FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

ASSIGNMENT OF BACHELOR'S THESIS

Title:	Unsupervised machine translation between Czech and German language
Student:	Mgr. Ivana Kvapilíková
Supervisor:	Ing. Daniel Vašata, Ph.D.
Study Programme:	Informatics
Study Branch:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	Until the end of summer semester 2020/21

Instructions

Recent advances in natural language processing enable to perform unsupervised machine translation successfully. The thesis should focus on approaches based on cross-lingual vector representations of words.

1) Review and theoretically describe the state of the art approaches to unsupervised machine translation based on vector representations of words.
2) Implement the most promising one and experimentally evaluate its performance on translation between Czech and German language. Use existing implementations as much as possible.
3) Propose a direction for further improvements.

References

Will be provided by the supervisor.

Ing. Karel Klouda, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague October 3, 2019

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Unsupervised Machine Translation between Czech and German Language

Mgr. Ivana Kvapilíková

Department of Applied Mathematics
Supervisor: Ing. Daniel Vašata, Ph.D.

January 8, 2020

Acknowledgements

I would like to express my gratitude to my supervisor Ing. Daniel Vašata, Ph.D. for his guidance and valuable insight. Furthermore, I would like to thank doc. RNDr. Ondřej Bojar, Ph.D. from the Institute of Formal and Applied Linguistics at Charles University and the members of the research group led by Prof. Eneko Agirre at the University of the Basque Country for their advice. Finally, I would like to thank my family and friends for their support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on January 8, 2020 .....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Kvapilíková, Ivana. *Unsupervised Machine Translation between Czech and German Language*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2020.

Abstrakt

Nedávný výzkum ukázal, že je možné navrhnout překladový systém, který se učí z čistě jednojazyčných textů. Ačkoli kvalita výsledného překladu stále zaostává za standardními systémy trénovanými pomocí textů předem přeložených člověkem, tyto výzkumné snahy otevírají nové možnosti pro datově chudé jazykové páry. Tato práce poskytuje přehled technik pro strojový překlad použitelných právě při nedostatku dat. Nejslibnější přístupy použijeme a porovnáváme jejich výsledky na česko-německém jazykovém páru. Jelikož použité metody závisí na vektorové reprezentaci slov ve vícejazyčném prostoru, zkoumáme tyto reprezentace, abychom ukázali, kolik nesou jazykově neutrální informace.

Klíčová slova neuronový strojový překlad, nesupervizovaný strojový překlad, jednojazyčný korpus, vícejazyčné reprezentace slov, XLM, BERT, česko-německý překlad

Abstract

Recent research has shown that it is possible to design a model that learns to translate entirely from monolingual texts. Even though the translation quality still lags behind the state-of-the art models trained on texts translated by humans, this line of research opens new doors for low-resource language pairs. This thesis provides an overview of unsupervised techniques for machine translation applicable in low-resource conditions. We apply the most promising approaches and compare their performance on the Czech-German language pair. Since the proposed methods depend on vector representations of words in a cross-lingual space, we experiment with these representations to show how much language-neutral information they carry.

Keywords neural machine translation, unsupervised machine translation, monolingual corpus, cross-lingual embeddings, XLM, BERT, German-Czech translation

Contents

List of Figures

List of Tables

Objective of the Thesis

The aim of this work is to explore existing approaches to unsupervised machine translation (MT), perform experiments on the German and Czech data sets, and assess the results. Since the unsupervised MT techniques rely on vector representations of words in a cross-lingual space, a secondary goal is to experiment with these representation to show how much language-neutral information they carry.

This work will provide a theoretical background of machine translation and vector representation of text. It will describe the concept of language model pretraining and analyze the cross-lingual representations hidden in cross-lingual language models. Furthermore, it will give an overview of MT strategies applicable in low-resource conditions. Several of these strategies will be applied to the translation from German to Czech and Czech to German. Both a neural and a statistical model will be trained and their performance will be compared using the BLEU metrics of translation quality. The performance of the unsupervised MT systems will also be assessed against a supervised benchmark.

Introduction

Modern machine translation (MT) heavily relies on parallel corpora, i.e. structured sets of sentence-aligned text documents in different languages, which are used for training the models. However, creating a parallel corpus is an expensive task as the text has to be translated by humans, ideally professional translators. While there are public sources of parallel data for several widely-spoken languages (e.g. EU legislation, public domain books, movie subtitles), many language pairs are so called *low-resource*, which means they have insufficient resources of pre-translated texts.

In contrast to the standard machine translation, unsupervised MT models are trained without any parallel documents, but rather use large monolingual corpora to learn the structure of each language separately. Since monolingual texts are usually easily available, unsupervised techniques are of particular significance for low-resource language pairs.

Significant advancements in the area of machine translation happened for data abundant language pairs (mostly translation to or from English) where large parallel corpora allow training of deep neural networks to translate with impressive results (Wu et al., 2016; Bahdanau et al., 2015; Vaswani et al., 2017). It is only recently that the focus has been turning to low-resource languages and scarce data conditions (Lample et al., 2018b). A recent line of research is exploring the fully unsupervised setting where no parallel data is available at training time (Artetxe et al., 2018c,b; Lample et al., 2018b). In our experiments, we compare several existing approaches to unsupervised MT on translation between German and Czech. We also assess the gap between supervised and unsupervised MT systems.

The key ingredient to functioning of a translation system trained on strictly monolingual data is a high quality semantic representation of the input words in the two languages. In this thesis, we focus on experimenting with these representations and we investigate the effect of generative pretraining on translation quality.

The structure of the thesis is the following. Chapter 1 summarizes the

most important concepts and theoretical building blocks of machine translation in general. Chapter 2 provides an overview of unsupervised techniques applicable in low-resource conditions. Chapter 3 focuses on the benefits of pretrained word embeddings or entire language models for unsupervised MT and Chapter 4 analyzes the internal representations of such pretrained models. Chapter 5 applies the unsupervised MT methods to translation between Czech and German, compares the resulting models and identifies possible directions for future improvements.

Part I

Theoretical Background & Literature Review

Chapter 1

Approaches to Machine Translation

This chapter provides a summary of the theoretical concepts behind machine translation. The two approaches we will be using in this thesis are neural machine translation (NMT) and statistical machine translation (SMT). NMT has recently become the dominant paradigm (Wu et al., 2016), reaching impressive results for many language pairs. However, in situation where not enough data resources are available, the phrase-based statistical machine translation approach still plays an important role (Koehn and Knowles, 2017).

After the initial focus of the MT community on data abundant languages, a recent line of research emerged to explore the extreme setting of unsupervised machine translation trained on strictly monolingual data. Artetxe et al. (2018b) show that the statistical phrase-based approach can be modified to the unsupervised settings and yield competitive results.

1.1 Statistical Machine Translation

A phrase-based statistical machine translation (SMT) model is a log-linear probability model (Koehn et al., 2003) capturing the probability of one sentence being a translation of another one. In order to estimate it, the input texts are split into phrases (n-grams), aligned and organized in a phrase table together with their frequencies of occurrence in the training data set. The following features are statistically measured on the training data to form part of the log-linear model:

- phrase translation probability (favoring phrase pairs which were frequently observed in the data as a mutual translation);
- language model (favoring phrases which sound fluent in the target language, i.e. they were frequently observed in the data);

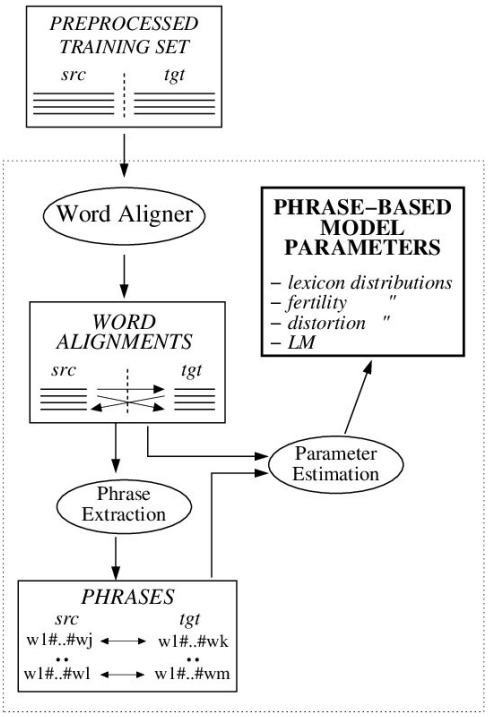

Figure 1.1: Training of an SMT model: estimation of bi-directional word-alignment, phrase extraction, estimation of phrase-based features

Source: Cettolo et al. (2005)

- distortion model (favoring phrases with a natural word ordering);
- word/phrase count penalty (penalizing short phrases and sentences).

Each of the features above is weighted before entering the model. The weights are tuned to maximize translation quality on a small set of parallel sentences (development set). Minimum error rate training (MERT) (Och, 2003) is used for the optimization.

The model can be formalized as follows

$$P(tgt|src) = \frac{\exp \sum_i \lambda_i f_i(tgt, src)}{\sum_{tgt'} \exp \sum_i \lambda_i f_i(tgt', src)} \tag{1.1}$$

where *src* is the original source sentence, *tgt* is the translated target sentence and tgt' iterates over all possible translations. f_is are the features listed above and λ_is are the feature weights. When training the model, we first statistically estimate individual features from the training data set and later tune the feature weights, maximizing translation quality on a development data set.

During decoding (translating), beam search is used to generate the most probable sentence by combining translation candidates for individual phrases based on their log-probability scores.

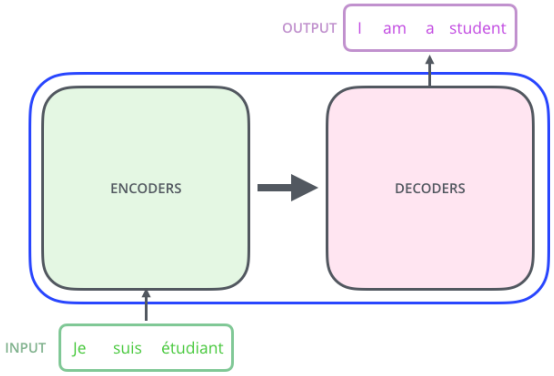

Figure 1.2: Encoder-decoder architecture of NMT
Source: Alammar (2019)

1.2 Neural Machine Translation

Neural machine translation (NMT) models use deep neural networks to find correspondences between the source and the target language (Sutskever et al., 2014). They are trained end-to-end and they are able to exploit distributed representations of text in a continuous space.

NMT models are based on the encoder-decoder architecture illustrated in Figure 1.2. A source sentence is first processed by an encoder which encodes every word to a deep vector representation of several hundred dimensions. The decoder, on the other hand, is trained to generate the target sentence based on the encoded source words. The concept which allows the decoder to attend to particular words of the source sentence while translating is called *attention* (Bahdanau et al., 2015).

