Karlsruhe Institute of Technology

Unsupervised Transfer Learning in Multilingual Neural Machine Translation with Cross-Lingual Word Embeddings

Master's Thesis
of

Carlos Mullov

KIT Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
Interactive Systems Lab (ISL)

Referees: Prof. Dr. Alexander Waibel
Prof. Dr.-Ing. Tamim Asfour

Advisor: M.Sc. Quan Pham Ngoc

Duration: May 12th, 2020 — November 11th, 2020

Erklärung

Ich versichere hiermit, dass ich die Arbeit selbstständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Karlsruhe, den 11. November 2020

Carlos Mullov

Zusammenfassung

In dieser Arbeit erweitern wir ein multilinguales NMT System mit universalem Encoder-Decoder um eine neue Sprache. Wir machen uns hierbei die sprachenunabhängige Respresentation des universalen Encoders zu Nutze um das gelernte, um die neue Sprache auf rein monolingualen Daten zu erlernen. Als Teil unseres Ansatzes integrieren wir hierfür sogenannte *cross-lingual word embeddings*, um dem NMT System bei dem Transfer zu helfen. Dazu bilden wir vortrainierte monolingualen Worteinbettungen für sowohl die neue Sprache, als auch alle anderen Sprachen in einen gemeinsamen Raum ab. Unter Ausnutzung dieser Worteinbettungen alleine und ohne jegliche Aussetzung des NMT systems gegenüber der neuen Sprache übersetzen wir in die neue Sprache, in einem Prozess den wir als *blinde Übersetzung* bezeichnen. Mit einem europäisch-romanischen multilingualen Grundsystem erreichen wir bis zu 36.4 BLEU auf Portugiesisch-Englisch, durch blinde Übersetzung nach Portugiesisch. Dieses Übersetzungssystem, welches im Verlaufe des Trainings nicht einem einzigen Satz von auch nur einer slavischen Sprache ausgesetzt wurde, erreicht auf Russisch bis zu 13 BLEU. Zusätzlich erreichen wir, unter Verwendung eines neuartigen Ansatzes für kontinuierliche Ausgabedichten in NMT, bis zu 16.2 BLEU in der blinden Übersetzung *nach* Portugiesisch.
Um den Ansatz zu erforschen, lediglich die Abbildung von der Satzrepresentation des Encoders auf die neue Sprache zu lernen, lehren wir das Übersetzungssystem von Portugiesisch nach Portugiesisch zu übersetzen. Durch einfrieren des Encoders im Lernprozess erreichen wir bei anschließendem Übersetzen nach Portugiesisch bis zu 26 BLEU auf Englisch-Portugiesisch. Durch zusätzliches Rauschen in den portugiesischen Eingabesätzen, welches das Übersetzungsmodell anregt die korrekte Wortreihenfolge zu lernen, erreichen wir zusätzliche 2 BLEU.
Zuletzt betrachten wir als Alternativansatz zum Erlernen der Übersetzung in die neue Sprache die Adaption auf synthetischen Daten, die durch Rückübersetzung generiert wurden. Durch Ausnutzung der guten Resultate bei der blinden Übersetzung von einer neuen Sprache generieren wir synthetische parallele Sätze aus einem monolingualen Korpus. Durch Training auf diesen Daten erreichen wir auf Englisch-Portugiesisch 34.6 BLEU, womit die Ergebnisse nur knapp unter die Werte eines Übersetzungssystems fallen, das auf echten Paralleldaten trainiert wurde.

Abstract

In this work we look into adding a new language to a previously trained multilingual NMT system in an unsupervised fashion. We seek to exploit a well generalized and language independent sentence representation of the multilingual model to easily generalize to a new language. To this end we first explore how well generalized this representation actually is and subsequently we explore the venue of learning the new language in a completely unsupervised fashion. As part of our approach we incorporate pre-trained cross-lingual word embeddings into the multilingual NMT system. In order help our model to translate from and to a yet unseen language, we manually align pre-trained monolingual word embeddings for that new language into the shared cross-lingual embedding space. Using cross-lingual embeddings alone allows us to decode from a yet entirely unseen source language in a process we call *blind decoding*. Blindly decoding from Portuguese using a basesystem containing multiple Romance languages we achieve scores of up to 36.4 BLEU on Portuguese-English. Using this same model which has never seen even a single sentence from any Slavic language we are also able to achieve up to 13 BLEU on Russian-English. Furthermore, applying blind decoding on the target side we are able to achieve up to 16.2 BLEU when decoding *to* an unseen Portuguese. To this end, employing a recently proposed approach, we use a continuous output representation as replacement for the softmax output layer.
In an attempt to train the mapping from our sentence representation to a new target language we use our model as an autoencoder. Training to translate from Portuguese to Portuguese while freezing the encoder we achieve up to 26 BLEU on English-Portuguese. Adding artificial noise to the source-side to let the model learn the correct word order gains us additional 2 BLEU.
Lastly we explore a more practical approach to learning the new language by training on back-translated data. To this end we exploit our model's ability to produce high quality translations on an unseen source-side language to generate the synthetic data. Training on the synthetic data yields us scores of up to 34.6 BLEU, again on English-Portuguese, attaining near parity with a model trained on real bilingual data.

Table of Contents

Chapter 1.

Introduction

Machine translation is a discipline of translating from one natural language to another. Currently the fully neural network based approach, dubbed as NMT, is dominating the research scene. This approach is more robust and scalable to data size compared to the previously common statistical approach. Such NMT systems consist of an encoder, which encodes the input sentence into a latent numerical sentence representation space, a decoder, which generates the target sentence from this latent representation, and finally an attention mechanism, which connects the encoder and the decoder. While under favourable circumstances neural networks bring significant improvements to translation performance, a major problem with them, however, is that they are extremely data hungry. For the best performance NMT systems require data in the scale of millions of sentences. While for the most common language pairs we might indeed have these amounts of parallel data, for most language pairs parallel data is scarce. Considering the quadratic combinatorics for the whole of the natural languages in the world, e.g. translating between n languages results in n^2 language pairs, one might even say that parallel data is extremely scarce to non-existent for most language pairs. As such one of the goals of multilingual NMT is to significantly reduce the amount of parallel data we require for each given individual language pair.

Universal multilingual NMT as described by Johnson et al. (2016) and Ha et al. (2016) employs a universal encoder and decoder, meaning that the encoder parameters are shared across all of the source languages. The idea is for the NMT system to learn a decoupled representation of its source and its target languages, so that it can effectively increase the available training data for each of its individual source and target languages. Ideally the universal encoder would further learn a language independent representation of the source sentence, e.g. learn to represent two semantically identical sentences onto a similar neural sentence representation – even across different languages. Ongoing research in the field of zero-shot translation provides evidence that multilingual NMT models exhibit this property up to a certain extent, and that enforcing the

similarity between sentence representations across different languages in turn also improves the zero-shot capabilities (Pham et al., 2019). Research in the field of cross-lingual transfer learning in NMT further shows that such multilingual systems can rapidly be extended to new languages on very little data (Neubig and Hu, 2018). Neubig and Hu (2018) furthermore show that going as far as training a basesystem on 58 source languages immensely helps the universal encoder in afterwards generalizing to new source languages in a supervised setting. In this work we take this approach one step further and investigate the ability to translate to a new language with no parallel data for this new language. While integrating cross-lingual word embeddings into the model end we provide two contributions in the field of multilingual NMT:

1. To explore the generality of the sentence representation we first apply the universal encoder to an entirely unseen language.

2. Using two distinct methods we adapt the universal decoder to a new language.

As a consequence of not providing any bilingual data for the new language in training we do not give the model the chance to learn the cross-lingual word correspondences. Thus the model is not able to learn a shared embedding space through conventional multilingual training. To alleviate this problem we, as our main contribution, manually provide these word correspondences in the form of cross-lingual word embeddings (Joulin et al., 2018; Conneau et al., 2017). We therefore use monolingual *fastText* embeddings for each of our languages, which we manually align into a common word embedding space. While the encoder could not possibly know the syntax of an unseen language, many syntactical concepts in language – such as grammatical cases, part-of-speech, grammatical genders, or tense – are encoded at word level. Take German for example, nouns have different surface forms for different grammatical cases, which the fastText model should take into account and appropriately encode in its word embeddings. As such we surmise for the word vectors trained on the new language alone to provide enough syntactic level information to perform the language comprehension task to a certain degree.

For testing the applicability of the learned multilingual sentence representation to sentences of an unseen language we thus devise three experiments:

1. first, we decode from sentences from a new language without any additional adaptation except for the cross-lingual word embeddings

2. in an attempt to *only* learn the mapping from the encoder representation space to the new language we adapt the model decoder through exposure to monolingual data in an autoencoding fashion

3. by combining these two methods we train the decoder mapping on the synthetic parallel data generated in the first experiment, in a backtranslating fashion

While using a German-English-Spanish-French-Italian multilingual basesystem – which we train in regular supervised translation – we perform experiments in translating to and from Portuguese and Russian as our new language. By simply decoding from Portuguese we achieve BLEU scores as high as 36.4 for Portuguese-English, while we achieve up to 13 BLEU when

decoding from Russian. Next, by adapting to the new language by simply exposing the model decoder to monolingual data in training, we further achieve up to 28.1 BLEU for English-Portuguese and 8.7 BLEU for English-Russian. Finally, we achieve up to 34.6 BLEU for English-Portuguese and 13.9 BLEU for English-Russian, by adapting on synthetic parallel data, generated via back-translation. Reaching a BLEU score of 28 average for all of the languages, the model adapted on the Portuguese synthetic data almost reaches the average 28.8 BLEU of a supervised baseline model that is adapted on real parallel data.

