosts are significant for *en–de*; in the case *en–es* this only occurs for the $es \rightarrow en$ direction on the *pubPsych abstracts test* set (the only instance where the NMT PP system's improvement is not significant with respect to the NMT baseline). In most cases, using in-domain data improves the SMT systems by 1–3 BLEU scores, which is in line with the NMT system's behavior. However, there is a remarkable positive change on the titles test set for the *en–de* language pair (around 9 points). As the baseline SMT system produces low-quality results in this scenario, the positive effect of the PP data is more pronounced when translating titles between these two languages.

3.2 Reranking with In-Domain Language Models and Similarity Features

As NMT systems are trained by optimizing the log-likelihood of the output given the input, the resulting output's quality in terms of translation evaluation metrics (e. g. BLEU score) will not be optimal. In the following section we examine the possibilities

of selecting the best candidates from the decoder's N-best list, and whether this approach can be applied to this particular case in order to further improve the translation quality.

3.2.1 Oracle Translations

As a first step, we generate an 1000-best list of translation candidates[3], and for each sentence we pick such translations that the sentence-level BLEU score is maximized. After obtaining *oracle translations* this way, we get an insight to the theoretically achievable best performance. We choose the $en \rightarrow es$, $en \rightarrow de$, and $es \rightarrow en$ translation directions and run experiments on the *pubPsych abstracts test* set with the model adapted in the PP scenario, as well as with the baseline models. As for the adapted models, BLEU scores showcase more than 7 points of improvement for $en \rightarrow de$, and an even higher (almost 13 points) for $en \rightarrow es$ and $es \rightarrow en$. The achievable improvements are lower for the unadapted systems (about 2 BLEU scores for $en \rightarrow de$, 6 for $en \rightarrow es$, and 10 for $es \rightarrow en$; cf. Table 3.9).

3.2.2 Reranking Approaches

As the goal is to pick the best candidates without knowing their quality in terms of BLEU score, a model needs to be designed that is capable of adequately finding high-quality translations and getting as close to the theoretical maximum as possible. As domain-specifig language models can successfully be used for in-domain data selection, we can treat the decoder's N-best list in a similar fashion for selecting the best translations. In the following paragraphs we describe and analyze the usability of various approaches within this framework. In all experiments we use 5-gram language models trained on in-domain and general data, made available within the CLUBS project for the purposes of this thesis.

[3]The achievable highest improvements would be possible by increasing the number of n during decoding. However, this also increases computational cost in terms of both memory and runtime. Therefore, we set n=1000.

Table 3.8. *Performance of NMT and SMT systems (BLEU) on in-domain data.*

	PP abs. (test)				*PP tit.*			
	NMT BL	SMT BL	NMT PP	SMT PP	NMT BL	SMT BL	NMT PP	SMT PP
$en \rightarrow es$	31.60	32.11	↑34.39	32.83	42.71	35.22↓	↑44.56	36.58↓
$es \rightarrow en$	31.20	29.94↓	31.77	↑31.07	38.55	33.48↓	↑40.29	35.51↓
$en \rightarrow de$	11.05	8.33↓	↑12.16	↑10.94↓	31.23	15.92↓	↑33.28	↑24.12↓
$de \rightarrow en$	15.80	12.77↓	↑17.11	↑15.61↓	40.79	23.64↓	41.70	↑32.79↓
$en \rightarrow fr$	19.01	20.14	↑21.16	–	40.00	22.79↓	↓38.35	–
$fr \rightarrow en$	24.60	23.80	24.89	–	39.09	36.00	39.98	–

Table 3.9. *Results of the N-best rescoring experiments (BLEU) for adapted (a.) and unadapted (ua.) systems on the pubPsych abstracts test set.*

Method	$en \rightarrow es$, a.	$en \rightarrow es$, ua.	$en \rightarrow de$, a.	$en \rightarrow de$, ua.	$es \rightarrow en$, a.	$es \rightarrow en$, ua.
Baseline	34.39	31.60	12.16	11.05	31.77	31.20
Oracle	47.00	37.40	19.26	13.72	44.45	41.42
PPL	30.69	27.85	9.64	5.09	28.63	25.66
IN-GEN	29.03	28.05	9.66	4.89	28.62	25.15
IN-GEN,SRC+TGT	29.03	28.05	9.66	4.89	28.62	25.15
SIM	30.20	21.02	6.99	2.17	42.66	39.67
SIM+LF	31.02	21.02	6.98	2.14	39.98	38.25
PPL+SIM+LF	30.97	27.85	9.75	5.09	28.63	25.66