The first successful NMT models of Sutskever et al. (2014) or Cho et al. (2014) were based on recurrent neural networks (RNNs). Later proposals experimented with convolutional neural networks (Gehring et al., 2017) and the currently popular Transformer model uses purely attention-based mechanisms with no sequential dependency on previous tokens (Vaswani et al., 2017). The NMT models used in this thesis are Transformer-based and the architecture will be described in Section 1.2.3.

1.2.1 Vocabulary

The first step of training an NMT model is defining the vocabulary. Machine translation is an open vocabulary problem which, however, needs to be solved with a fixed vocabulary size. Sennrich et al. (2016) introduced a method to create a fixed-size vocabulary by splitting words into subwords using the byte-pair encoding (BPE) algorithm. The BPE algorithm, originally designed for

Word	BPE Subwords
Preisgestaltung	Preis@@ gestaltung
Ausländische	Aus@@ ländische
Kellermans	Kell@@ er@@ mans
Mittlerweile	Mittler@@ weile
střešních	stře@@ š@@ ních
Kyrgyzstánu	Kyr@@ gy@@ z@@ stánu
důchodcům	důchod@@ cům
zvířátka	zvíř@@ átka

Table 1.1: A sample of Czech and German subwords generated by the BPE encoding algorithm

data compression, generates the vocabulary by iteratively grouping the most common pairs of characters or subword tokens together and replacing them with a new subword token until a desired vocabulary size is reached. Unknown words are encoded as a sequence of subwords and characters.

In contrast to whole word tokens, subword units reduce the vocabulary size and they eliminate the presence of unknown words in the output translation. They also provide flexibility for translating unfamiliar words composed of familiar word parts which is especially useful for languages which form noun compounds (e.g. German) or inflections (e.g. Czech). Table 1.1 shows a sample of segmented words from the Czech and German training corpora. In some cases, the subword units correspond to linguistic phenomena, in other cases they are just groups of characters.

In the rest of this work, the term *subword* and *word* will be used interchangeably to refer to individual tokens which can either represent full words, subwords or individual characters.

1.2.2 Embeddings

Words are required to have a numeric representation in order to be processed by any machine learning model. The easiest approach is representing each word of the vocabulary with a one-hot vector where all but one elements are zero. However, one-hot encoding generates sparse vectors carrying no semantic information and are unsuitable for further computations. In contrast, neural models are able to learn dense representations, i.e. real valued vectors called *word embeddings*. An interesting property of these representations is that words with a similar meaning have a similar embedding vector (Mikolov et al., 2013c).

The first layer of every NMT system is always an embedding layer. It is implemented as a lookup table where each word is assigned a column (N-dimensional vector) of this table and its values are updated during the NMT training as gradients come from the network. The dimension of the lookup table is $V \times N$ where V is the vocabulary size, N is the embedding dimension

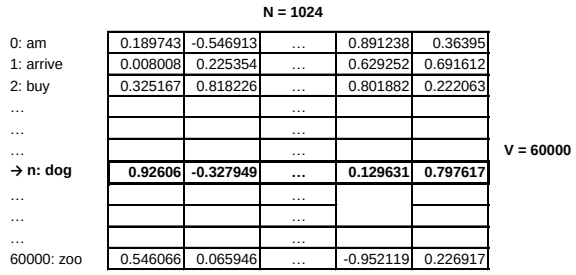

Figure 1.3: Embedding lookup table corresponding to a vocabulary of size V and hidden dimension N. Extracting a word embedding for the word *dog*.

and $V \gg N$[1] see Figure 1.3 for illustration. Embeddings can also be viewed as a mapping from high-dimensional space to lower-dimensional one.

The lookup table can either be initialized randomly or it can be filled with pretrained embeddings. More details on pretrained embeddings will be given in Chapter 3. In some applications, pretrained embeddings are fixed and not updated during training, e.g. Artetxe et al. (2018c).

1.2.3 Transformer Model

Model Architecture

The Transformer NMT model is composed of a stack of encoders and a stack of decoders (Vaswani et al., 2017). The role of the encoder is to process the source sentence and return a deep bidirectional representation vector for each word of the sentence. The role of the decoder is to process the encoded source sentence and produce a correct translation to the target language. In addition to the encoded source words, the decoder also sees the target words it already generated. In Chapter 4, we analyze the vector representations of words generated by individual encoders in the stack.

Each encoder has the following structure (schematically illustrated in Figure 1.4):

- self-attention layer;
- feedforward layer;
- normalization layer;
- dropout.

The **self-attention layer** allows the model to see other words of the sentence when encoding each individual word. The concept of self-attention is described in Section 3.2. In contrast to the encoders, each decoder has an additional attention layer allowing it to also see the words of the source sentence

[1] In our experiments, $V = 60000$ and $N = 1024$

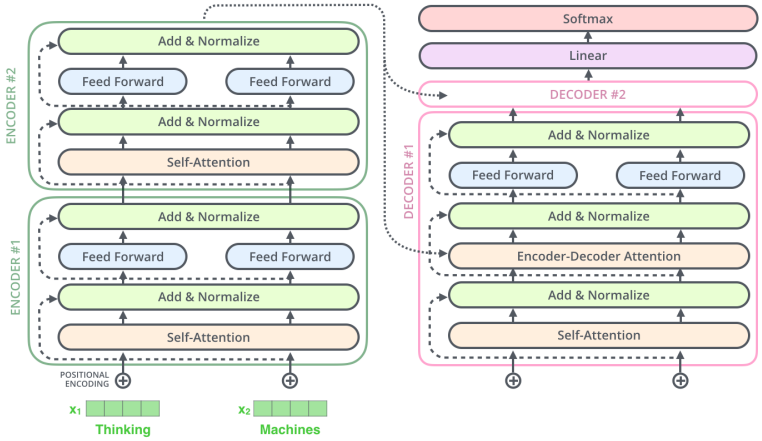

Figure 1.4: Architecture of a Transformer-based NMT model. The arrows illustrate the attention. The number of encoder/decoders is a hyperparameter that is set before the training and can be tuned.

Source: Alammar (2019)

and use that knowledge for translation. Since translations are generated left to right, it is essential that the decoder is only *aware* of the words on the left side of the word being decoded. Word masking is used to achieves this.

The **feedforward layer** is a fully connected layer with ReLU or GELU activations (Hendrycks and Gimpel, 2017).

Layer normalization is applied both after the self-attention layer and after the feed forward layer to reduce training time (Ba et al., 2016).

Dropout is a regularization technique applied after each layer to improve the generalization ability of the model by deactivating a part of the neurons during training (Srivastava et al., 2014).

The entrance layer of both the encoder and the decoder stack is the input embedding layer mentioned in the previous section, associating each vocabulary token with a fixed-size embedding vector. The embedding layer is shared between the encoder and the decoder. To be able to account for the word order of the sentence, each input embedding is enriched with information about its position in the sentence. The positional embedding vector of a token is derived from its position index and is summed with the token embedding vector before being processed by the model (Vaswani et al., 2017).

The final layer of the decoder stack is a linear layer which transforms the decoder output for each word to a V-dimensional vector of next-word probability scores (V is the vocabulary size). The vocabulary index with the highest score corresponds to the most probable word to be generated. The weight matrix of the final linear layer is identical to the input embedding layer and their parameters are shared.

After each training step, the output probability scores are used to calculate

the loss of the model and back-propagate its gradients to the entire model. During inference, the decoder output probability scores are used to predict the following word of the translated sentence. If greedy decoding is used, the vocabulary index with the highest score corresponds to the word which will be generated.

Depending on the design of the model, there can be a separate encoder and a separate decoder for each language. In our models, we only train one encoder and one decoder and share them for both translation directions, i.e. the same encoder is used to encode Czech sentences and German sentences.

More details about the Transformer architecture and the formulas behind can be found in the original paper *Attention Is All You Need* by Vaswani et al. (2017).

1.2.4 Training

Loss Function

NMT models are trained by minimizing the cross-entropy loss function which measures their ability to predict each following word correctly. The model is penalized every time the predicted word is not the correct one.

Cross-entropy loss is defined as follows:

$$H(y,p) = \sum_{i=0}^{V} y_i log(p_i) \tag{1.2}$$

where p is the V-dimensional decoder output and y is the V-dimensional vector with real labels which are equal to 1 if the next word corresponds to the current vocabulary index and 0 otherwise. Therefore, cross entropy penalizes the model for predicting a less-than-one probability for the correct next word.

In order to be able to interpret decoder outputs as word probabilities, they first have to be passed through a softmax function which takes a real-valued vector and transforms it into a non-negative real-valued vector with elements which add up to one. During next-word prediction in NMT, the decoder outputs a score for each token in the vocabulary. The softmax function is used to transform the decoder outputs to probabilities according to

$$p_i = \frac{\exp o_i}{\sum_{j=0}^{V} \exp o_j} \tag{1.3}$$

where p_i is the i-th element of the softmax output and o is the V-dimensional pre-softmax decoder output and V is the size of the vocabulary.

Learning Rate

Neural models are trained by iteratively adjusting their weights according to the gradients of the loss function with respect to these weights. The size of

the adjustment is governed by the learning rate. The initial learning rate is a hyperparameter which is set before the training. A small learning rate leads to slower convergence while a high learning rate may cause the model to diverge (Popel and Bojar, 2018).

When using the stochastic gradient descent (SGD) algorithm, a single learning rate is maintained for all weight updates and it does not change during training. In contrast, the Adam optimization algorithm (Kingma and Ba, 2015) uses adaptive learning rates for different parameters and it is commonly used when training Transformer NMT models.

Batch Size

NMT training is organized in batches. For efficient GPU usage, the model processes several sentences at a time. The size of each batch is another hyperparameter and depends on the available memory. When training a Transformer model, larger batch size leads to better results (Popel and Bojar, 2018).

1.3 Evaluating Machine Translation

1.3.1 BLEU Score

The quality of a machine translation output can be automatically evaluated by the BLEU metrics introduced by Papineni et al. (2002). The metric compares the candidate translation against the reference translation (possibly multiple translations) and assigns a score, depending on the number of overlapping words. Despite having its limitations (due to the large number of ways one can translate a sentence in another language), BLEU has demonstrated a high correlation with human judgment and is widely used to assess results of research in machine translation.

1.3.2 Manual Evaluation

While automatic measures are necessary for the development of machine translation systems, they are only an imperfect substitute for human assessment of translation quality (Koehn and Monz, 2006). When truly evaluating an MT system, manual evaluation is crucial. The clear disadvantage of manual evaluation is that it is expensive, time-consuming, and also subjective.