As another contribution we explore the recent proposal of treating NMT as a regression problem (Kumar and Tsvetkov, 2018). In training this approach replaces the softmax-cross entropy layer with the von Mises-Fisher loss function, giving us memory and computational complexity independent of the vocabulary size. Due to our extremely large multilingual vocabularies as well as the usage of fixed, pre-trained word embeddings this approach is especially well suited to our research. While in our experiments this model mostly exhibits sub-par performance considering raw BLEU scores only, it still provides us good enough reasons to promote the usage of this approach. Performing our first experiment using this model we furthermore achieve up to 16.2 BLEU decoding to Portuguese as a target language.

Chapter 2.

Fundamentals

In this chapter we will give a background on the topics required for reading this work. While we give a brief overview of the basics of neural networks and machine translation, for an in depth understanding of these topics and their history we would like to refer to dedicated literature (Neubig, 2017; Koehn, 2017).

2.1. Machine Translation

Machine translation is a task in natural language processing (NLP), where we translate from one natural language to another. Due to historical influence of translation from French to English, the source language is often denoted as f while the target language is denoted as e. While the history of machine translation stretches back to the early beginnings of the computational era, and many different approaches to machine translation have been attempted, it is the statistical approach to this task that is predominantly used. In the statistical approach we use a statistical model to estimate the probability that a target sentence $e = (e_1, \ldots, e_m)$ is the correct translation given a source sentence $f = (f_1, \ldots, f_l)$

$$\mathbb{P}(e \mid f) = \mathbb{P}(e_1, \ldots, e_m \mid f_1, \ldots, f_l) \tag{2.1}$$

Note that the above notation is a shorthand version for the mathematically correct but rather verbose notation $\mathbb{P}(\mathbf{e} = e \mid \mathbf{f} = f)$, which describes the source and target sentences as observations of a random variable.

2.1.1. Neural Machine Translation

Traditionally, the probability 2.1 had been modelled in the so called statistical machine translation (SMT) approach to machine translation (Brown et al., 1990; Koehn et al., 2003). Using the so

called noisy channel model, it is split into a conditional probability and a probability for the target sentence

$$\max_e \{\mathbb{P}(e \mid f)\} = \max_e \{\mathbb{P}(f \mid e)\mathbb{P}(e)\}$$

which are modelled via a translation model estimating $\mathbb{P}(f \mid e)$ and a language model estimating $\mathbb{P}(e)$. While there are many possibilities to incorporate neural networks into the SMT approach such as modelling $\mathbb{P}(e)$ using a neural language model, recent state-of-the-art translation systems use neural networks in an end-to-end fashion. The so called neural machine translation (NMT) takes the direct approach to the probability 2.1 and models it as

$$\begin{aligned}\mathbb{P}(e \mid f) &= \mathbb{P}(e_1, \ldots, e_m \mid f) \\ &= \mathbb{P}(e_1 \mid f) \cdot \mathbb{P}(e_2 \mid e_1, f) \cdot \ldots \cdot \mathbb{P}(e_m \mid e_1, e_2, \ldots, e_{m-1}, f)\end{aligned}$$

The conditional probabilities for each target word $\mathbb{P}(e_k \mid e_1, \ldots, e_{k-1}, f)$ are modelled by a neural network. In this so called sequence-to-sequence task usually an encoder-decoder neural network architecture is employed. An encoder network *enc* encodes the input sentence f into a latent sentence representation c. Given the so called context c a decoder network dec then – usually in an stateful iterative manner – calculates the target word probabilities, e.g.

$$c = enc(f) \tag{2.2}$$

$$(P_k, h_k) = dec(e_k, h_{k-1}, c) \tag{2.3}$$

$$P_k = \mathbb{P}(e_k \mid e_1, \ldots, e_{k-1}, f) \tag{2.4}$$

Most commonly the final decoder layer is the $softmax$ function, whose output P_k is interpreted as a multinomial probability distribution over a fixed output vocabulary $\mathcal{V} = (w_1, \ldots, w_N)$.
While calculating the conditional probability 2.2 for a known e is the core concept in NMT, this process called rescoring is usually not what interests us the most in machine translation. Usually, our target is to find the most probable translation for a known input sentence f

$$\hat{e} = \operatorname*{argmax}_e \{\mathbb{P}(e \mid f)\} \tag{2.5}$$

The process of searching the most probable sentence e – or usually an approximation thereof – is called *decoding*, the most widely employed method being the *beam search* algorithm.

For more details on NMT and the specific architecture we employ in this work refer to section 2.2. For an outline of how the NMT system is trained refer to section 2.1.2.

2.1.2. Parallel Data and Parameter Estimation

Statistical machine translation systems, including NMT, try to model the probability distribution $\mathbb{P}(e \mid f)$ (see 2.1). As described in subsection 2.2 this modelling can be done in various ways. For all those methods, however, there is a common framework: a parameterized function $g_\vartheta : (F \times E) \mapsto [0,1]$ models the desired distribution, mapping pairs of source and target sentences onto a probability. The free parameter $\vartheta \in \Theta$ defines the concrete model, and needs to be

estimated from the available data. Given a data set $\mathcal{D}$ a parameter $\tilde{\vartheta} \in \Theta$ is estimated using a suitable statistical parameter estimator T. This estimator T is commonly a maximum-likelihood estimator, e.g. for $\tilde{\vartheta} = T(\mathcal{D})$ the probability of observing our dataset is maximized:

$$\tilde{\vartheta} = \underset{\vartheta \in \Theta}{\operatorname{argmax}} \{\mathbb{P}(D \mid \vartheta)\} \tag{2.6}$$

$$= \underset{\vartheta \in \Theta}{\operatorname{argmax}} \left\{ \prod_{(f,e) \in D} g_\vartheta(f,e) \right\} \tag{2.7}$$

$$= \underset{\vartheta \in \Theta}{\operatorname{argmin}} \left\{ -\log \left(\sum_{(f,e) \in D} g_\vartheta(f,e) \right) \right\} \tag{2.8}$$

In the case of NMT this function g_ϑ is defined by the neural network itself while the parameter ϑ are the neural network weights. The typical case of estimating $\tilde{\vartheta}$ is represented by iterative minimization of the *cross-entropy* loss of our data set $\mathcal{D}$. This iterative maximization is performed via stochastic gradient descent

$$\vartheta' = \vartheta - \eta \frac{\partial \left(\text{crossentropy}\left(g_\vartheta(f), e\right)\right)}{\partial \vartheta} \qquad (f,e) \in \mathcal{D} \tag{2.9}$$

In the common case of optimizing via a softmax-cross-entropy output layer with *one-hot* vectors[1] as target labels, this method happens to be equivalent to the minimization of the negative log-likelihood of our data, which in turn equates to the maximizing the likelihood (see equation 2.8). A common alternative for the plain stochastic gradient descent is the *Adam* optimization method (Kingma and Ba, 2014).

Data In NMT we estimate the neural network weights $\tilde{\vartheta}$ on the data set $\mathcal{D}$ in a process called *training* of the neural network. The data set, called parallel text corpus, consists of a series of *parallel sentences*. These parallel sentences are aligned pairs (f,e) of a sentence f in the source language and it's translation e in the target language. These translations are typically human made, sometimes explicitly, and sometimes automatically gathered from various sources of multilingual texts.

The above is a description of the standard case of supervised training of an NMT system. Recently, however, a branch of machine translation called *unsupervised MT* has emerged (Lample et al., 2017; Lample and Conneau, 2019; Artetxe et al., 2017, 2019). Such NMT systems are trained without availability of any parallel data whatsoever using only monolingual data for each of the languages involved. Unsupervised machine translation commonly uses the technique of generating synthetic parallel data with a method called *backtranslation* (Sennrich et al., 2016). Therefore a monolingual text corpus $\mathcal{D}_e$ is translated using a trained NMT system, yielding a set of synthetic monolingual data $\mathcal{D}_f$ in the original target language f. Aligning $\mathcal{D}_e$ and $\mathcal{D}_f$ sentence-wise and reversing the translation directions gives us parallel sentences with clean data in the target language e which we can train on.