In the first approach (PPL), we use the method proposed in [Klakow, 2000]. We use the in-domain language model for obtaining perplexities on each candidate sentence. For each source translation we choose the corresponding target that has the lowest perplexity. The second approach (IN-GEN) is based on the method described by the authors of [Moore & Lewis, 2010]. We obtain cross-entropy scores by using both general (H_{gen}) an in-domain (H_{in}) language models and use their difference for rescoring:

$$score = H_{in} - H_{gen} \tag{3.1}$$

The assumption of this approach is that the best translations should be *like* in-domain sentences while being as $different$ from general domain sentences as possible. The third approach (IN-GEN,SRC+TGT) builds on the second one, this time using the exact method described in [Axelrod et al., 2011]:

$$score = (H_{in}^{tgt} - H_{gen}^{tgt}) + (H_{in}^{src} - H_{gen}^{src}) \tag{3.2}$$

The interpretation of this scenario is that source side sentences should also be taken into consideration. The fourth approach (SIM) is based on the similarity averages rather than language models, similar to the method used for parallel sentence extraction rescoring (i. e. average of complementary similarity features, cf. 2.3.2 for details). We experiment with both including (+LF) and excluding the length factor parameter as an additional term for the average score. Finally, we experiment with combining the scores of the best performing language model approach and the similarity average method (PPL+SIM+LF). In this case, we convert the score given by the LM rescoring to the [0,1] interval in order to comply with the similarity scores.

3.2.3 Results

The results of the various approaches are summarized in Table 3.9. It can generally be concluded that methods including LMs cannot be successfully applied in the case of

these particular translation directions and test sets, as all results lie below the baseline. This is due to the fact that scores based on perplexities/cross-entropies prefer shorter sentences, and select these from the N-best lists. It is also worth pointing out that the inclusion of target-side cross-entropies does not affect the results (IN-GEN and IN-GEN,SRC+TGT).

Among the various rescoring methods, SIM+LF performs the best for $en \rightarrow es$ (although the result is still more than 3 points below the baseline). However, for $en \rightarrow de$, this approach yields the worse results. Looking at the N-best list it is revealed that there are a high number of candidates that are not translated into the correct target language, and appear in *en* on the target side. This phenomenon is similar to the one observed during the WP adaptation experiments with a lower batch size (cf. Subsection 3.1.3). If similarity features are used for reranking, these candidates will receive a higher score due to their being in the same language, which explains the low BLEU scores. Incorrect target language translations appear for both of these directions, however, for $en \rightarrow de$, they occur more frequently, that explains the deteriorating translation quality, while for $en \rightarrow es$ it is able to outperform LM-based methods.

On the other hand, for the $es \rightarrow en$ direction, the approaches based on similarity features yield considerable improvements. The SIM method selects candidates from both the unadapted and adapted models' output that lie only two points below the theoretically achievable best performance, and overperform the baseline results by approximately 8 and 11 points, respectively. Including the length factor (SIM+LF) leads to slightly inferior translation quality, while results are still well above that of the baseline models. The explanation for this behavior is that in this case, there are no candidates in the incorrect target language. This fact leads to the conclusion that if the problem of wrong translation directions is eliminated, similarity features can successfully be used for reranking of N-best translation lists in order to boost the performance both before and after domain adaptation. Investigating all translation directions and overcoming the target forcing issue should be a priority of future research.

Combining the similarity-based reranking with perplexity scores (PPL+SIM+LF) yields results that are in most cases upper bound by the performance of the PPL system (their BLEU scores are a few decimals higher for the adapted $en \rightarrow es$ and $en \rightarrow de$ directions), which assumes a bias towards the LM scores. This could be overcome by introducing weighing factors between the two terms in future work.