Evaluators usually assess the translations based on fluency and adequacy on a scale of 0-100. Fluency only reflects how natural the translated sentence sounds, regardless of the original text. Fluency improved significantly with the adoption of neural models which are able to produce very naturally sounding sentences but sometimes hallucinate a new meaning which is not present in the source sentence (Lee et al., 2018). Adequacy, on the other hand, refers to

how accurately the translation captures the meaning of the original sentence and is therefore, in most scenarios, a more reliable metric.

Every year there is a machine translation competition WMT[2] where state-of-the-art translation systems are evaluated both automatically and manually.

[2] http://www.statmt.org/wmt19/

Chapter 2

Machine Translation for Low-resource Languages

According to Ethnologue[3], there are 7,111 languages spoken in the world as of December 2019 and only a small fraction of them is covered by large parallel data sources, others are considered *low-resource*. This work summarizes and compares different approaches applicable in low-resource settings.

2.1 Overview of Methods

2.1.1 Unsupervised Machine Translation

Unsupervised machine translation is the task of performing machine translation without any pre-translated texts available for training. The model learns all necessary information from unrelated monolingual texts.

This novel research area has been explored by Artetxe et al. (2018c,b) and Lample et al. (2018b), who propose both a statistical model, a neural model and a combination of both in order to extract the necessary translation information from monolingual data.

2.1.2 Pivoting

Another option for low-resource languages is to use translation through a third language - *a pivot*. In such a setup, if we are interested in translating Czech-German, we train a Czech-English model and an English-German model. This approach uses the assumption that some languages are mutually low-resource but there might be a third language (such as English) offering richer parallel data resources with the languages of interest. Among disadvantages of this approach are the training cost of two models instead of one and also a

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

more costly inference where we first need to generate the translation to the pivot languages using one model and then process the pivot translation by the second model. This procedure also incurs translation quality loss.

2.1.3 Transfer Learning in Machine Translation

Transfer learning is a general concept in machine learning, referring to the problem of learning some general knowledge when solving one task which could be exploited to improve generalization when solving a different, but related task (Goodfellow et al., 2016).

Transfer learning is relevant for NMT in low-resource settings where it could leverage the knowledge learned when training to translate for a different language pair (Zoph et al., 2016). Kocmi and Bojar (2018) suggest a simple transfer learning method, where they first train a *parent* model for a high-resource language pair of choice and then simply replace the training corpus with the small parallel data of the language pair of interest. They show that the *parent* model pretraining significantly improves over the baseline trained on the low-resource data pair only. Interestingly, they observe improvements even for unrelated languages with different alphabets.

2.1.4 Zero-shot Machine Translation

Zero-shot learning is a special case of transfer learning and generally refers to solving a task without having received any training examples of that task, only by exploiting additional information from the training (Goodfellow et al., 2016). In the NMT setup, it refers to training a multilingual model and translating between language pairs the model never saw during training. For example, we train a multilingual model on German-English and English-Czech translations. If zero-shot is possible, it means that we are able to use such a model for translating between German and Czech. A successful zero-shot application means a proof that the deep neural models are able to learn some sort of an interlingua - language-independent latent representation of meaning (Lu et al., 2018).

There have been successful attempts at multilingual translation systems which translate between multiple language pairs via a shared encoder (Johnson et al., 2017). The architecture of such models is unchanged from a standard one-to-one setup, but they are trained on parallel data for several language pairs and an extra token is added to each input sentence indicating the target language we wish to translate to. Johnson et al. (2017) or Gu et al. (2019) show that training one model for several languages yield comparable or even superior results for some language pairs. Furthermore, they show that a zero-shot translation is possible, albeit yielding lower quality translations than simple pivoting.

2.2 Unsupervised Statistical Machine Translation

Unsupervised SMT is a phrase-based model trained only on monolingual data. As described in Section 1.1, the traditional SMT model has several parts: phrase table, language model, distortion model and word penalties. When only monolingual data is available, we can still estimate the language model without any limitation, as it only depends on monolingual data. We can also calculate the penalties, which are parameterless and the distortion model can be disregarded in the first step (Artetxe et al., 2018b). The key element is populating the initial phrase table from monolingual data only and then tuning weights of the SMT model.

The following concepts are crucial for the estimation of the unsupervised SMT model

1. initial phrase table population;
2. unsupervised tuning;
3. back-translation.

2.2.1 Initial Phrase Table Population

The first step towards unsupervised statistical machine translation, as suggested by Lample et al. (2018b) and Artetxe et al. (2018b), is training monolingual phrase embeddings, aligning them to a cross-lingual embedding space and using them to infer a bilingual lexicon. Details on unsupervised learning of the alignment will be given in Section 3.1.2.

The resulting bilingual lexicon allows us to derive the initial phrase table for the SMT model. As shown by Artetxe et al. (2018c) and Lample et al. (2018a), approximate translations inferred from cross-lingual embeddings can be used to populate the initial phrase table of a SMT system. A practical example is illustrated in the second part of this thesis in Section 5.2.1.

2.2.2 Unsupervised Tuning

In standard SMT, the weights of the log-linear model are tuned on a small parallel data set using MERT. Since parallel data is not available in the unsupervised setting, we create a parallel development set artificially by using the existing model to translate a small set of monolingual data. When MERT finds the optimal weights, we use the optimized model to create another synthetic development set for optimizing the reverse model. The optimized reverse model is used to again create a synthetic developments set for optimizing the first model on better data. This iterative procedure is repeated until convergence. The algorithm is described in detail in Artetxe et al. (2018b).

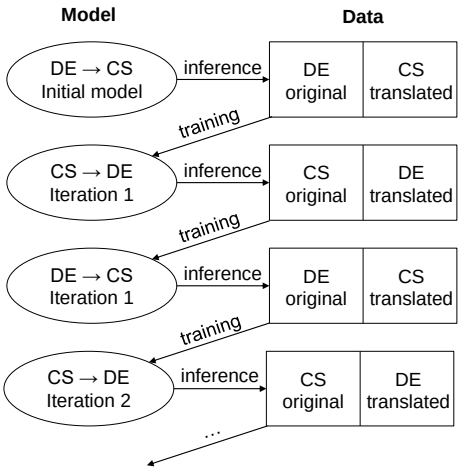

Figure 2.1: Step-by-step illustration of iterative back-translation

Source: Kvapilikova et al. (2019)

2.2.3 Back-translation

The idea of back-translation was first introduced by Sennrich et al. (2016) as a method to leverage monolingual data during MT. When we have an existing SMT model for both translation directions, we can use it to translate both of our monolingual corpora and generate two synthetic parallel corpora. At that point we can discard the existing models and train new ones from scratch. The new SMT models can be trained in a standard supervised way, using the synthetic corpora for supervision. This procedure can be repeated several times, creating synthetic corpora of increasing quality. The translation quality increases over several iterations of back-translation (Artetxe et al., 2018b). The procedure is illustrated in Figure 2.1.

2.3 Unsupervised Neural Machine Translation

Unsupervised NMT is an encoder-decoder neural model trained only on monolingual data. The models of Lample et al. (2018b), Artetxe et al. (2018c) have a RNN architecture, the model of Lample and Conneau (2019) is Transformer-based. They all have a shared encoder which is a requirement for producing language independent representations of the input text. Lample and Conneau (2019) show that the decoder can be shared as well.

Model initialization plays an important role in unsupervised MT. The training is performed iteratively on two sub-tasks: recovering the input from a noisy version of itself (de-noising) and recovering the input from a synthetic translation (back-translation). The training pipeline consists of switching between these two sub-tasks, one batch each in every step.

2.3.1 Model Initialization

The importance of model initialization in unsupervised NMT indicates that there is a lot to be gained from careful model pretraining. Lample et al. (2018b) initialize the embedding layer of their unsupervised NMT model with pretrained cross-lingual word embeddings. Lample and Conneau (2019) take this idea even further by pretraining the entire encoder and decoder with a cross-lingual language model. Fine-tuning the pretrained model with the iterative training process for unsupervised MT described in the following paragraphs brings state-of-the-art results for unsupervised MT.

Different methods of pretraining cross-lingual embeddings or entire language models will be described in Section 3.1.2

2.3.2 De-noising

De-noising is a monolingual training objective teaching the model to recover corrupted sentences. The training data is created by adding noise to the input sentence. The noise is added by randomly shuffling words within a predefined window. The de-noising training step is conceptually equivalent to a translation training step; we are essentially translating from a noisy source sentence to the original source sentence.

De-noising helps the MT system to learn to generate proper sentences in a given languages and it is especially important in the beginning of the training when there is not enough cross-lingual information for actual inter-language translation.

2.3.3 Back-translation

Back-translation was already mentioned in Section 2.2 in relation to SMT. It brings significant improvements in translation quality when the training data is augmented by additional (synthetic) sentence pairs created by translating monolingual data with the NMT model which is currently being trained. This procedure is crucial for unsupervised NMT where we do not have any authentic parallel data available at all. Back-translation is happening "on-the-fly" during training where the model first generates a batch of synthetic parallel data and immediately trains itself on it.

In the back-translation step, the model is first set to the inference mode a used to translate a batch of sentences. The synthetic translations serve as source sentences for a training step where the target side is the original sentence.

2.4 Unsupervised Hybrid Machine Translation

Previous work in the area of unsupervised MT shows that combining features of both statistical and neural modeling can have a positive complementary effect on translation quality (Marie et al., 2019; Stojanovski et al., 2019). As described in Section 2.2, SMT relies on the rotation of monolingual embedding spaces to induce a seed bilingual lexicon. Therefore, it can be effectively used to bring the initial cross-lingual signal to the final translation system. However, NMT is superior over SMT in terms of translation fluency (Popovic, 2017). SMT outputs are composed of n-grams and cannot be as fluent as NMT outputs which are generated with the knowledge of a full sentence context.

In the hybrid setting, a seed SMT model is estimated as described in Section 2.2 and used to translate the monolingual corpus. Synthetic translations produced in this manner are used to bootstrap the training of a neural model. In later stages of the training, the synthetic data can be augmented with the translations produced by the hybrid model itself. As suggested by Artetxe et al. (2019), the whole procedure can be repeated several times, iteratively generating synthetic data of increasing quality as the underlying SMT and NMT systems improve.

Chapter 3

Unsupervised Pretraining

The previous chapter showed that pretraining plays an important role in unsupervised MT. The following paragraphs will provide an overview of unsupervised pretraining methods which are available in natural language processing and applicable to MT in particular.

3.1 Pretrained Word Embeddings

As mentioned in Section 1.2.2, the embedding layer of an NMT system can be initialized using pretrained embeddings. This technique is especially effective in low-resource scenarios when training with small amounts of data and it is absolutely essential in scenarios where no parallel data is available at all. Similarly, pretrained embeddings are used to initialize the phrase table of an unsupervised SMT system described in Section 2.2.