[1] a one-hot vector v is a unit vector with only a single entry $v_i \neq 0$

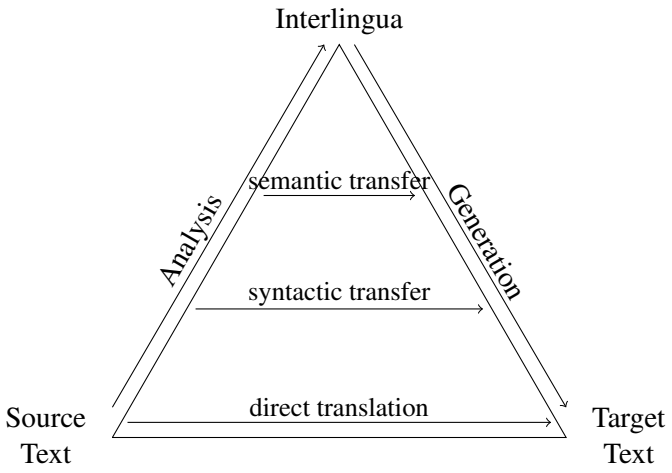

Figure 2.1.: The Vauquois Triangle illustrates the different linguistic approaches to machine translation

2.1.3. Interlingua Translation

As illustrated by the Vauquois Triangle (see figure 2.1) there are several linguistic approaches to machine translation. The most direct one being the direct word-by-word translation of the source sentence. With increasing depth of analysis of the source sentence the translation process produces a more language independent representation of the sentence, and the transfer from the analysis to the generation side becomes less dependent on the specific language pair. The so called *interlingua translation* represents the translation approach with the highest depth of analysis. The sentence is therefore transferred into an *interlingua* representation, which is a completely language independent representation universal across all languages. As a result, semantically identical sentences in different languages would be encoded into the exact same interlingua representation. Translation in such an interlingua-based translation system is then performed by encoding a sentence in the source language into its interlingua representation, and then subsequently decoding from the interlingua representation into a sentence in the target language. Such an interlingua translation system must therefore, besides the interlingua itself, provide two components: for each source language ℓ we require an encoder and a decoder that translate between ℓ and the interlingua representation. The interlingua-translation approach is comparable to *pivot translation*, a technique in multilingual translation where we first translate to a pivot language – usually English – to then translate it to the desired target language. This is useful for learning to translate between low resourced language pairs, since parallel data for the pivot language usually has a much higher availability. The pivot translation approach, however, presents multiple disadvantages. For one the translation error for each translation pass in the chain adds up, especially through the additional ambiguity inherent to all natural languages. Especially in the case of optimization via gradient descent, where the resulting systems are optimized in a non-end-to-end fashion, the error propagation has a large impact. Furthermore

the pivot language may be unable to convey the exact meaning of the source sentence, due to the semantic gap between the pivot language and the source language. A prominent example for this is the distinction between various levels of politeness common to many languages, which is missing in English. The interlingua approach resolves this issue through the interlingua being expressive enough to convey the full semantics of the source sentence. While there exist interlingua representations for very specific closed-domain translation tasks (Mitamura et al., 1991; Levin et al., 1998), finding an open-domain universal interlingua for general purpose translation has oftentimes been deemed impossible. Attempts have also been made at learning interlingua representations automatically, using statistical methods (Kauers et al., 2002). Recent developments in multilingual neural machine translation, however, indicate a strong resemblance to such an interlingua in the latent sentence representation of the neural network (Johnson et al., 2016). Various works also specifically attempt to induce such a neural interlingua, e.g. through cross-lingual regularization methods (Pham et al., 2019; Lu et al., 2018; Escolano et al., 2019). For more information on multilingual NMT and its latent representation refer to section 2.2.4.

2.1.4. Evaluation

As a means to automatically evaluate translation quality, the most commonly used metric is provided by BLEU (bilingual evaluation understudy) (Papineni et al., 2002). In an attempt to strike a balance between modelling evaluation of *fluency* and *adequacy* BLEU measures the amount of 1-gram to 4-gram overlap between the hypothesis and a gold reference sentence. The 1-gram accuracy hereby models the hypothesis adequacy, while accuracies for longer n-grams are responsible for modelling hypothesis fluency. The BLEU score is calculated as the geometric mean of the n-gram relative accuracies

$$\begin{aligned} \text{BLEU-}n &= BP \cdot \frac{\#n\text{-gram overlap}}{\#n\text{-grams}} \\ BLEU &= 100 \cdot \sqrt[4]{\prod_{n=1}^{4} \text{BLEU-}n} \end{aligned} \tag{2.10}$$

BP is the *brevity penalty*, which penalizes short translations, as they tend to get higher scores.

2.2. Neural Machine Translation

2.2.1. Neural Networks

While neural networks have a long history and many different aspects to them, we will provide a very general view on them. Abstractly speaking a neural network g_ϑ is best described as a universal function approximator for a function $g : \mathbb{R}^n \mapsto \mathbb{R}^m$. Oftentimes – especially in the context of NLP tasks – this function g represents a probability mass function. Refer to section 2.1.1 for a description of this probability mass function in NMT and section 2.1.2 for a description of the statistical framework. The neural network g_ϑ itself is a end-to-end differentiable function taking a vector $x \in \mathbb{R}^n$ as its input and producing a vector $y \in \mathbb{R}^m$ as its output. The input x is therefore non-trivially combined with a set of *network parameters* $\vartheta \in \mathbb{R}^l$ which define the

concrete neural network. To approximate the function g the parameters ϑ are then optimized towards a set of values that produce the desired output y for a given input x. Since the function g is usually unknown, or not representable in a closed form, we require a set of example values

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, \qquad y_i = g(x_i) \text{ for } i = 1, \ldots, N$$

for this optimization process. This set of examples $\mathcal{D}$ is usually called the *data set* in the context of supervised learning. This data might, however, also take different forms, such as in the case of unsupervised learning where a technique to generate this data might be employed, or instead we might have $y_i = g'(x_i)$ on a g' related to g. A differentiable function $\mathcal{L}$ called *loss function* gives us a measure of the quality of an approximation y'_i for y_i: $\ell = \mathcal{L}(y'_i, y_i)$. In optimization we now try to find a set of parameters that minimize the cumulative loss on $\mathcal{D}$. This is usually done via a variation of the *gradient descent* optimization method: the negative gradient with respect to our parameters $-\nabla_\vartheta \mathcal{L}(y'_i, y_i)$ points towards a local minimum point of the loss function for the particular training instance (x_i, y_i). Here y'_i is the network output for the input x_i. Moving the parameters one little step towards this minimum thus reduces the loss ℓ of the network output on this particular instance

$$\vartheta' = \vartheta - \eta \cdot \nabla_\vartheta \mathcal{L}(g_\vartheta(x_i), y_i)$$

The factor η determines the size of this step. To minimize the loss for the whole function g instead of just this single point $g(x_i)$ we repeat this minimization process for more of our available points in $\mathcal{D}$ in an iterative fashion. While in general there is no guarantee whatsoever that this pointwise minimization will lead to a successful minimization on the whole of g, neural networks have proven to oftentimes approximate values outside of our known instances in $\mathcal{D}$ remarkably well. This ability to perform well on *unseen* instances $x \notin \mathcal{D}$ is called *generalization ability*.

While, up to this point, we have just described a neural network as an arbitrary differential function g_ϑ that combines its input with ϑ, there are certain patterns in the structure of this function that empowers neural networks. For one, part of the generalization ability of neural networks can be attributed to the *layered* nature of most networks. Most of the time neural networks can be described as a chaining of functions

$$g_\vartheta = g^l_{\vartheta_l}\left(g^{l-1}_{\vartheta_{l-1}}\left(\ldots g^2_{\vartheta_2}\left(g^2_{\vartheta_2}(x)\right)\ldots\right)\right)$$

With increasing *depth* of the network the performance has in recent years oftentimes been observed to increase – thus the rise in popularity of the so called *deep learning*. In practice the networks are mostly made up of common, oftentimes repeating components, combining a subset of the network parameters $\vartheta_i \subset \vartheta$ with an intermediate internal representation. Common examples for these components include matrix multiplication, $1d$ or $2d$ convolution and non-linearities. These components might be combined into larger common components, such as a convolution-layer, a *tanh*-layer, or an attention mechanism. Finally these components are combined in patterns resulting in a certain type of network architecture. Examples include *multi-layer perceptrons*, *long-short term memory networks* (Hochreiter and Schmidhuber, 1997), *time-delay neural networks* (Waibel, 1987), *convolutional neural networks* (Fukushima, 1980;

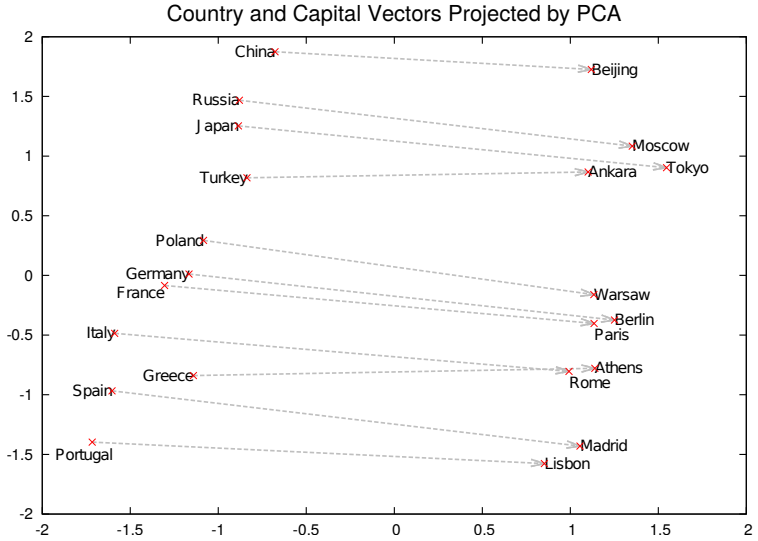

Figure 2.2.: A two-dimensional PCA projection of word embeddings trained via the skipgram method illustrates how a model by itself organizes concepts and implicitly learns the relationships between them (Mikolov et al., 2013b)

Lecun et al., 1989), *encoder-decoder networks* (Cho et al., 2014; Sutskever et al., 2014) or a Transformer network (Vaswani et al., 2017). For more information on the network architecture relevant to NMT refer to 2.2.3. For more information on the layered structure and what the network might learn refer to 2.2.6.