Conclusions and Future Research

In this thesis project we investigated ML-NMT systems, focusing on domain adaptation. First, we gave of an overview of the NMT architecture used in this thesis, its multilingual extension, and possible domain adaptation methods. We then described the available resources for the purposes of this project, and proposed an automatic method for acquiring in-domain parallel corpora from *Wikipedia* using NMT context vector embedding similarities, and various complementary similarity measures. We have seen that these features can successfully be used to train supervised classifiers and regression models for extracting parallel sentences from comparable corpora. Due to the nature of the task, our final approach used a regression model trained on continuous similarity labels, and we selected different partitions of extracted sentence pairs, reranked by average similarity feature scores.

We have run several domain adaptation experiments in order to get a clear picture of how the quality and quantity of the adaptation data affects the translation performance of NMT systems. We have drawn several conclusions regarding this question. Firstly, clean parallel adaptation data has a clear benefit for the language pairs it contains, and it can even improve the quality of translation directions that it does not. This behavior is similar to what has been observed regarding zero-shot directions when training NMT systems from scratch. Secondly, we have shown that by using our proposed automatic in-domain adaptation corpus creation method, significant improvements can be achieved for under- or zero-resourced language pairs. Furthermore, the combination of parallel and comparable adaptation data often proved to be useful, and could yield further improvements, especially in cases where the *Wikipedia* extractions alone are beneficial. Finally, we have concluded that translation directions can behave differently within particular language pairs, as well as that the exact behavior of the adapted systems is dependent on the given test sets.

In addition to transfer learning, we also experimented with selecting the best translations from the decoder's N-best list, using language models and similarity features. We have shown that the adapted systems produce higher-quality oracle translations. Methods based on LM scores did not result in improvements for any tested translation direction due to their bias towards short sentences. However, we have shown that using similarity features for reranking can successfully be used as long as the output candidates do not include instances in the source language, as the quality of translations selected this way lied only 2 BLEU scores below the theoretically achievable best performance.

In future work, the quality of the extractions from *Wikipedia* can be further improved by creating training sets that better mirror the nature of the task. Related to this, the quality of the automatic extractions can also be evaluated. As both these tasks would involve extensive human labor, they can be performed via crowdsourcing. Another point to consider is narrowing the domain coverage of the extractions that can boost the performance on in-domain data sets as long as it does not lead to loosing too many high-quality parallel sentence pairs.

Regarding domain adaptation, one other line of investigation to be carried out in the future is testing the adapted systems on in-domain data for language pairs not including *en*. As at the time of writing this thesis these tests are not yet available, our results are only reported on close- and general-domain test sets. The possibilities of maximizing translation quality using reranking also need to be further researched. The priority should be overcoming the issues with incorrect target forcing in the N-best lists, so that other translation directions can also benefit from the remarkable improvements achieved by the similarity-based approach for $es \rightarrow en$. Furthermore, testing on additional translation directions and test sets would also be necessary in order to investigate the full potential of our proposed methods.

In this thesis project, the focus was laid on NMT systems. In future research, the same set of domain adaptation experiments can be repeated with SMT architectures with all adaptation sets and for all language pairs. This would allow for drawing comparisons between the two paradigms with respect to the amount and quality of the adaptation data, and the noise tolerance of the respective approaches. Furthermore, it would make it possible to compare the behavior zero-shot and under-resourced translation directions between NMT and SMT systems during adaptation.

List of Abbreviations

BLEU	Bilingual Evaluation Understudy
CNN	Convolutional Neural Network
BRNN	Bidirectional Recurrent Neural Network
DNN	Deep Neural Network
GRU	Gated Recurrent Unit
LM	Language Model
LSTM	Long Short-Term Memory
ML-NMT	Multilingual Neural Machine Translation
MT	Machine Translation
NMT	Neural Machine Translation
RBMT	Rule-Based Machine Translation
RNN	Recurrent Neural Network
SMT	Statistical Machine Translation

List of Tables

Bibliography

[Axelrod et al., 2011] Axelrod, A., He, X., & Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 355–362). Edinburgh, Scotland, UK.: Association for Computational Linguistics.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations 2015* San Diego, California, USA.

[Barrón-Cedeño et al., 2015] Barrón-Cedeño, A., España Bonet, C., Boldoba Trapote, J., & Márquez Villodre, L. (2015). A factory of comparable corpora from wikipedia. In *Proceedings of the Eighth Workshop on Building and Using Comparable Corpora* (pp. 3–13).: Association for Computational Linguistics.