Available embedding models include Word2Vec (Mikolov et al., 2013c) or GloVe (Pennington et al., 2014).

3.1.1 Monolingual Embeddings (Word2Vec)

The Word2Vec model (Mikolov et al., 2013a) is a popular solution for the word representation task as it was shown to have many favorable properties (e.g. adding, subtracting of words) as shown below in Figure 3.1. The underlying idea is that similar words tend to appear in similar context and thus have similar embedding vectors (as measured by cosine similarity).

Word2Vec is a two-layer neural network which is trained on raw text data to reconstruct linguistic contexts of words. There are two methods how to design and train a model - Continuous Bag of Words (CBOW) and Skip-Gram. The former model learns to predict the current word based on its context (surrounding words) while the latter learns to predict the surrounding words given the current word. The architecture is illustrated in Figure 3.2.

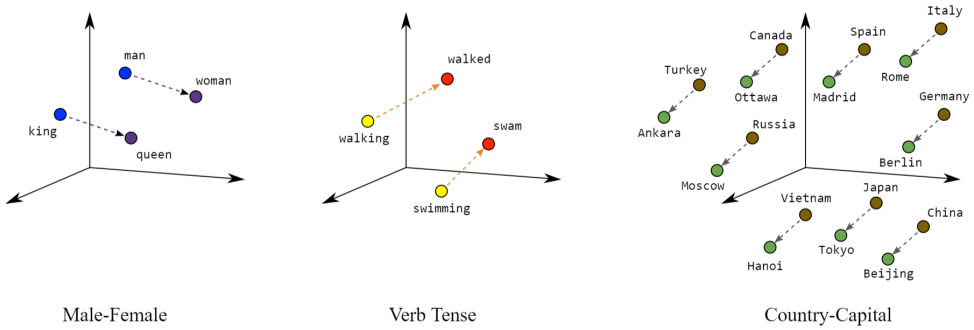

Figure 3.1: Relations between Word2Vec embeddings reflect semantic and grammatical notions between the embedded words. Mathematically, the leftmost graphics can be described with the popularized formula of $king - man + woman = queen$ which holds for the Word2Vec vectors.

Source: Google (2019)

We are not interested in the solution of the task itself. However, the model has to create useful internal representations to be able to solve the task and these representations serve as our word embeddings. They are stored in the hidden layer of the model and the word embeddings are obtained by simply taking the hidden weights.

While embeddings of entire words are useful for semantic processing and tasks such as word similarity search, other tasks, such as machine translation, operate on subwords. FastText by Bojanowski et al. (2017) extends the Skip-Gram training to subword units.

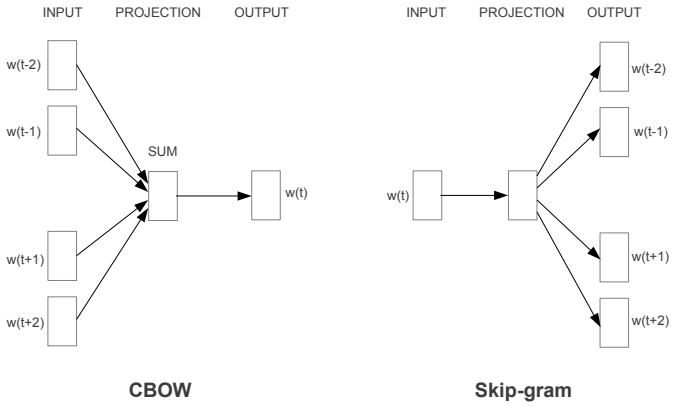

Figure 3.2: Word2Vec model architectures. The CBOW architecture predicts the current word based on the context, the Skip-gram predicts surrounding words given the current word.

Source: Mikolov et al. (2013a)

3.1.2 Cross-lingual Embeddings

For unsupervised MT, we need to project embeddings to a cross-lingual space where similar words have similar representations regardless of their language. Therefore, a cross-lingual extension of Word2Vec is necessary. We present two unsupervised strategies for projecting words to a cross-lingual space.

Aligning Monolingual Embedding Spaces

One option is to train monolingual Word2Vec embeddings individually as described in the previous Section and subsequently align them. The aligned spaces can be directly used for inferring a bilingual lexicon. This method is based on the idea that embedding spaces of different languages are approximately the same and there exists a linear mapping between them (Mikolov et al., 2013b), as illustrated in Figure 3.3.

While there is a range of supervised methods to learn the mapping, other approaches are completely unsupervised. Conneau et al. (2018) use adversarial training to align monolingual word embedding spaces and infer a bilingual lexicon without parallel data (MUSE). This method is particularly effective in favorable conditions of related languages. In order to overcome this restriction, Artetxe et al. (2018a) use self-learning to map monolingual embeddings into a shared space (VecMap). This approach exploits the structural similarity of the embeddings which holds even for distant languages and iteratively improves upon it by self-learning.

The procedure can be extended to n-gram embeddings where we learn a mapping between word sequences. This method is used to initialize an unsupervised SMT system described in Section 2.2.

Shared Vocabulary

The second option is to simply train FastText subword embeddings on a non-aligned multilingual corpus created by concatenating the two monolingual corpora. The requirement for this method is that the two languages have a

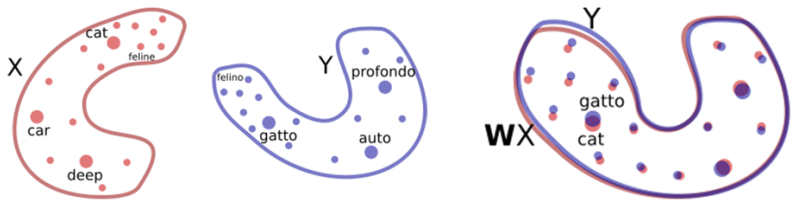

Figure 3.3: Mapping monolingual embeddings to cross-lingual space

Source: Conneau et al. (2018)

common alphabet and the subword vocabulary be generated from the multilingual corpus to be shared for the two languages. While there is no explicit objective for the model to learn a joint embedding space instead of two sub-spaces, (Smith et al., 2017) show that identical character strings (proper names, digits) serve as anchors and force the representation spaces together.

Lample et al. (2018b) use the cross-lingual embeddings to initialize the embedding layer of the unsupervised MT model and report better results than other work (Lample et al., 2018a) relying on aligned monolingual spaces and inferred bilingual dictionaries.

3.2 Pretrained Language Models

Language models tell us how likely is a given sequence of words to occur in a language. They play an important role in MT systems in generation of the translated sentence. SMT systems include an explicit language model as part of the log-linear model introduced in Section 1.1, which favors translations with high LM log-probabilities. Encoder-decoder NMT architectures include a language model implicitly as they have the capacity to learn the same information as a language model themselves during the MT training (Sennrich et al., 2016). Lample and Conneau (2019) show that initializing an NMT model with a pretrained language model leads to a higher translation quality and faster convergence.

Pretrained language models can be used to initialize models and improve their performance on a variety of natural language understanding tasks (Devlin et al., 2018). The goal of unsupervised pretraining is to use abundant unlabeled data to learn a general structure of the data. Specifically, languagei models learn deep bidirectional representations which must carry information on the syntax and the semantics of text. However, these representations are not easily interpretable by humans.

Pretrained models can be fine-tuned to a specific task without modifying the supervised learning objective. There are several benefits to unsupervised pretraining (Erhan et al., 2010):

- the models are pretrained on large unlabeled data which are abundant, as opposed to expensive labeled data;
- it is less computationally expensive to fine-tune a pretrained model for a specific task than to train the model from scratch;
- finding a proper initialization point can *lock* the training in a region of the parameter space that is essentially inaccessible for models that are trained in a purely supervised way;
- pretraining acts as a regularizer, enabling better generalization in deep neural models.

While pretrained models have been widely used in image processing for some time, they overtook the NLP field only in 2018 with the introduction of BERT. Devlin et al. (2018) and Conneau et al. (2018) show that using pretrained language representation models and only fine-tuning them for particular NLP tasks leads to state-of-the-art models for a wide range of tasks, such as question answering, text classification and language inference. The final model can be built without substantial architecture changes. Adding just one additional classification layer can transform BERT to a universal classifier to detect paraphrases or identify named entities (e.g. geographical or proper names) in text. Pretraining a shallower model and copying it to both the encoder and the decoder can be used to intitalize a NMT model. More details on initializing an NMT model will be given in Section 3.1.2.

Since the textual input is sequential, it can be processed by a RNN architecture where the contextual information from past inputs is modeled with the help of recurrent connections. Bidirectional RNN network can even model both left and right context (Arisoy et al., 2015). Alternatively, the Transformer architecture can be used for language modeling. Devlin et al. (2018) showed that the Transformer encoder described in Section 1.2.3 can be used as a very powerful language representation model and gave rise to the famous BERT.

3.2.1 Monolingual Pretraining (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a pretraining language model developed by researchers at Google (Devlin et al., 2018). It has a Transformer architecture with 12 (24) encoder layers and 110M (340M) parameters in its base (large) version. The English BERT was trained on a total of around 3,300M words extracted from the BookCorpus[4] and English Wikipedia. Google made several pretrained model available for download[5], including a multilingual model trained on 104 languages. Fine-tuning a pretrained BERT (or one of its offshoots) yields impressive results in many downstream NLP task[6] and requires substantially less computation resources than training from scratch.

Training

Transformer-based language models can be effectively trained using the *masked language model (MLM)* training objective (Devlin et al., 2018), illustrated in Figure 3.4. In contrast to a left-to-right language modeling objective, MLM

[4] `https://yknzhu.wixsite.com/mbweb`

[5] `https://github.com/google-research/bert`

[6] General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems `https://gluebenchmark.com/leaderboard`.

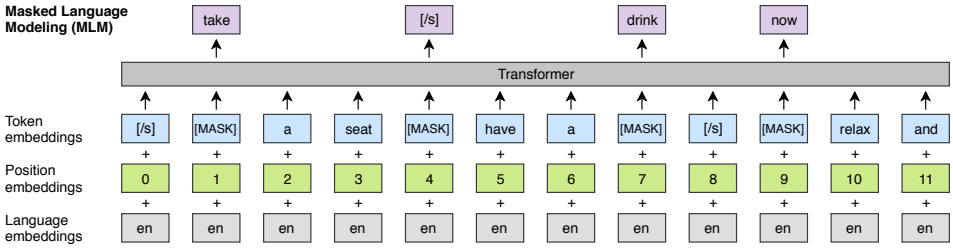

Figure 3.4: Cross-lingual language model design for training with the Masked Language Modeling (MLM) objective

Source: Lample and Conneau (2019)

allows the model to see the context from both sides of the predicted word. Random tokens of a word sequence are masked and the model is trained to fill in the missing words given the context. The final encoder outputs corresponding to the mask tokens are fed into an output softmax layer transforming the encoder vector representations to probabilities over all words of the vocabulary. This final step is common to all standard LMs.