2.2.2. Word Embeddings

Word embeddings are a technique to represent discrete word units w from a vocabulary $\mathcal{V}$ as numerical vectors. These word embeddings are commonly used in NLP tasks when a neural network receives words as its input. By some appropriate means we therefore learn a representation E for each word $w \in \mathcal{V} = (w_1, \ldots, w_N)$ and then feed the embedded word $x = E(w)$ into the neural network during the forward-pass. This commonly involves learning a concrete vector representation for each word w and then arranging those vectors in an *embedding matrix* E. Word vectors can be learned in various ways. Usually in NMT the encoder embedding layer, as well as the decoder embedding layer, are learned in an end-to-end fashion when learning to translate. These embedding layers are therefore treated as input layers of the network, while the embedding layer forward pass is treated as a multiplication of the embedding matrix E with a one-hot encoded input vector. Including this work, various works, however, use *pre-trained* embeddings in their NMT models, where embeddings are learned in a separate step in the training pipeline (Qi et al., 2018). This commonly involves learning word representations by training a neural network – a so called skipgram model – to predict the context of a word w (Mikolov et al., 2013a; Bojanowski et al., 2017). The neural network in this process learns to arrange the words in its representation space in a meaningful manner, e.g. similar words become close to each other

in Euclidean space. In a study on the organization of this representation Mikolov et al. (2013b) have found the embedding space to exhibit additive properties, such as:

$$emb(\text{king}) - emb(\text{man}) \approx emb(\text{queen}) - emb(\text{woman})$$

Similar behaviour can be observed in the arrangement of countries and their capitals, as seen in figure 2.2.

fastText An issue with the original *word2vec* method (Mikolov et al., 2013a) is that the model does not learn to represent *out-of-vocabulary* (OOV) words, e.g. words that it had not seen in training. This is an issue especially with morphologically rich languages where different surface forms of the same word are considered different words. The *fastText* method (Bojanowski et al., 2017) solves this issue by, instead of learning to represent whole words, learning to represent the sub-word pieces the word consists of and combining them into a representation for the whole word. In particular the model learns to represent the character n-grams in a word. The n-gram representations for some bounds $l \leq n \leq r$ are then summed up for the word representation. Providing an example for $3 \leq n \leq 4$ the word `where` is split up into

```
<wh, whe, her, ere, re>
<whe, wher, here, ere>
```

This allows for representation of unseen words and thus eliminates the issue with OOV words.

2.2.3. Attention Encoder-Decoder Networks

Machine translation belongs to the category of the so called sequence-to-sequence tasks. The neural network receives an input sequence of length n and is tasked to produce a sequence of length m. The neural network architecture used for this kind of task is the encoder-decoder architecture. As the name implies the network consists of an encoder and a decoder. The encoder network reads in the input sentence and produces a latent encoder representation. The decoder takes this encoded representation and from it produces the target sentence, usually in a word-for-word iterative manner[2]. Initial proposals of such encoder-decoder networks (Sutskever et al., 2014; Cho et al., 2014) encode the source sentence in a single fixed size vector c. This fixed size summary of the source sentence, however, lacked the ability to encode the while meaning for longer sentences. Bahdanau et al. (2015) thus proposed a *attention mechanism*, which dynamically generates a summary of the relevant parts of the source sentence, a so called context vector, for each target word. A visualization for this attention in an example sentence can be found in figure 2.3. To this end the encoder *enc*, while reading in the source sentence $f = (f_1, \ldots, f_n)$, produces a hidden state h_i for each input word f_i. The decoder recurrently produces the output words e_i in a stateful manner, producing a decoder intermediate state s'_i in the i^{th} decoding step. In each decoding step i the attention mechanism then compares each of the encoder states h_j with the current decoder state s'_i, producing an *attention score* α_j for each of the input positions j. The attention scores are then used as weights to calculate the context vector c_i

[2]There are, however, proposals for non-autoregressive approaches to decoding, such as Ghazvininejad et al. (2019)

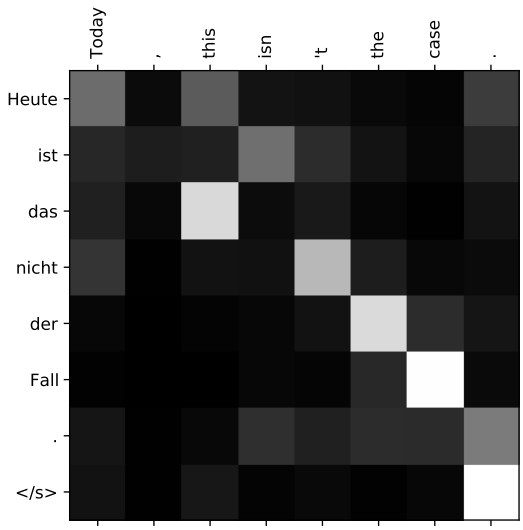

Figure 2.3.: The attention mechanism produces an alignment for each target word, determining the relevance for each of the words in the source sentences. Light values represent high relevance scores α_{ij}.

as a weighted sum of the encoder states:

$$h_j = enc(f_j, h_{j-1}) \tag{2.11}$$

$$(e_i, s_i) = dec(e_{i-1}, s_{i-1}, c_i) \tag{2.12}$$

$$c_i = \sum_{j=1}^{n} \alpha_{ij} \cdot h_j \tag{2.13}$$

$$\alpha_{ij} = normalize\{rel_{i1}, \ldots, rel_{in}\} \qquad rel_{ij} = att(s'_i, h_j) \tag{2.14}$$

The initially proposed system uses a *long short-term memory* recurrent neural network for the encoder, as well as the decoder network each. The encoder hidden states h_j and the decoder hidden states s_i are thus simply the RNN hidden states. The attention model is a simple *tanh*-feed forward layer.

Current state-of-the-art translation systems usually use the encoder-decoder architecture known as the *Transformer* (Vaswani et al., 2017). While previous proposals for encoder-decoder networks all use either RNNs or convolutions for sequence processing, the Transformer relies solely on *self-attention* (Lin et al., 2017). This self-attention produces a new contextual sequence representation from a sequence, by using attention to attend to the sequence itself. For each sequence position i the self-attention attends to each of the sequence positions j, calculating attention scores for all $j \in \{1, \ldots, n\}$. The new representation for position i is then calculated as the weighted sum of all of the input values in the sequence, after calculating the weights from the attention scores. Using multiple layers of self-attention followed by feed-forward layers, the encoder thus encodes

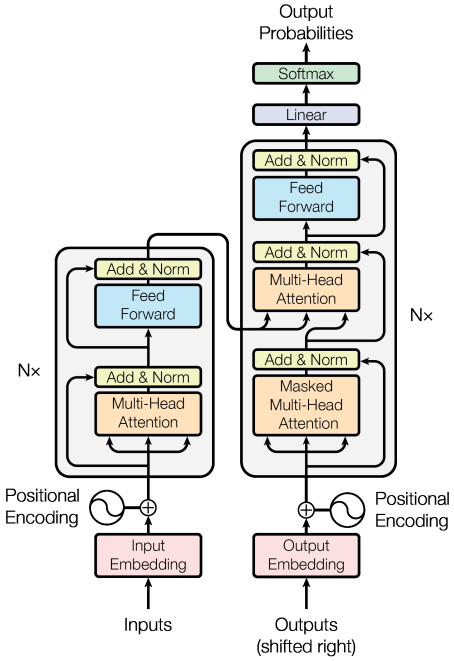

Figure 2.4.: A graphical representation of the Transformer architecture (Vaswani et al., 2017).

the words in the input sentence in a context aware manner. Similarly the decoder word-by-word iteratively produces the output while applying self-attention to the previously generated output words. Additionally the decoder simultaneously also attends to the encoder output sequence using a second attention mechanism. Vaswani et al. (2017) further extend the regular attention mechanism to *multi-head attention*. This extended attention mechanism simultaneously applies k parallel attention layers, while projecting down the input values to each one of the parallel attention layers to different individual representations. This causes each one of these parallel layers, called *attention heads*, to learn to pay attention to different features of the input sequence representation. Figure 2.4 for a graphical representation of the Transformer architecture.

2.2.4. Universal Multilingual NMT

In order to use the available parallel data more effectively research in the field of multilingual NMT tries to combine all the available parallel data to simultaneously train a single unified NMT system on multiple language pairs. The *universal* NMT approach to multilinguality shares all of its network components across all of its source and target languages: it shares the multilingual word embeddings, encoder, decoder and attention mechanism. For any individual source language ℓ this helps the encoder to learn a better sentence encoding for sentences of ℓ, as we effectively have more data available for ℓ as the source language. Similarly it helps train the decoder language model for ℓ if it is the target language, producing better and more fluent target sentences. Ideally the NMT system will learn a common representation for all of its languages, mapping semantically identical source sentences onto the exact same sentence representation. This ideal

case represents a form of interlingua translation (see 2.1.3), where the universal encoder sentence representation presents a form of a neural interlingua. The observed ability of such a multilingual NMT system to do *zero-shot* translation supports the idea that, to a certain degree, the universal NMT system does indeed learn such a universal representation. This zero-shot translation is the ability to translate between a language pair $\ell_1 \rightarrow \ell_2$ for which the translation system has never seen parallel data in training. However, it has also been shown that given enough learning capacity the network will simply partition its parameters, allocating some to certain language pairs (Arivazhagan et al., 2019).