[Bisazza et al., 2011] Bisazza, A., Ruiz, N., Federico, M., & Kessler, F.-F. B. (2011). Fill-up versus interpolation methods for phrase-based smt adaptation. In *International Workshop on Spoken Language Translation 2011* (pp. 136–143). San Francisco, California, USA.

[Bojar et al., 2016] Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., & Zampieri, M. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation* (pp. 131–198). Berlin, Germany: Association for Computational Linguistics.

[Bridle, 1990] Bridle, J. S. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2* (pp. 211–217). Morgan-Kaufmann.

[Britz et al., 2017] Britz, D., Goldie, A., Luong, T., & Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

[Chen & Eisele, 2012] Chen, Y. & Eisele, A. (2012). Multiun v2: Un documents with multilingual alignments. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)* Istanbul, Turkey: European Language Resources Association (ELRA).

[Cheng et al., 2016] Cheng, Y., Liu, Y., Yang, Q., Sun, M., & Xu, W. (2016). Neural machine translation with pivot languages. *arXiv preprint arXiv:1611.04928*.

[Cho, 2015] Cho, K. (2015). Introduction to Neural Machine Translation with GPUs. `https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-with-gpus/`. [Online; accessed 17-March-2017].

[Cho et al., 2014a] Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (pp. 103–111). Doha, Qatar: Association for Computational Linguistics.

[Cho et al., 2014b] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). 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)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics.

[Chu et al., 2017] Chu, C., Dabre, R., & Kurohashi, S. (2017). An empirical comparison of simple domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 385–391). Vancouver, Canada.

[Cohn & Lapata, 2007] Cohn, T. & Lapata, M. (2007). Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 728–735). Prague, Czech Republic: Association for Computational Linguistics.

[Devlin et al., 2014] Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., & Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1370–1380). Baltimore, Maryland: Association for Computational Linguistics.

[Duh et al., 2013] Duh, K., Neubig, G., Sudoh, K., & Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 678–683). Sofia, Bulgaria: Association for Computational Linguistics.

[Elman, 1991] Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3), 195–225.

[España-Bonet et al., 2017] España-Bonet, C., Varga, Á. C., Barrón-Cedeño, A., & van Genabith, J. (2017). An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *CoRR*, abs/1704.05415.

[Finch & Sumita, 2008] Finch, A. & Sumita, E. (2008). Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 208–215). Columbus, Ohio: Association for Computational Linguistics.

[Firat et al., 2016] Firat, O., Cho, K., & Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)* (pp. 866–875). San Diego, California, USA: Association for Computational Linguistics.

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

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

[Gehring et al., 2017] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research* (pp. 1243–1252). International Convention Centre, Sydney, Australia: PMLR.

[Ha et al., 2016] Ha, T., Niehues, J., & Waibel, A. H. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

[Hochreiter & Schmidhuber, 1997] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

[Jean et al., 2015] Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1–10). Beijing, China: Association for Computational Linguistics.

[Johnson et al., 2016] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., & Dean, J. (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *CoRR*, abs/1611.04558.

[Klakow, 2000] Klakow, D. (2000). Selecting articles from the language model training corpus. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3 (pp. 1695–1698). Istanbul, Turkey: IEEE.

[Koehn, 2004] Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In D. Lin & D. Wu (Eds.), *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 388–395). Barcelona, Spain: Association for Computational Linguistics.

[Koehn, 2005] Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the Tenth Machine Translation Summit* (pp. 79–86). Phuket, Thailand: AAMT AAMT.

[Koehn, 2009] Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

[Koehn et al., 2007] 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., & Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07 (pp. 177–180). Stroudsburg, PA, USA: Association for Computational Linguistics.

[Koehn & Schroeder, 2007] Koehn, P. & Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 224–227). Prague, Czech Republic: Association for Computational Linguistics.

[Lambert et al., 2011] Lambert, P., Schwenk, H., Servan, C., & Abdul-Rauf, S. (2011). Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation* (pp. 284–293). Edinburgh, Scotland: Association for Computational Linguistics.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10 (pp. 707–710).

[Ling et al., 2016] Ling, W., Trancoso, I., Dyer, C., & Black, A. W. (2016). Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 357–361). Berlin, Germany: Association for Computational Linguistics.