By training the model with the MLM objective, it learns interesting statistical properties about the language which are stored in its hidden vector representations. These vectors can be extracted as contextualized word embeddings and used for other downstream NLP tasks. More details on contextualized word embeddings will be given in Section 1.2.2.

Self-attention

As mentioned in Section 1.2.3, each encoder includes a self-attention layer which helps it look at other words in the source sentence as it encodes a specific word. The encoding of each word thus carries information about the surrounding words and learn the structure of the language.

For example, as illustrated in 3.5, when encoding the word ”it”, the model is *paying attention* mostly to the words ”the” and ”animal”. This knowledge can be used later e.g. for machine translation where it would allow the model to translate the pronoun correctly in the target language (e.g. as ”to” (neuter) rather then ”ten” (masculine) or ”ta” (feminine) when translating to Czech). Such enriched encoding is then fed through a feed-forward layer and passed on to the following encoder.

In Transformer, all these dependencies are handled by the self-attention layers. Transformer models have several self-attention *heads* in each layer and each head is attending to a different part of the sentence. For more details refer to Vaswani et al. (2017).

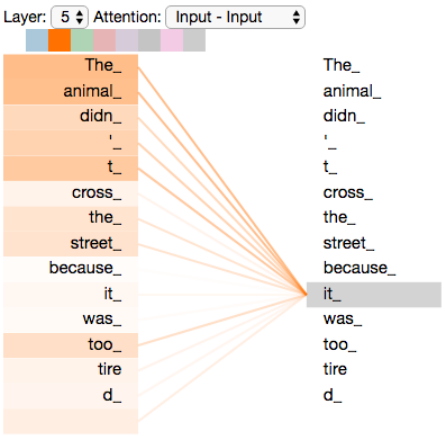

Figure 3.5: Self-attention: when encoding the word "it", the model is *paying attention* mostly to the words "the" and "animal", allowing it to translate the pronoun correctly in the target language

Source: Alammar (2019)

BERT Representations

Encoder hidden states extracted from BERT-like models are sometimes called *contextualized word embeddings*. Like Word2Vec embeddings, BERT embeddings carry information about the usual context of each word. BERT embeddings, however, also change according to the context they word appears in (are contextualized).

Each word (or subword) of the vocabulary is tied to one input embedding vector. As the vector is passed through the encoder, it is enriched with information about its context, position etc. Experiments show that different encoder layers represent different linguistic phenomena, similarly to the image processing models where the shallow layers represent generic features such as edges or lines, later curves, and the deepest layers learn to distinguish specific features such as a human face (Erhan et al., 2010).

When encoding each individual word, BERT sees the context of the entire sentence and enriches the word vector with the information about the other words. In contrast to Word2Vec, which provides one vector per word, there are several BERT vectors for each word and the vector changes when the word is used in a different sentence. The ability of BERT to model how word meaning varies across linguistic contexts (i.e., polysemy) makes them superior to *non-contextualized* embeddings in many applications. For example, unlike Word2Vec, they allow us to distinguish homonyms (e.g. the word "bank" is represented differently when it is used in the context of "river bank" vs. "world bank").

Contextualized embeddings cannot be directly used in the embedding layer

of the NMT models because the embedding lookup table only has one vector per word and there are infinitely many embedding vectors per one word. However, the entire language model can be used to initialize a NMT system. Initializing our translation models with a pretrained multilingual LM is among the experiments that we conduct.

The following Section will show that BERT can also be trained in a multilingual setting, giving rise to embeddings which exhibit some cross-lingual features which will be analyzed in the second part of this thesis.

3.2.2 Cross-lingual Pretraining (mBERT and XLM)

Asides from the vanilla BERT model of the English language, Devlin et al. (2018) released a multilingual model (mBERT) trained on non-aligned Wikipedia dumps in 104 languages. Similarily, Lample and Conneau (2019) trained a Transformer-based multilingual model and called it XLM. The architecture is identical to the monolingual BERT but the models are trained on streams of sentences in different languages using the MLM objective. Although there is no cross-lingual training objective and no explicit alignment, these models learn joint multilingual representations.

Several authors (Pires et al., 2019; Karthikeyan et al., 2019) analyzed how multilingual these representations really are. Although the training does not require any parallel data, mBERT and XLM prove surprisingly good at cross-lingual transfer to NLP tasks such as cross-lingual natural language inference (XNLI)[7] (Pires et al., 2019). Fine-tuning a pretrained XLM model, Lample and Conneau (2019) reach state-of-the-art performance on both XNLI and unsupervised machine translation. While Pires et al. (2019) suspect that the cross-lingual ability of mBERT is linked to the lexical overlap between related languages, Karthikeyan et al. (2019) show that the transfer exists even for languages with different alphabets and with no lexical overlap at all, suggesting that the cross-lingual ability arises rather due to the structural similarities of languages.

Our experiment with XLM representations of Czech, German and English sentences is presented in Chapter 4. We use XLM to initialize an NMT model in Chapter 5.

[7] https://www.nyu.edu/projects/bowman/xnli/

Part II

Experiments & Results

Chapter 4

Analyzing Cross-lingual Representations

Neural models clearly yield impressive results in the area of machine translation but it is still not clearly understood what they learn. The improvement in accuracy and performance came at the cost of our understanding of the system. We understand the architecture and the mechanics of the model, but it is up to the model to decide how to efficiently store information and represent the data. Therefore, exploring the structures that the model learns and assessing the representations is an active research area not only in NLP (e.g. International Conference on Learning Representations [8] or BlackBoxNLP[9] workshop).

Since the topic of this thesis is unsupervised machine translation, we focus on cross-lingual knowledge of the models gained exclusively from monolingual data. We dissect the pretrained language model used to successfully initialize unsupervised MT systems and assess how much cross-lingual information is hidden in its internal representations on different layers. Since the model is trained in a completely unsupervised way, any evidence of cross-lingual transfer is surprising. In this Chapter we present the results of our analysis of the internal representations of the mBERT multilingual model. Similarly to Pires et al. (2019), Karthikeyan et al. (2019) or Libovický et al. (2019), we ask ourselves the following question:

How multilingual is the multilingual BERT?

Previous research has shown that pretrained multilingual models exhibit a strong ability to transfer knowledge from one language to another. Lample and Conneau (2019) show that on the task of language inference (XNLI) where they fine-tune the model on English data and evaluate on data in other

[8] `https://iclr.cc/`

[9] `https://blackboxnlp.github.io/`

languages. In our experiment, we explore the multilinguality of mBERT by assessing its representations directly on a task of *parallel sentence matching*.

4.1 Experiment Design

The task of *parallel sentence matching* is defined as follows – find a correct translation of a sentence in language *L1* from a pool of sentences in language *L2*. This task is substantially easier than machine translation, since the model has to pick a sentence out of a pool of possible translations, it does not have to generate it from scratch. Furthermore, the task is made even easier by the fact that the correct translation is always there. However, the capability of a model to perform this task still means that it learned some cross-lingual knowledge during the unsupervised training which can be leveraged later during fine-tuning on any downstream task, e.g. unsupervised MT.

We use the pretrained model to encode a set of 3K parallel sentences in Czech, German and English and observe, how distant are the sentence embedding vectors of equivalent sentences (in different languages).

We test the following hypothesis:

The nearest neighbor of each encoded sentence is its translation.

4.2 Model Details

We download the pretrained mBERT (`bert-base-multilingual-cased`) model published by Google. The mBERT model was trained on 104 languages. It has 12 encoder layers, 12 attention heads and the embedding dimension is 768. In total, the model has around 110M parameters.

Since there are 12 layers plus the input embedding layer, there are 13 representation layers we can look at when analyzing mBERT. At each layer, the model outputs one vector per input token and this vector can be interpreted as a word embedding.

We derive sentence embeddings from word (subword) embeddings by simply averaging them. Even though mean-pooling seems like a naive approach, it is often used for its simplicity and gives more reasonable results than max-pooling (Pires et al., 2019).

$$E_{sent} = \frac{\sum_{i=0}^{k} E_{word_i}}{k} \tag{4.1}$$

where S is a sentence embedding, W is a word (subword) embedding and k is the number of subwords in a sentence (excluding the special `[CLS]` and `[SEP]` tokens marking the beginning and the end of a sentence).

We process all data by mBERT and we keep all interim outputs. The hidden encoder outputs represent word embeddings on different layers which

cs	Objevená pravicově extremistická síť odhaluje však rozsah, jehož dimenze už dávno nelze dohlédnout.
de	Das aufgedeckte rechtsextremistische Netzwerk offenbart jedoch Ausmaße, deren Dimension noch längst nicht absehbar ist.
en	The right-wing extremist network that has now been discovered, however, is on a scale that has not yet been fully understood.

Table 4.1: Data excerpt from the newstest2012 data sets in Czech, English and German

must be averaged to obtain sentence embeddings. For each sentence embedding in language L1, we retrieve the nearest neighbor in languages L2 and L3 and calculate the ratio of sentences where the nearest neighbor corresponds to the correct translation from the set. The nearest neighbor is retrieved by selecting the candidate with the highest cosine similarity (lowest cosine distance) defined as

$$cossim(E1, E2) = \frac{E1E2}{|E1||E2|} = \frac{\sum_{i=0}^{n} E1_i E2_i}{\sqrt{\sum_{i=0}^{n} E_i}\sqrt{\sum_{i=0}^{n} E_i}} \tag{4.2}$$

$$cosdist(E1, E2) = 1 - cossim(E1, E2) \tag{4.3}$$

where *E1, E2* are sentence embeddings and n is the dimension of the embedding ($n = 768$ in mBERT).

4.3 Tools

The Transformers[10] library is used for the PyTorch implementation of the BERT model. The FAISS[11] library is used for efficient similarity search. TensorBoard is used to visualize embeddings. Data preprocessing is handled by standard Moses scripts[12].

4.4 Data

We use the Czech, English and German versions of the multi-parallel data set `newstest2012` from the WMT workshop. The data set consists of 3,003 sentence triplets, see Table 4.1. The texts belong to the domain of newspaper articles.

The pretrained mBERT model had been trained (by Google) on the entire Wikipedia dump (excluding user and talk pages) for each of the 104 languages selected for training.