In order to extend a standard bilingual translation system to multilingual translation, the NMT system is simply trained on a multilingual parallel corpus without any modifications to the NMT architecture. This multilingual parallel corpus is in turn simply the concatenation of all of the available bilingual parallel corpora for all of the language pairs involved. The shared vocabulary must therefore be generated from the multilingual corpus data. The NMT system is then subsequently trained to simultaneously translate to multiple languages, e.g. a single mini-batch might contain sentence pairs of different source and target languages.
Additionally a mechanism is required for selecting the desired target language in translation. As originally proposed in Johnson et al. (2016) and Ha et al. (2016) this can be done via a special token in the beginning-of-sentence (BOS) position, where each possible target language has its own token. Alternatively, through a second input channel in the decoder a *language embedding* can also be provided to specify the target language (Ha et al., 2017). Provided a suitable selection mechanism the neural network will, by itself, learn to translate from each of its source languages to each of its target languages, in the process automatically recognizing the input sentence source language without requiring any further input.

2.2.5. Multilingual Word Embeddings

Similarly to the workings of the shared encoder, the universal multilingual NMT system also learns in training to arrange words in its multilingual embedding space in a meaningful way. Ha et al. (2016) show that words similar in meaning, even across language boundaries, are clustered together in such a multilingual embedding space. Figure 2.5 illustrates this concept in a shared embedding space extracted from their multilingual translation model. While this multilingual embedding space is induced in end-to-end training on the translation task, different methods to induce such an embedding space have recently been proposed. Recent work on so called *cross-lingual word embeddings* (CLWE) allow to train an alignment mapping between different monolingual word embedding spaces (Conneau et al., 2017; Joulin et al., 2018; Artetxe et al., 2018a,b). To this end either a bilingual word lexicon can be employed to train the alignment in a supervised fashion, or a fully unsupervised alignment can be learned without any bilingual data whatsoever. For a comparison between the supervised and the fully unsupervised methods see Ruder et al. (2019). In both cases the proposed methods train a linear mapping $A_{\text{src}\rightarrow\text{tgt}}$ from the source embeddings space E_{src} to the target space E_{tgt} through iterative alignment of the embedding matrices. The new E_{src} aligned into the target embedding space is then simply calculated as

$$E'_{\text{src}} = A_{\text{src}\rightarrow\text{tgt}} \cdot E_{\text{src}}$$

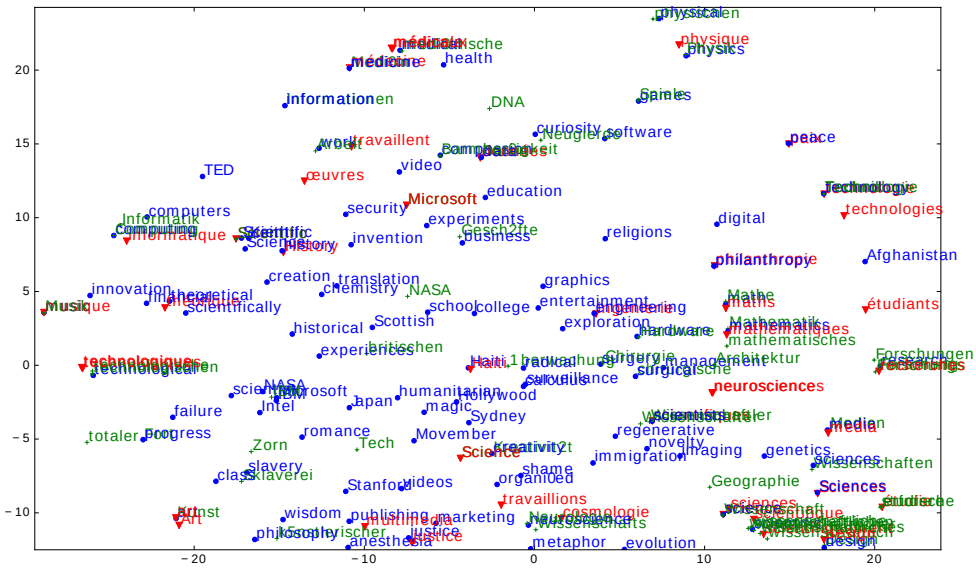

Figure 2.5.: The visualization of the shared embedding space of a universal multilingual NMT system shows the networks tendency to cluster together words which are direct translations of each other (Ha et al., 2018).

In combination with a monolingual word embedding method such as a *fastText* model (see section 2.2.2) it is thus possible to induce a multilingually aligned word embedding space using monolingual data only.

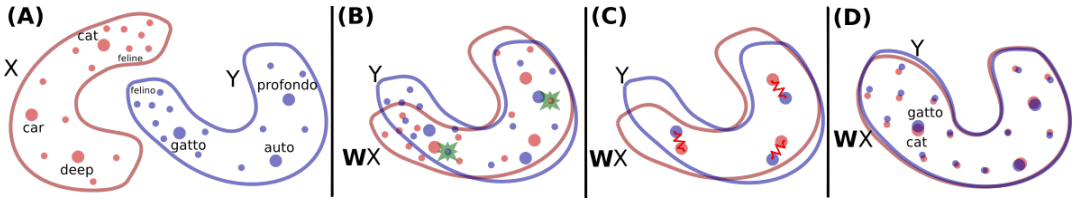

Figure 2.6.: An outline of the unsupervised alignment method by Conneau et al. (2017) shows how the embeddings space X is iteratively stretched and rotated to fit to the embedding space Y.

2.2.6. Transfer Learning

Given two random variables X and Y and a model g_ϑ parameterized by ϑ to estimate the distribution $\mathbb{P}(Y \mid X)$ we will use the data set $\mathcal{D}$ to estimate a fitting ϑ (as described in section 2.1.2). This data set should consist of pairwise observations of (X,Y): $\mathcal{D} = \{(x_1,y_1),\ldots,(x_n,y_n)\}$. It might, however, be the case that the data set $\mathcal{D}$ is of insufficient size to approximate the distribution to an adequate quality. In such case we might find a larger data set $\tilde{\mathcal{D}}$ with observations for variables $\tilde{X}$ and $\tilde{Y}$ which share a similar distribution with X and Y. Given an iterative method of parameter estimation $\tilde{\mathcal{D}}$ can thus be used to estimate an initial set of parameters $\tilde{\vartheta}$ suitable for $\mathbb{P}(\tilde{Y} \mid \tilde{X})$. Continuing the estimation process on $\mathcal{D}$ to find parameters better suited to $\mathbb{P}(Y \mid X)$ is a process we call *fine-tuning*.
This process works particularly well when using deep neural networks for our model g_ϑ. Early attempts at transfer learning with neural networks are described by Waibel (1989); Waibel et al. (1989). Research in visualization of deep convolutional neural networks for image classification tasks shows how lower network layers learn to extract generic features particular to the distribution of X (Zeiler and Fergus, 2013). Lower layers are thus mostly independent of Y and only with increasing depth do the neurons start to react to specific image classes. This property has led to successful transfer learning on image recognition tasks, training a classification network on the large *ImageNet* data set (Deng et al., 2009). The classification model is then adapted to a lower resource classification task, going as far as even freezing lower layers in fine-tuning. Similarly in NLP, training a general model on large data sets and fine-tuning it on domain specific data is a process called *domain adaptation* and has been common practice before even the introduction of neural networks to machine translation (Chu and Wang, 2018). Further, recently successes in transfer learning between different NLP tasks have been achieved, most notably through the application of BERT (Devlin et al., 2018).

2.2.7. Continuous Output NMT

Since taking the first place in the 2012 ILSVRC (Krizhevsky et al., 2012) neural networks have gathered a great deal of attention for achieving superior performance to traditional methods in computer vision and various other tasks. Much of this attention can be attributed to superior performance on classification tasks in particular. The same also applies NMT and many other NLP tasks, which all belong to the category of sequence classification tasks. A most common approach to this classification is to train a neural network with a softmax output layer to minimize the cross-entropy loss between the network output and one-hot encoded class labels. In the case of NMT these classes are represented by the words in our output vocabulary. A major disadvantage to this method, however, is that the large output layers, whose size stands in direct relation to the output vocabulary, results in very high computational complexity as well as memory complexity of the softmax output layer. As such keeping the output vocabulary size at an acceptable level is critical in NMT. To this end various methods such as the usage of sub-word units (Sennrich et al., 2015) or hierarchical softmax have been employed.
The approach proposed by Kumar and Tsvetkov (2018) on the other hand solves the softmax issue by replacing the softmax layer with a continuous embedding layer in the output. Instead of the prevalent treatment of NMT as a classification task they formulate NMT as a regression task. In

training they minimize the distance between the final layer output and the target word embedding. This target word embedding is taken from a pre-trained embedding model, such as *word2vec* or *fastText*. The loss function they put to use in their regression task is the negative log-likelihood of the *von Mises-Fisher* distribution. The von Mises-Fisher distribution hereby plays the role of a probabilistic cosine loss, as they are looking to maximize the directional similarity of the output and target word embeddings. The distribution density, as they use it, is defined as

$$p(\mathbf{e}(w);\hat{\mathbf{e}}) = C_m\left(\|\hat{\mathbf{e}}\|\right) e^{\hat{\mathbf{e}}^T \mathbf{e}(w)} \tag{2.15}$$

whereas $\mathbf{e}(w)$ is the pre-trained word embedding of the target word w, $\hat{\mathbf{e}}$ is the network output of dimension m and C_m is a normalization factor. In inference the output word for decoding step i is chosen as the word whose embedding is most similar to the decoder output, according to the von Mises Fisher loss:

$$\hat{w_i} = \underset{w \in \mathcal{V}}{\operatorname{argmax}} \left\{ p(\mathbf{e}(w);\hat{\mathbf{e}}) \right\} \tag{2.16}$$

This approach tries to directly optimize the network output towards the semantic information encoded by the embeddings and thus allows for a computational complexity and memory complexity independent of the vocabulary size.