[Lu et al., 2007] Lu, Y., Huang, J., & 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)* (pp. 343–350). Prague, Czech Republic: Association for Computational Linguistics.

[Luong et al., 2015a] Luong, M.-T., Pham, H., & Manning, C. D. (2015a). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1412–1421). Lisbon, Portugal: Association for Computational Linguistics.

[Luong et al., 2015b] Luong, T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, (Volume 1: Long Papers)* (pp. 11–19). Beijing, China: Association for Computational Linguistics.

[Manning & Schütze, 1999] Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press.

[Mayor et al., 2011] Mayor, A., Alegria, I. n., De Ilarraza, A. D., Labaka, G., Lersundi, M., & Sarasola, K. (2011). Matxin, an open-source rule-based machine translation system for basque. *Machine translation*, 25(1), 53.

[McNamee & Mayfield, 2004] McNamee, P. & Mayfield, J. (2004). Character n-gram tokenization for european language text retrieval. *Information retrieval*, 7(1-2), 73–97.

[Moore & Lewis, 2010] Moore, R. C. & Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers* (pp. 220–224). Uppsala, Sweden: Association for Computational Linguistics.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., & 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* (pp. 311–318). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

[Pouliquen et al., 2006] Pouliquen, B., Steinberger, R., & Ignat, C. (2006). Automatic identification of document translations in large multilingual document collections. *CoRR*, abs/cs/0609060.

[Rauf & Schwenk, 2011] Rauf, S. A. & Schwenk, H. (2011). Parallel sentence generation from comparable corpora for improved smt. *Machine translation*, 25(4), 341–375.

[Schwenk, 2007] Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*, 21(3), 492–518.

[Schwenk, 2008] Schwenk, H. (2008). Investigations on large-scale lightly-supervised training for statistical machine translation. In *International Workshop on Spoken Language Translation 2008* (pp. 182–189). Honolulu, Hawaii, USA.

[Sennrich, 2011] Sennrich, R. (2011). Combining multi-engine machine translation and online learning through dynamic phrase tables. In *15th Annual Conference of the European Association for Machine Translation (EAMT)* (pp. 89–96). Leuven, Belgium.

[Sennrich et al., 2016a] Sennrich, R., Haddow, B., & Birch, A. (2016a). 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)* (pp. 86–96). Berlin, Germany: Association for Computational Linguistics.

[Sennrich et al., 2016b] Sennrich, R., Haddow, B., & Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers)* (pp. 1715–1725). Berlin, Germany.

[Simard et al., 1993] Simard, M., Foster, G. F., & Isabelle, P. (1993). Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing (Volume 2)* (pp. 1071–1082). Toronto, Ontario, Canada: IBM Press.

[Skadiņa et al., 2012] Skadiņa, I., Aker, A., Mastropavlos, N., Su, F., Tufis, D., Verlic, M., Vasiļjevs, A., Babych, B., Clough, P., Gaizauskas, R., et al. (2012). Collecting and using comparable corpora for statistical machine translation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)* (pp. 438–445). Istanbul, Turkey.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

[Tiedemann, 2009] Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, & R. Mitkov (Eds.),

Recent Advances in Natural Language Processing, volume V (pp. 237–248). Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia.

[Watanabe et al., 2016] Watanabe, Y., Hashimoto, K., & Tsuruoka, Y. (2016). *Proceedings of the 1st Workshop on Representation Learning for NLP*, chapter Domain Adaptation for Neural Networks by Parameter Augmentation, (pp. 249–257). Association for Computational Linguistics.

[Yasuda et al., 2008] Yasuda, K., Zhang, R., Yamamoto, H., & Sumita, E. (2008). Method of selecting training data to build a compact and efficient translation model. In *Third International Joint Conference on Natural Lanugage Processing* (pp. 655–660). Hyderabad, India.

[Zhu et al., 2014] Zhu, X., He, Z., Wu, H., Zhu, C., Wang, H., & Zhao, T. (2014). Improving pivot-based statistical machine translation by pivoting the co-occurrence count of phrase pairs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1665–1675). Doha, Qatar: Association for Computational Linguistics.

[Zoph et al., 2016] Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1568–1575). Austin, Texas, USA: Association for Computational Linguistics.