[10] `https://github.com/huggingface/transformers`
[11] `https://github.com/facebookresearch/faiss`
[12] `https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer`

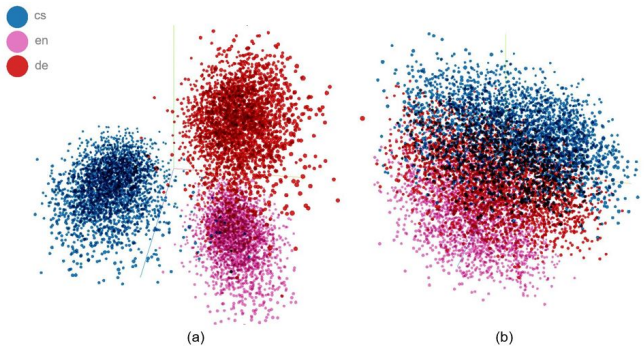

Figure 4.1: Visualization of embedding spaces at the first layer (a) and at the fifth-to-last layer (b) of mBERT. The space was reduced by the PCA algorithm, reducing it to the top three principal components, describing the total variance of 19.5 %.

We performed the following preprocessing steps before encoding the sentences: normalize punctuation, remove non-printable characters, tokenize. The conversion to subword units is handled by the pretrained BERT tokenizer implemented in the Transformers library. Since mBERT was trained on cased data, we keep the capital letters in the sentences.

4.5 Results

Our experiments show that the extent to which we are able to detect cross-lingual structures in mBERT differs across layers and the representations seem to be most language-agnostic in the middle layers of the model. The shallow layers are probably too close to the input to be able to encode the complex structure to a cross-lingual space and are representing the sentences in separate subspaces depending on the source language, as illustrated in Figure 4.1. On the other hand, the deeper layers are close to the final softmax layer and usually learn task-specific knowledge at the expense of general knowledge about the language.

The concept of *language-agnosticism* or *language-independence* describes how much the representations depend on the meaning of the represented sentences and not on their original language. Several examples of embeddings extracted from the fifth-to-last layer are depicted in Figure 4.3, showing language-agnosticism in practice. The Figure also demonstrates how the model recognizes geographical names – a sentence about Portugal is projected close to a sentence about Syria.

In order to statistically measure the results and test our initial hypothesis, we evaluated the performance of mBERT on the parallel sentence matching task described above. The results for embeddings on all 13 layers are depicted in Figure 4.2. The best accuracy is achieved by the fifth-to-last layer, where the model is able to find the correct translations in 90% of cases when matching

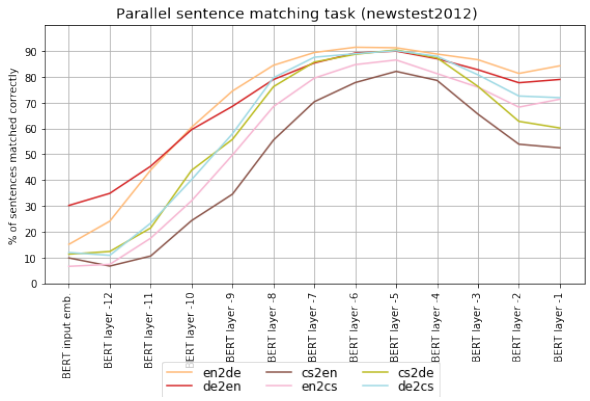

Figure 4.2: Results of the parallel sentence matching task

Czech-German and Germa-English. Retrieving translations in Czech-German seems to be more difficult as the success rate is around 85%.

From our experiments, we conclude that pretrained multilingual language models such as mBERT are able to learn cross-lingual structures which are effective for cross-lingual transfer. These structures are most detectable in the representations from the middle layers of the model. Although the results are already impressive for an unsupervised model, further alignment of the representations would be necessary to serve as powerful language-agnostic sentence embeddings.

As future work, the sentence embeddings could be further aligned and used as fixed length embedding vectors, e.g. to filter parallel sentences out of non-parallel corpora. It would also be interesting to explore how the representation change after fine-tuning the model on a downstream task, e.g. paraphrasing or sentence similarity.

In the following Chapter, we will use a pretrained Czech-German model and fine-tune it for machine translation. In the unsupervised scenario, the cross-lingual transfer capacity of the pretrained model is crucial for jump-starting the training of the model as it is the only source of initial cross-lingual information to the model.

(a)

(b)

Figure 4.3: Neighboring sentences in a cross-lingual space. The space was generated by the fifth-to-last layer of mBERT. The color of the dots indicates the cosine distance (the closest points are red).

Chapter 5

Machine Translation between Czech and German

5.1 Experiment Design

In the theoretical part of this thesis, we provided an overview of MT techniques which are applicable in low-resource conditions. In this Chapter, we apply several of these techniques on translation between German and Czech and compare the results.

Czech-German is not an authentic low-resource language pair; there are parallel data sources available (e.g. movie subtitles, EU legislation) extensive enough to train a standard supervised model. However, simulating the low-resource scenario gives us the opportunity to make a comparison between supervised and unsupervised techniques. As future work, the tested models could be compared in authentic low-resource conditions, for example for translation to or from languages such as Basque or Urdu.

We will compare the following unsupervised systems which were theoretically described in Chapter 5: statistical (USMT), neural with XLM pretraining (XLM+UNMT) and hybrid (USMT+NMT). Two other systems will be used as benchmarks for comparison: supervised NMT model and pivoting NMT model.

5.2 Model Details

5.2.1 Unsupervised Statistical MT (USMT)

We estimate an unsupervised SMT system following the design by (Artetxe et al., 2018b) in the following steps. The theory behind unsupervised SMT was given in Section 2.2.

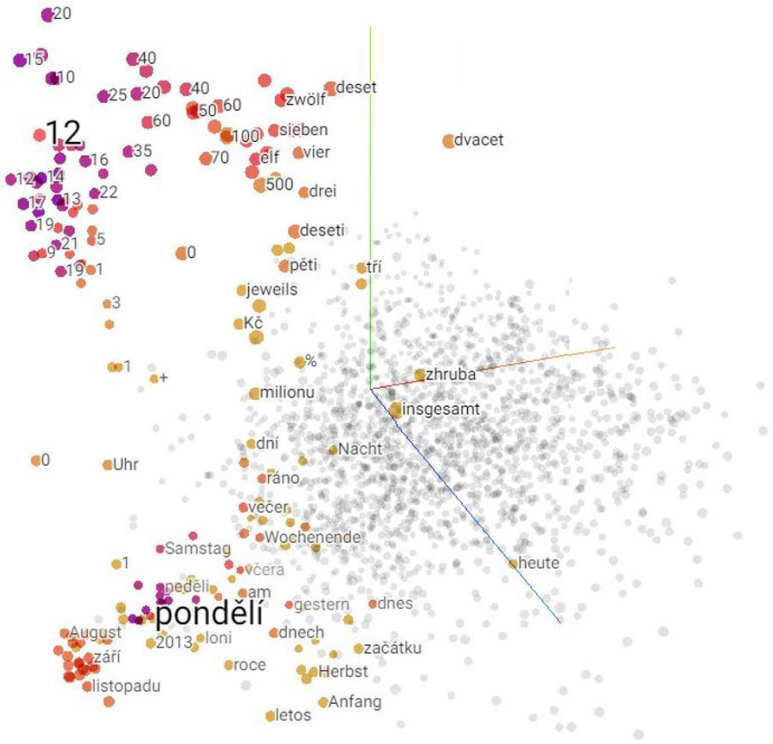

Figure 5.1: PCA visualisation of the aligned Wor2Vec embedding spaces. The highlighted points are the nearest neighbors of the words ”12” and ”pondělí”.

Cross-lingual Embeddings

We apply the methods from Section 3.1. For each language, we extract unique phrases of one, two and three words from the training data and count their occurrences in the training corpus. In order to keep the size of the vocabulary manageable, we restrict it to to the most frequent 200,000 unigrams, 400,000 bigrams and 400,000 trigrams.

We use the Phrase2Vec (Artetxe et al., 2018b) extension of the Word2Vec skip-gram model with negative sampling to train embeddings of the extracted phrases individually in the two languages. The embedding model uses a window size of 5, embedding dimension of 300, 10 negative samples, 5 iterations and no subsampling.

The VecMap (Artetxe et al., 2018a) technique is used to align the two monolingual embedding spaces into a cross-lingual space. A visualization of the cross-lingual space is available in Figures 5.1 and 5.2. In the resulting space, phrases with the same or similar meaning are close to each other, regardless of the language they are expressed in. Figure 5.1 shows how numericals and date-related terms are projected close to each other.

Nearest Neighbors	Cosine Distance
sagte	0.000
erklärte	0.099
řekl	0.122
meinte	0.136
uvedl	0.162
prohlásil	0.162
betonte	0.164
sagt	0.180
dodal	0.197
meint	0.273
říká	0.287
betont	0.288

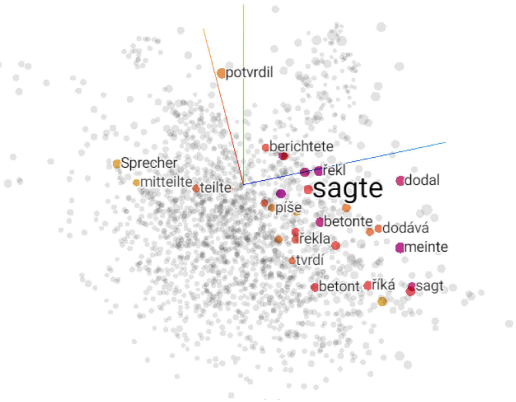

Figure 5.2: Nearest neighbors of the word ”sagte” (which is a German translation of ”said”) and their cosine distance from the original word

Unsupervised Phrase Table

The next step is to populate the phrase table with translation candidate pairs. The phrase table is a dictionary of phrases featuring a translation probability for each phrase. To populate it, we induce a dictionary from the aligned embedding space. For each Czech phrase, we extract 100 nearest neighboring phrases in German from the cross-lingual embedding space and vice versa. When translating from Czech to German, the translation probability of each candidate pair is calculated as follows

$$p(de|cs) = \frac{\exp cossim(cs, de)/\tau}{\sum_{de'} \exp cossim(cs, de')/\tau} \tag{5.1}$$

where cs is the original phrase, de is the selected translation and de' iterates over the 100 possible translations. τ is a constant temperature parameter controlling the confidence of the predictions tuned during the model estimation (Artetxe et al., 2018b).

An illustration of the nearest neighbors of the word ”sagte” (which is a German word for ”said”) is available in Figure 5.2, together with the cosine distances which are used to estimate the translation probabilities of each candidate pair. The nearest neighbors, both in German and in Czech, are all related to the activity of *speaking, claiming, announcing* etc., proving a successful alignment of the two monolingual embedding spaces.