Chapter 3.

Related Work

3.1. Pre-Trained Embeddings in NMT

In a study by Qi et al. (2018) the authors look at the effectiveness of using pre-trained embeddings in NMT. Similar to this the work in this thesis, they – among other things – ask whether the alignment of the embedding vectors into a shared embedding space helps in NMT, coming to the conclusion that it is helpful in a multilingual setting. This, to the best of our knowledge, is the only work which combines multilingual NMT with cross-lingual word embeddings. We presume that this is the case due to the difficulty of handling the large multilingual vocabulary without the use of subword units.

3.2. Cross-Lingual Transfer Learning

Neubig and Hu (2018) look into the possibility of rapidly extending a multilingual NMT model by a new language. They consider a low-resource language for their new language and compare between bilingual training, multilingual training alongside a highly resourced similar source language and multilingual training with as many languages as possible. They come to the conclusion that a highly multilingual setting – this in their case is a system with 58 source languages – significantly improves the ability to learn the low-resourced language. Similar to this thesis they also achieve significantly good performance on a yet entirely unseen language. Unlike this thesis they do not use cross-lingual embeddings, but rely on bilingual data to teach their model the cross-lingual word correlations.

To the best of our knowledge Kim et al. (2019a), who also look into cross-lingual transfer learning in NMT, presents the most closely related research to the work in this thesis. They look into swapping out the source language to a yet unseen language, teaching the model to translate from it via unsupervised transfer learning. This concept is analogous to our approach of extending

a model with a new language, the difference being their bilingual basemodel, as opposed to our multilingual basemodel. As with our approach to the transfer learning task, Kim et al. (2019a) also use cross-lingual embeddings in conjunction with denoising autoencoding as one of their core strategies. Unlike this thesis they, however, only consider swapping out the source language and leave the target language to the future work. Furthermore, while our main focus lies on exploring transfer learning in a multilingual universal encoder-decoder NMT system, they do not touch upon the subject of multilingual NMT. Lastly, we provide some differences in the execution of the transfer, e.g. they align their monolingual word embeddings for the new language into the parent model embedding space post-training of the parent model. This strategy is less suited towards a multilingual setting, as in training the NMT model would induce a separate embedding space for each of its languages, thus leaving unclear what embedding space to align the new language embeddings to.

Escolano et al. (2019) devise an approach to multilingual NMT with independent encoders and decoders that allows for zero-shot translation, as well as the addition of new languages. They train separate encoders for each source language, which they then train to map source sentences into a shared sentence representation space. To this end they minimize a joint objective function consisting of the losses for translation in both directions, source sentence autoencoding and cross-lingual representation distance. They then investigate the incremental addition of new languages by adding a new encoder or decoder, which they then proceed to train in mapping to or from the shared representation space. This in principle is similar to what we try to achieve in this work, however, in a unsupervised fashion. Furthermore, it is our hope that the universal encoder, being exposed to many different languages, learns to generalize well across different languages. As such we hope that a universal encoder is much easier to adapt to new languages.

Much like the work in this thesis Siddhant et al. (2020) look into leveraging monolingual data for multilingual NMT. Amongst other things they look into extending the model by a new language through masked denoising autoencoding. They, however, neither employ cross-lingual word embeddings, nor do they perform backtranslation. They suggest that their model presents a promising avenue for jump-starting the backtranslation process. The work in this thesis demonstrates this to indeed be the case.

3.3. Unsupervised NMT

The work we present in this thesis is also closely related to unsupervised NMT (Lample et al., 2017, 2018; Artetxe et al., 2017, 2019). Unsupervised machine translation attempts to train a translation system without parallel data at all. This field of research has recently made significant progress through the emergence of unsupervised cross-lingual word embeddings. The most basic unsupervised machine translation systems perform word-by-word translation using bilingual dictionaries which were induced in an unsupervised fashion from the cross-lingual word embeddings (Conneau et al., 2017). Lample et al. (2017) train a full neural unsupervised machine translation system translating between languages ℓ_1 and ℓ_2. Using the word-by-word translations produced by Conneau et al. (2017) they train an initial system performing some semblance of translation in both directions $\ell_1 \rightarrow \ell_2$ and $\ell_2 \rightarrow \ell_1$. The simultaneous translation in both direction is achieved analogously to universal multilingual NMT, by sharing an encoder and a decoder

between both languages. In an iterative process they then train the NMT model to generate ℓ_1 sentences as well as ℓ_2 sentences through denoising autoencoding, while simultaneously training to maximize the similarity between the encoder sentence representation for sentences from ℓ_1 and ℓ_2 through adversarial training. Through unsupervised means they thus essentially induce a universal encoder-decoder multilingual NMT system with a shared sentence representation for ℓ_1 and ℓ_2. Unlike Lample et al. (2017) we consider in this work a different setting. While they essentially bootstrap a multilingual NMT system from monolingual data, we consider a more realistic scenario where we have enough data from higher resource language pairs to train up a regular multilingual NMT system from bilingual data. This supervised NMT system we then extend by a new language by purely monolingual data. We thus learn the shared sentence representation in regular supervised training resulting in a more effective approach to this problem. This difference in the nature of our setting means that if we learn to translate in the direction $\ell_1 \rightarrow \ell_2$ with ℓ_2 being the new language we add, the NMT system already knows how to translate to and from ℓ_1. Assuming a perfectly language independent sentence representation this means the NMT system only needs to learn to translate to and from ℓ_2 while we can ignore the ℓ_1 side, thus opening up different possible approaches to this task. Inspired by this approach, as part of our work we look into denoising autoencoding and backtranslation adapted to our scenario.

Kim et al. (2019b) look into improving upon unsupervised machine translation, based on word-by-word translation through unsupervised cross-lingual word embeddings. They therefore employ word-by-word translation in combination with a language model, as well as a postprocessing step for local reordering. Similar to the work in this thesis they do away with the costly iterative backtranslation, since, while we do employ backtranslation, we merely perform a single round.

3.4. Language Independent Representation

While this is mainly a work on transfer learning in multilingual NMT, exploring the language independence our sentence representation is also an important part of our motivation. Since the very introduction of multilingual universal encoder-decoder NMT by Johnson et al. (2016) the idea of an universal neural interlingua has been a central topic of research. Several works look into inducing a more language independent representation in an encoder-decoder model. Pham et al. (2019) explore the question of how the language independence of the encoder latent sentence representation affects the ability to perform zero-shot translation. By adding constraints during the training they force the model to learn a more similar representation for sentences of different languages, thereby improving the zero-shot translation quality. Lu et al. (2018) look into inducing an neural interlingua in a multilingual NMT system without universal encoder and decoder. They therefore use language specific encoders and decoders but connect these through an intermediate layer in between the encoder and the decoder. They describe this layer as an explicit neural interlingua and use it to perform zero-shot translation, providing – at that time – the only alternative to universal encoder-decoder systems for zero-shot translation.

3.5. Continuous Output Representation

To the best of our knowledge the continuous output approach presented by Kumar and Tsvetkov (2018) has not yet been used in multilingual NMT, nor has it been used for the purposes of unsupervised NMT.

Chapter 4.

Approach

In training the multilingual NMT system we aim to estimate the probability

$$\mathbb{P}(Y_{\ell_{tgt}} = y \mid X_{\ell_{src}} = x) \tag{4.1}$$

that the target sentence y is a suitable translation of the source sentence x. The universal encoder will ideally learn to map the sentences from the different input distributions of $X_{\ell_{src}}$ – one for each source language ℓ_{src} – onto a single, shared latent distribution H_{enc}. The decoder is then tasked to model the distribution 4.1 from this latent variable: $\mathbb{P}(Y_{\ell_{tgt}} = y \mid H_{\text{enc}} = h)$. In this work we aim to answer two questions

- How well can our universal encoder generalize to an unknown input distribution $X_{\ell_{\text{new}}}$ corresponding to a new language ℓ_{new}?
- How well can we learn to translate to ℓ_{new} by merely adapting the universal decoder to estimate $\mathbb{P}(Y_{\ell_{tgt}} \mid H_{\text{enc}})$ given a set of H_{enc} observations for the target sentences from ℓ_{new}?