Language Model

We trained a 5-gram language model in both languages using the KenLM toolkit incorporated in Moses. We pruned n-grams of order three (and higher) with only one occurrence to eliminate infrequent phrases. Modified Kneser-

Ney smoothing is used as the default smoothing method of the KenLM toolkit to deal with unseen phrases.

Unsupervised Tuning

We use the Moses implementation of MERT to iteratively tune the weights of the log-linear model on 10k synthetic parallel sentences as described in Section 2.2.

Back-translation

We finally have two SMT models (German → Czech and Czech → German) and we iteratively improve them by three rounds of iterative back-translation, as described in Section 2.2 and depicted in Figure 2.1. In each round, we select 2M sentences to be back-translated. We use Moses to estimate the improved SMT models, supervised by the synthetic corpora generated by back-translation.

5.2.2 Unsupervised Neural MT (XLM+UNMT)

The unsupervised NMT system with language model pretraining was described in Section 2.3. The training pipeline consists of a pretraining phase and a fine-tuning phase.

Model Architecture

In our experiment we build a Transformer NMT model with 6 encoder layers, 1024 hidden units and 8 attention heads. Following Conneau et al. (2018), we use GELU activations and a dropout rate of 0.1.

The general architecture of the model was described in Sections 1.2.3 and 3.2, more details can be found in Lample and Conneau (2019).

Vocabulary

We are using a shared subword vocabulary for both the source and the target language. The BPE segmentation is learned from a concatenation of the two corpora with a target vocabulary size of 60,000.

Pretraining

We train the model on streams of sentences from three monolingual corpora (Czech, German ans English) using the masked language model (MLM) training objective defined in Section 3.2.1. The text streams are 256 tokens long, the model is trained with an Adam optimizer and the learning rate starts at 10^{-4}. We set the batch size to 2400 tokens per batch in order to fit the model on a Quadro P5000 GPU with 17 GB of RAM. However, the model is trained

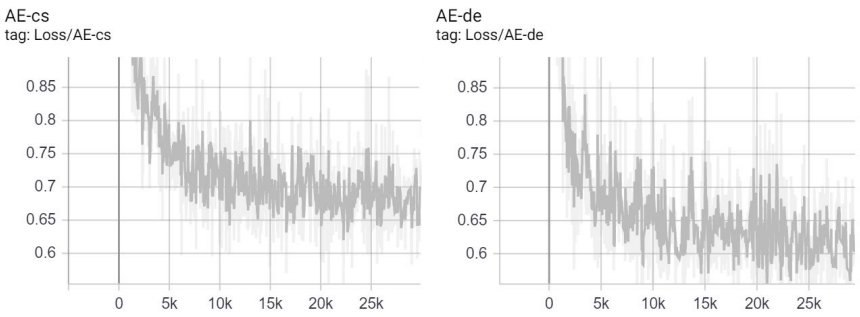

Figure 5.3: De-noising loss during training of the XLM+UNMT system

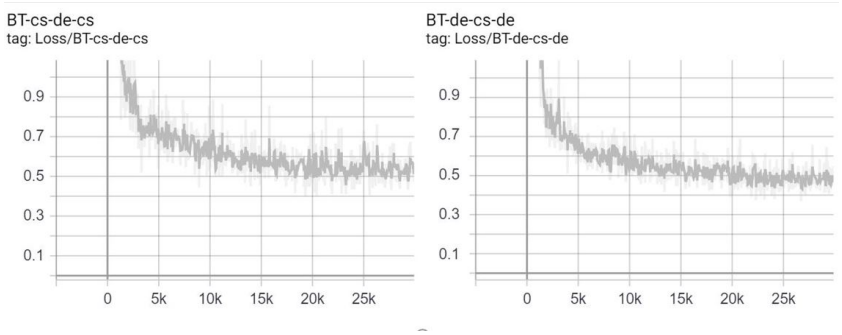

Figure 5.4: Back-translation loss during training of the XLM+UNMT system

on several GPUs so the effective batch size is larger. We trained the model on 8 GPUs in parallel for 50k steps (1 day).

Fine-tuning

The model is fine-tuned with a de-noising loss and a back-translation loss as described in Section 2.3. The noise is added to each de-noising batch by shuffling the input words within a window of length 3, dropping 10% of the words and replacing 10% of the words. We train the model to reconstruct the original sentence from its noised version. In the back-translation step, the model first translates the batch and then tries to reconstruct the original sentences from the automatic translation.

One training step consists of processing one de-noising batch and one back-translation batch. The corrsponding learning curves are shown in Figures 5.3 and 5.4. The model is trained with an Adam optimizer and the learning rate starts at 10^{-4}. The batch size is 2400 tokens. The training ran on 8 Quadro P5000 GPUs in parallel for 3 days (35k steps). We used the same hyperparameters as Lample and Conneau (2019) in their original paper. Experimenting with higher learning rates led to divergence.

The development of the BLEU score on the validation set during training

	Pretraining	Fine-tuning		
Model Name	**MLM**	**De-noising**	**Back-translation**	**Translation**
USMT	-	-	-	-
XLM+UNMT	cs,de	cs,de	cs-de, de-cs	-
USMT+NMT	-	-	-	de*-cs
Pivoting cs-en	cs,de,en	-	cs-en, en-cs	cs-en,en-cs
Pivoting en-de	cs,de,en	-	cs-de, de-cs	cs-en,en-cs
Supervised	cs,de	-	cs-de, de-cs	cs-de

Table 5.1: Overview of trained models and their training objectives. * indicates synthetic text.

is shown in Figure 5.5.

5.2.3 Unsupervised Hybrid MT (USMT+NMT)

We generate a synthetic parallel corpus by using the SMT model from Section 5.2.1 to translate 26M sentences of the Czech monolingual corpus and use it to train a German→Czech NMT model. We do not use a pretrained model to initialize the training but rather train the entire system from scratch, using the standard supervised MT objective described in Section 1.2.4. Since the synthetic corpus only works for translation from German to Czech (the synthetic text must not be on the target side), we only train a unidirectional model and cannot use on-the-fly back-translation. The training ran on 8 Quadro P5000 GPUs for 11 hours (30 k steps).

As future work, it would be interesting to train the hybrid model iteratively in both directions as suggested in Artetxe et al. (2019), switching the synthetic side and possibly improving the translation quality in both directions.

5.2.4 Pivoting Benchmark

Pivoting is applicable in scenarios where parallel data is not available for language pairs of interest but it is available for other language pairs. In our case we can use a German-English and Czech-English parallel corpora to train two models and eventually translate from German to Czech using their combination.

We pretrain an English-German-Czech cross-lingual language model according to the setup from 5.2.2. Since we have twice as many English sentences as Czech or German sentences, we subsample them to one half.

We train two supervised Transformer-based NMT models initialized with a pretrained English-German-Czech model. The translation models are fine-tuned using the back-translation and supervised translation objectives. To generate final translations between German and Czech, we pass each source sentence through both of the models in sequence.

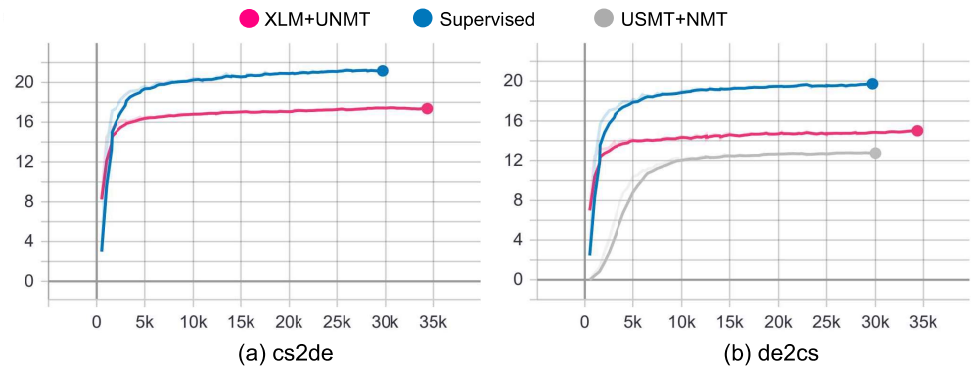

Figure 5.5: Learning curves of the XLM+UNMT, USMT+NMT and Supervised models. The development of BLEU scores on newstest2019.

5.2.5 Supervised Benchmark

We train one neural model on authentic parallel data to have a supervised benchmark for comparison. The fine-tuning details are identical to the Pivoting models from Section 5.2.4.

5.3 Tools

We use the Monoses[13] (Artetxe et al., 2018b) training pipeline implemented in Python to estimate the USMT model. The implementation relies on Moses[14] (Koehn et al., 2007) for the majority of the training steps. Moses is a system for automatic training of translation models and Monoses extends it for unsupervised translation from monolingual data. The KenLM[15] toolkit (Heafield et al., 2013) is integrated into Moses for language modeling. Phrase2Vec and VecMap are integrated in Monoses for learning and mapping of embeddings, respectively.

Neural models are implemented in Python using the PyTorch (Paszke et al., 2017) framework. We use the XLM (Conneau et al., 2018) toolkit for language model pretraining and NMT fine-tuning. TensorBoard[16] is used to visualize the training progress. FastBPE[17] is used to generate the subword vocabulary of the NMT model.

Data preprocessing is handled by standard Moses scripts.

[13] https://github.com/artetxem/monoses
[14] http://www.statmt.org/moses/
[15] https://kheafield.com/code/kenlm/
[16] https://github.com/tensorflow/tensorboard/blob/master/README.md
[17] https://github.com/glample/fastBPE

SOURCE (German)													
Raw text	Mein Name ist Ivana Kvapilíková.												
Tokenized & segmented	[CLS]	Mein	Name	ist	Ivana	Kva##	pilí##	ková	.	[SEP]	[PAD]	[PAD]	
Vocabulary IDs	0	49107	15729	10298	50278	148	10362	30678	119	1	2	2	
Positional indices	0	1	2	3	4	5	6	7	8	9	10	11	
Mask	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	
TARGET (Czech)													
Raw text	Jmenuji se Ivana Kvapilíková.												
Tokenized & segmented	[CLS]	Jmen##	uji	se	Ivana	Kva##	pilí##	ková	.	[SEP]	[PAD]	[PAD]	
Vocabulary IDs	0	147	65361	10775	50278	148	10362	30678	119	1	2	2	
Positional indices	0	1	2	3	4	5	6	7	8	9	10	11	
Mask	Variable (masking words which have not yet been generated)												

Figure 5.6: Text preprocessing for a Transformer model

5.4 Data

Monolingual training data was obtained from NewsCrawl[18] which is a collection of newspaper articles amounting to 300 million sentences in German and 100 million sentences in Czech. We randomly selected 26M sentences from each corpus. We used WMT[19] test sets for validation (newstest2013) and testing (newstest2019).

For training the supervised benchmark model, we used the following Czech-German parallel corpora available at the OPUS[20] website: OpenSubtitles (18M), MultiParaCrawl, Europarl, EUBookshop, DGT (5M), EMEA and JRC. The combined dataset has 26M sentence pairs.

For training the pivoting Czech-English-German model, we extracted 26M sentence pairs from the CzEng 1.6 corpus of Czech-English parallel data and 26M sentence pairs from the Europarl (2M), EUBookshop (10M) and OpenSubtitle (14M) corpora.

Preprocessing for SMT

We tokenized and truecased the data using standard Moses scripts. Sentences with less than 3 or more than 80 tokens were removed. The text was converted to its true case to eliminate capital letters in the beginning of sentences while keeping them when grammatically correct (names, German nouns etc.).

Preprocessing for NMT

When training a Transformer NMT model, we feed it with preprocessed parallel sentences. The preprocessing includes the following steps: normalize punctuation; tokenize; add special tokens (`[CLS]` for sentence beginning, `[SEP]` for sentence ending, `[PAD]` for padding token); apply BPE codes; and convert subwords to vocabulary ids. Before applying the BPE segmentation, the BPE

[18] `http://data.statmt.org/news-crawl/`
[19] `http://www.statmt.org/wmt19/`
[20] `http://opus.nlpl.eu/`

Model Name	BLEU de→cs	BLEU cs→de
USMT	11.72	12.39
XLM+UNMT	15.93	15.79
USMT+NMT	13.71	-
Pivoting	16.50	17.46
Supervised	20.83	21.03

Table 5.2: Translation quality of our models measured by BLUE scores

sequences are learned with FastBPE on the concatenation of the source and the target traning corpora.

There are three kinds of input to the model during training (see Figure 5.6): vocabulary ids, position indices and a mask.

5.5 Results

We measured translation quality of the systems by translating 2k sentences (newstest2019) and measuring the BLEU score. We used the `multi-bleu.perl` script from Moses to calculate the score. Table 5.2 summarizes the results.

Out of the unsupervised models we compared, using a pretrained model and fine-tuning it on a de-noising and back-translation task gives the highest BLEU scores. Our experiments confirm that neural training yields substantial improvements over the SMT system. The statistical model can be used as the initial seed (as in the hybrid USMT+NMT model), but a neural model is necessary in the final stage for optimal performance.

By training both the neural and the hybrid MT system, we were able to compare two different approaches to introducing a cross-lingual signal to an NMT system trained on monolingual data:

1. pretraining a multilingual LM (XLM+UNMT);
2. generating a synthetic parallel corpus from a SMT model (USMT+NMT).

Based on our experiment, we conclude that XLM pretraining is more effective and converges to a higher BLEU score of 15.93. However, it is more computationally demanding, both during pretraining and fine-tuning. It took the hybrid model 11 hours to converge on the synthetic data set whereas the UNMT model required almost 3 days largely because of the expensive back-translation steps.

By comparing the unsupervised systems to our benchmarks, we see that XLM+UNMT does not lag far behind the pivoting approach. However, it must be noted that the pivoting model was trained mostly on out-of-domain data (movie subtitles, EU legislation) which might be detrimental to its performance on a test set composed of newspaper articles. The supervised model is ahead by around 5 BLEU points. The benchmark systems do not directly

Model Name	Sentence
Source	Wie vorhergesagt, schwächt sich der Hurrikan Rosa über den kühleren Gewässern der Nordküste Mexikos ab.
Reference	Podle předpovědi hurikán Rosa slábne, jak se přesouvá nad chladnějšími vodami severního pobřeží Mexika.
USMT	Jak předpověděli, oslabí se hurikán Milada o chladnějších vodách severním pobřeží Mexika rozmyslel.
XLM+UNMT	Jak bylo odhadnuto, hurikán Rosa se schwäbe nad chladnějšími přímořskými přístavy Jižního Mexického zálivu.
USMT+NMT	Jak předpověděla, oslabuje hurikán Muriel nad mořskými vodami pobřeží Mexika.
Pivoting	Jak předpověděl hurikán Rosa slábne nad chladnými vodami na severním pobřeží Mexika.
Supervised	Jak bylo předvídáno, hurikán Rosa se oslabuje nad chladnějšími vodami severního pobřeží Mexika.

Table 5.3: Sample translations of the sentence: *As predicted, the hurricane Rosa is weakening over the cooler waters of the north coast of Mexico.*

compete with the unsupervised systems since they have higher data requirements (parallel Czech-German data for the supervised benchmark and parallel Czech-English and English-German data for the pivoting benchmark) and it was expected that they will perform significantly better. The were estimated in order to show a full picture about Czech-German translation and for comparison.

To give an idea about the translation quality corresponding to the aforementioned BLEU scores, Table 5.3 shows the following sample sentence translated from German to Czech.

> *As predicted, the hurricane Rosa is weakening over the cooler waters of the north coast of Mexico.*

Clearly, the translation produced by the supervised model is the best and is almost flawless. The translation via an English pivot is slightly misleading because of a missing comma. Both models with a USMT component have a problem with translating named entities and translate the hurricane's name as *Milada* or *Muriel*. This is a frequent phenomenon for this type of models because vector representations of names are similar and the model is not able to align them properly. The XLM+UNMT model gets the name right but it mistranslates other words and also adds an extra word *bay* which was not mentioned in the original sentence. Furthermore, it is interesting how the neural model translated only half of the German word *schwächt* and generated a hybrid word *schwäbe* which does not exist in either of the two languages.

It is clearly visible that different models make different kinds of mistakes. Efficiently combining the traits of both neural and statistical unsupervised MT in a more sophisticated hybrid system could lead to further improvements in the future.

Conclusion

This thesis contributes to a recent line of research in machine translation based on monolingual data. We compared several approaches and created three unsupervised MT systems. Our experiments confirmed that it is possible to train an MT system exclusively on monolingual texts.

Based on our results, we conclude that neural training initialized with a pretrained model provides a significant improvement over a pure statistical phrase-based model. Furthermore, unsupervised statistical and neural models capture different features of the language and their efficient combination is desirable.

Since pretraining proves efficient for unsupervised MT, we investigated the sources of cross-lingual transfer in pretrained multilingual models such as mBERT or XLM. Such models are trained in a completely unsupervised way, without having access to any translation resources at training time, and yet are able to recognize phrases of similar meaning expressed in different languages. We conclude that multilingual models are learning semi language-agnostic representations which are hidden in their mid layers. As future work, further alignment of the representations could lead to improvements not only in unsupervised MT but also in other downstream tasks.

Language model pretraining brought a significant improvement to the task of unsupervised MT. While the translation quality of the unsupervised systems still lags behind the supervised systems, the results and the training algorithms are impressive both from the linguistic and the machine learning point of view.

Bibliography

Jay Alammar. The illustrated Transformer. In *jalammar.github.io [online]*. [cited on 11-12-2019]. Available from: `http://jalammar.github.io/illustrated-transformer`.

Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, and Stanley Chen. Bidirectional recurrent neural network language models for automatic speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2015.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the ACL*. Association for Computational Linguistics, Melbourne. 2018a.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on EMNLP*. Association for Computational Linguistics, Brussels. 2018b.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the ACL*. Association for Computational Linguistics, Florence. 2019.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*. 2018c.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv [e-Print archive]*, abs/1607.06450. 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. 2017.

M. Cettolo, M. Federico, N. Bertoldi, R. Cattoni, and B. Chen. A look inside the itc-irst smt system. In *Proceedings of the 10th Machine Translation Summit*. 2005.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on EMNLP (EMNLP)*. Association for Computational Linguistics, Doha. 2014.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*. 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv [e-Print archive]*, abs/1810.04805. 2018.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. PMLR, Sardinia. 2010.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122. 2017.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*. MIT Press. 2016. Available from: `http://www.deeplearningbook.org`.

Google. Embeddings: Translating to a lower-dimensional space. In *developers.google.com [online]*. [cited on 11-12-2019]. Available from: `https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space`.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the ACL*. Association for Computational Linguistics, Florence. 2019.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the ACL*. Sofia. 2013.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv [e-Print archive]*, abs/1606.08415. 2017.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistic*, 5:339–351. 2017.

Kaliyaperumal Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-lingual ability of multilingual BERT: An empirical study. *arXiv [e-Print archive]*, abs/1912.07840. 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*. 2015.

Tom Kocmi and Ondřej Bojar. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, Brussels. 2018.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*. Association for Computational Linguistics, Prague. 2007.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, Vancouver. 2017.

Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between European languages. In *Proceedings on the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, New York City. 2006.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the ACL on Human Language Technology - Volume 1*, NAACL '03. Association for Computational Linguistics, Stroudsburg. 2003.

Ivana Kvapilikova, Dominik Machacek, and Ondrej Bojar. CUNI systems for the unsupervised news translation task in WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence. 2019.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291. 2019.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the 6th International Conference on Learning Representations*. 2018a.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on EMNLP*. 2018b.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in neural machine translation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Montreal. 2018.

Jindřich Libovický, Rudolf Rosa, and Alexander M. Fraser. How language-neutral is multilingual BERT? *arXiv [e-Print archive]*, abs/1911.03310. 2019.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, Brussels. 2018.

Benjamin Marie, Haipeng Sun, Rui Wang, Kehai Chen, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. NICT's unsupervised neural and statistical machine translation systems for the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence. 2019.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv [e-Print archive]*, abs/1301.3781. 2013a.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168. 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc. Curran Associates, Inc. 2013c.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the ACL*. Association for Computational Linguistics, Sapporo. 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the ACL*. Association for Computational Linguistics, Philadelphia. 2002.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Long Beach. 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on EMNLP*. Association for Computational Linguistics, Doha. 2014.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the ACL*. Association for Computational Linguistics, Florence. 2019.

Martin Popel and OndÅŹej Bojar. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110. 2018.

Maja Popovic. Comparing language related issues for NMT and PBMT between German and English. *The Prague Bulletin of Mathematical Linguistics*, 108. 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin. 2016.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958. 2014.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. The LMU munich unsupervised machine translation system for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence. 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 3104–3112. Curran Associates, Inc. Curran Associates, Inc. 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762. 2017.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144. 2016.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on EMNLP*. Association for Computational Linguistics, Austin. 2016.

Appendix A

Acronyms

BERT Bidirectional Encoder Representations from Transformers

LM Language Model

mBERT Multilingual BERT

MLM Masked Language Model

MT Machine Translation

NMT Neural Machine Translation

SGD Stochastic Gradient Descent

SMT Statistical Machine Translation

XLM Cross-lingual language model

Appendix B

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `src` ..... the directory of source codes
  - `experiments` ..... source scripts
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format