The missing piece in this approach are the ℓ_{new} word embeddings: in conventional training of a universal multilingual NMT system the ℓ_{new} word vectors are randomly initialized. The model then learns in training to represent its words in a shared multilingual embedding space, by learning cross-lingual word correlations from the parallel data. In a monolingual data only setting we aim to achieve an equivalent result through cross-lingual word embeddings. We integrate these word embeddings into our translation model, by supplying these in the form of pre-trained embeddings. This yields us a three step approach to our transfer learning task

1. train the cross-lingual word embeddings on monolingual data for each of the involved languages
2. train the multilingual basesystem on parallel data for the set of base languages $\ell_{\text{base}} = \left\{\ell^1_{\text{base}}, \ldots, \ell^m_{\text{base}}\right\}$

3. add a new language ℓ_{new} to the system on monolingual data only

4.1. Cross-Lingual Word Embeddings

In universal multilingual NMT as it is proposed by Ha et al. (2016), the NMT system – in training – learns a shared embedding space and a mechanism to correlate between words of different languages. Given bilingual training data it might thus learn to minimize the distance between words in this embedding space if they are direct translations of each other, or to cluster words in some other meaningful way. However, adding a new language without any bilingual data we do not give the NMT system the chance to do the same with the words from this new language. As a means to manually supply the word-level correlations in our embedding space, we want to explore the approach of using pre-trained cross-lingual word embeddings in the NMT system, instead of training them end-to-end in the translation task. We will use pre-trained monolingual word embeddings learned with *fastText* (Bojanowski et al., 2017) for each one of our languages and manually align them into one common embedding space. For this common alignment we pick one of our base languages as a *pivot*. For each language ℓ we then train a linear mapping $A_{\ell \to pivot}$ from the embedding space of ℓ into the pivot's embedding space. We then calculate the effective embedding matrix E'_ℓ for ℓ as

$$E'_\ell = A_{\ell \to pivot} \cdot E_\ell$$

where E_ℓ is the monolingual fastText embedding matrix for ℓ. The linear mapping $A_{\ell \to pivot}$ is learned by means of iterative alignment between E_ℓ and E_{pivot}, as described by Joulin et al. (2018). We then finally concatenate the resulting embeddings E'_ℓ into one shared embedding matrix E, which we then use to initialize the embedding layers in our NMT system. We therefore use shared the network parameters across our encoder embeddings, decoder input embeddings and decoder output embeddings. When translating to a new language ℓ_{new} we then simply swap out the vocabulary and replace the common embedding matrix with $E'_{\ell_{\text{new}}}$.

As the word embedding matrix takes up a large portion of the network parameters, we expect this to greatly ease up the adaptation process towards the new language. Minimizing the amount of trainable network parameters we further hope to increase the neural networks generalization ability, promoting its ability to learn a universal shared representation.

4.2. Continuous Output Representation

Instead of calculating a multinomial distribution over a closed vocabulary, the continuous output approach does – to describe it in Kumar and Tsvetkov's (2018) words – optimize the network output directly towards the information encoded in the pre-trained word embeddings. Using a word embedding method which operates on a semantic level, as it is the case with fastText, lets us optimize towards a semantic output. Conceptionally this is ideal to our use case, since we want to eventually decode towards unseen words of a new language, but in a shared semantic embedding space. As such in training our basesystem, instead of learning to output the words in

the vocabularies of our base languages, we want the output to approximate the semantics in this shared space itself.
Additionally, in a more practical vein, the continuous output approach helps us deal with the issue of having a massive multilingual vocabulary. It gives us computational complexity, and more importantly memory complexity independent of our vocabulary size. This especially allows us to add more languages to our NMT system which, due to memory constraints would be difficult in the case of a regular softmax model. Alongside a softmax model we will thus also evaluate our experiments on a multilingual continuous output model.

4.3. Adding a New Language

As the domain of our work we will focus on adding a new language using monolingual data only. In order to explore the questions we are asking in regard to our multilingual sentence representation we propose three experiments, in increasing level of supervision

1. *blind* decoding
2. (denoising) autoencoding
3. backtranslation

Figure 4.1 describes the initialization process of the transfer system common to all of the performed experiments. While the first experiment serves to answer the question of how well the multilingual model can generalize to an unseen language, the latter two experiments present two different methods of adapting on ℓ_{new} monolingual data. Given the ℓ_{new} monolingual dataset $\mathcal{D}_{\ell_{\text{new}}} = \{y_1, \ldots, y_n\}$, these two adaption methods represent different ways of obtaining H_{enc} observations h for a given $y \in \mathcal{D}_{\ell_{\text{new}}}$.

4.3.1. Blind Decoding

To provide an indication of how well the universal encoder generalizes to unseen languages we let our multilingual model decode from ℓ_{new} sentences, without any sorts of prior exposure to ℓ_{new}. We call this method blind decoding. We therefore simply swap out the source vocabulary and the corresponding embeddings with the ℓ_{new} cross-lingual embeddings and subsequently decode from ℓ_{new}. It is important that we only have words from ℓ_{new} in our target vocabulary if we try to generate sentences from ℓ_{new}, since our model has not yet learned to correlate between BOS_{new} and ℓ_{new} in its internal target language selection mechanism.
By blindly decoding from ℓ_{new} we hope to answer the question of how well the encoder can encode and represent sentences from an unseen language. We want to see how much information the encoder is able to extract from these sentences and then further how well its sentence representation can represent information in a language independent – ideally purely semantic – manner. While we also want to test the NMT system ability to blindly decode *to* a new language, we consider the target side the less interesting side in this experiment, as the decoder couldn't possibly know language features such as word order for ℓ_{new}. As such the language production ability should be very limited, while the language comprehension ability should only

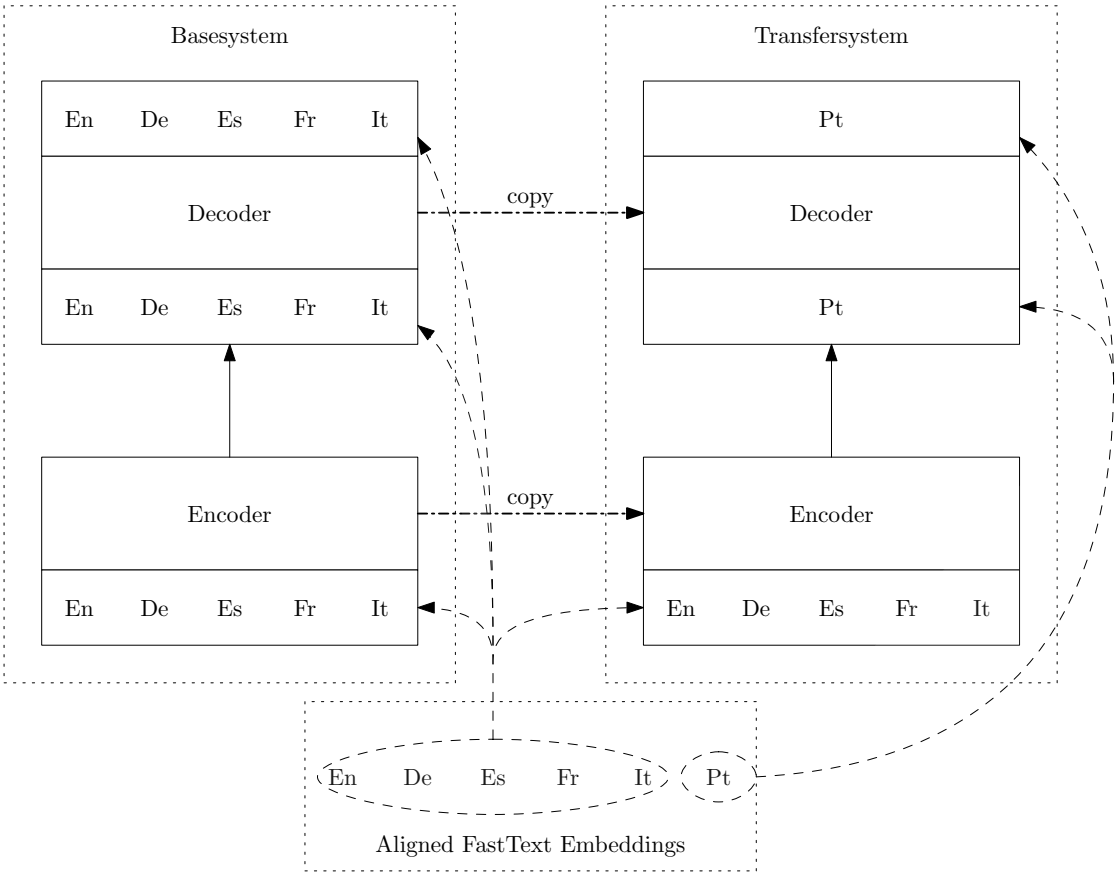

Figure 4.1.: A depiction of the initialization process for the transfer system including ℓ_{new}. It describes how the trained model parameters – that is the encoder and the decoder parameters – are kept as is, while the embedding layers are swapped out. The embedding vectors are provided by monolingually pre-trained fastText models for each language, then aligned into a common embedding space. This example has Portuguese (**Pt**) as ℓ_{new} on the target-side (*decoder*). Depending on the experiment, however, ℓ_{new} might also be on the source-side (*encoder*).
For the experiments on *blind decoding* the transfer system on the right-hand side is used as is to decode to Portuguese, while for the other experiments we further adapt the transfer system to ℓ_{new}.

be constrained to a lesser extend. This method of decoding to and from an unknown language with only the help of cross-lingual word embeddings can be compared to the analogy of a human translator who is given a dictionary for a language he doesn't know, and is asked to comprehend or produce sentences from this language.

4.3.2. Autoencoding

As our second experiment we would like to explore the efficacy of learning a mapping between the sentence representation and ℓ_{new}. To this end we propose two different ways of using our NMT model as an autoencoder, e.g. we train our model to translate from ℓ_{new} to ℓ_{new}:

Denoising autoencoder According to our hypothesis of a well generalized, interlingua-like sentence representation, for our model to learn to translate to ℓ_{new} it suffices to teach the model

1. to correlate between the target language indicating BOS marker BOS_{new} and ℓ_{new}
2. the target language syntax

For the second task we train the model in denoising autoencoding: the model gets noisy input sentences and is tasked with reconstructing the original, grammatically correct version of this sentence. One major aspect of language syntax which sets apart different languages from each other is the word order. By applying slight permutations to the input sentence, one can easily force the model to learn the correct word order. Being exposed to sentences from the new language, we believe the model will implicitly also learn aspects of the syntax other than word order. As described by Lample et al. (2017), we therefore apply *slight permutations* to our input sentences, displacing every word by up to n positions in the original sentence. Additionally we also apply word dropout as an additional noise function: any single word from the input sentence is removed with a probability of p_{wd}.

Plain autoencoder As our second autoencoding method we try simply training $\ell_{\text{new}} \rightarrow \ell_{\text{new}}$ while freezing the Transformer encoder parameters. With this method we rely on the ability of our encoder to encode sentences from ℓ_{new} in a suitable manner. We simply take the encoder output for sentences from ℓ_{new}, which is the latent sentence representation, and learn to generate sentences from ℓ_{new} from it. In order for the model to not just learn to copy words from the input sentence to the output sentence, we rely on the encoder output to be language independent enough to not retain any word level information from the source sentence. This method thus tests the raw ability of our NMT system to do translation in an interlingua-like manner.

Finally we would like to try and combine both methods and see how our results change if we train to denoise while also freezing the encoder.

4.3.3. Backtranslation

Lastly we would like to combine our model's ability to translate from ℓ_{new} and its ability to adapt to ℓ_{new}. We therefore use the in unsupervised NMT commonly employed approach of training

on *backtranslated* data. Therefore one can use a language model in combination with an NMT system trained to translate in both directions to iteratively produce better translations and then tune the bootstrapped model on the improved generated data. In our approach to backtranslation – exploiting the model's ability to perform blind decoding – we choose a simplified approach: we perform one single round of backtranslation in which we use monolingual ℓ_{new} data to

1. blindly decode from ℓ_{new} to each one of our base languages ℓ_{base} in order to generate synthetic parallel data
2. reversing the translation direction we train our model on this synthetic parallel data to translate from ℓ_{base} to ℓ_{new}

We also try freezing the encoder parameters in the process of training. This is in part due to the fact that our main goal is to teach the model decoder to translate to ℓ_{new}, but also in order to give the model less exposure to the noisy synthetic input data.

Chapter 5.

Evaluation

The evaluation for our approach consists of two parts: the evaluation of the multilingual base model, and then finally for adding a new language.

5.1. General Experimental Setup

Cross-lingual word embeddings As the basis for all of our monolingual word embeddings we use the pre-trained *fastText* models provided on the fastText website[1]. These models provide 300 dimensional ℓ_2-normalized word vectors and each contains 2,000,000 entries as full word units accumulated on large web crawled monolingual text corpora. The *MUSE* repository[2] website provides word embeddings for a multitude of languages, which have been pre-aligned into one shared embeddings space. They induce this shared space by picking a pivot language – such as English – and aligning all of the monolingual embeddings to the embeddings for the pivot language, using the supervised alignment technique provided by MUSE. This supervised alignment uses a bilingual seed dictionary to learn a linear mapping between embedding spaces using iterative Procrustes alignment. While in our initial experiments we used these pre-aligned embeddings provided by the MUSE repository, unfortunately the repository provides neither the fastText models for the embeddings, nor the alignment matrices they used. We require these, however, in order to map OOV words into the shared embedding space (as described in 5.1). Using the same single hub pivot approach we thus align our own cross-lingual embeddings from the provided monolingual fastText embeddings. For the alignment we use direct optimization on the RCSLS retrieval criterion (Joulin et al., 2018). Alignment accuracies are listed in table 5.1. In order to reduce the vocabulary size, we subsequently regenerate our vocabulary and the corresponding embedding vectors using only the words in our NMT training corpus.

[1] `https://fasttext.cc/docs/en/crawl-vectors.html`
[2] `https://github.com/facebookresearch/MUSE`

lng	vocab size	% UNKs	CLWE acc
en	47,067	5.9%	1.000
de	98,747	19.8%	0.554
es	74,267	5.8%	0.750
fr	60,416	9.2%	0.732
it	74,962	7.1%	0.696
total	355,459		
pt	67,237		0.734
ru	166,651		0.627

Table 5.1.: The number of words in the vocabulary generated from the training corpus for each of the languages, as well as the percentage of words not contained in the pre-trained fastText models. *total* represents the size of the shared vocabulary of the base model. The column *CLWE acc* describes the nearest neighbour accuracies for the cross-lingual embedding alignments to $\ell_{\text{pivot}} = en$.

Full word-units Using the BPE approach – if learned on a multilingual corpus – gives us the ability to share information between vocabularies, as well as a significant reduction in the size of the resulting multilingual vocabulary. For this work, however, we choose to work with full word units, for the main reason that cross-lingual word embeddings perform poorly when used with subword units (Kim et al., 2019b). We confirm this in our preliminary experiments, where the BPE model in combination with cross-lingual embeddings is vastly outmatched by the full-word unit variant. Furthermore, since we are adding a new language to the model, in order to use BPE we would either have to

- learn the BPE codes on the merged dataset including data for the new language *before* starting the training for the base model. This method is out of scope in the context of our scenario to add a completely unseen language.
- deal with extending the existing BPE codes to include the new language. The fact that relearning the BPE codes to include the new language would change the vocabulary of the base model complicates matters here.
- learn the BPE codes on the new language separately. This, however takes away the above stated advantages of using BPE.

Besides the reduction in vocabulary size the main advantage of the BPE approach is that arbitrary words can be split into tokens present in the BPE vocabulary, therefore eliminating out-of-vocabulary (OOV) words. Our experiments with full word units show that for morphologically rich languages such as German, the pre-trained embeddings from the MUSE repository containing 200,000 words, around 19% of words are OOV words. There is therefore a strong need for a solution to the OOV word problem. We alleviate this by using fastText as the basis for our monolingual word embeddings: by using subword level information to construct embedding vectors for words, fastText gives us the ability to map unseen words into the embedding space. We then subsequently map the word vector into the shared embedding space.

model	nofilter	filter	Δ	nofilter	filter	Δ
softmax	24.50	24.56	+0.06	26.60	26.65	+0.05
frozen	23.31	23.78	+0.47	25.09	25.47	+0.38
continuous	20.11	22.27	+2.16	22.05	24.11	+2.07

Table 5.2.: The comparison between decoding while restricting the target vocabulary to a single language (*filter*) shows an improvement over unrestricted decoding (*nofilter*) for each of our models. While improvements for the regular softmax model are marginal only, both models operating on frozen word embeddings show significant improvements.

The vocabulary of our NMT model consists of all of the single vocabularies and their embedding vectors merged together into one large multilingual vocabulary. Table 5.1 lists the resulting vocabulary sizes, as well as the amount of words in this vocabulary not contained in the pre-trained fastText embeddings. In order to deal with duplicate words across different languages we encode each word with a language specific prefix. Therefore we would encode the English word *bank* as ***en***@*bank*. This also allows for easy filtering of individual languages, e.g. when restricting the target vocabulary to a single language (see chapter 5.1). Because we use full word units, we can easily match any token in the vocabulary to its corresponding language, which is not the case in a model with BPE units that are shared across multiple languages. In inference we thus restrict the target vocabulary to the tokens matching the language we decode to. As the comparison between decoding with and without this restriction in table 5.2 shows this method, besides a speedup in inference, also provides an improvement in translation quality. Using this restriction to a single language, we can also force the model to only choose tokens from a single language in a situation where it would otherwise not do so. We later make use of this to force the model to decode to a yet entirely unseen language (see section 5.3.1).

Datasets As our training, development and evaluation data sets we use multilingual transcriptions of TED talks (Cettolo et al., 2012). For our basesystems we include English, German, Spanish, French and Italian as the base languages. We train the basesystem on a total of 20 language pairs, and a parallel corpus size of $3,251,582$ sentences, which is around $160,000$ sentences on average per language pair. For our experiments on adding a new language we use monolingual TED data for Portuguese and Russian, and parallel test data for the evaluation of the BLEU score. The monolingual data corpora are around the same size as our parallel data, specifically $148,321$ sentences of Portuguese data and $187,843$ sentences of Russian data. For test data and development data we use the IWSLT dev2010 and tst2010 data sets. Development data is in the following tables abbreviated as *dev*.

NMT model For our NMT model we use a multilingual Transformer model with relative position encodings (Shaw et al., 2018). In accordance with our 300-dimensional pre-trained word embeddings for our Transformer model we use a model size of 300, and inner size of 1200, 6 attention heads and 6 layers. Our implementation is based on the repository provided

by Kumar and Tsvetkov (2018)[3], which in turn is based on *OpenNMT-py*[4]. We generally use the same set of languages on source and target side and thus use one shared vocabulary and word embeddings. We further share the same set of pre-trained embedding parameters across the encoder embeddings, decoder input embeddings as well as the output layer of the decoder. For the full set of training parameters, please refer to Appendix A.1.

5.2. Multilingual Base System

For our experiments we consider three different settings for the multilingual systems

1. regular softmax
2. softmax with frozen embeddings
3. continuous output representation

In all of these settings the embeddings – including the output layer embeddings – will be initialized with the pre-trained embeddings.

Regular Softmax The regular non-frozen softmax model mainly serves as a baseline model for comparison with the frozen softmax model, as we do not know how well the monolingually pre-trained fastText embeddings fit the translation task. We initialize this model to the trained cross-lingual embeddings while in training adapting them to the training data. As described by Kim et al. (2019a) we try to extract the trained embeddings from the adapted softmax model, to then align our Portuguese fastText embeddings into the extracted embedding space. Using this approach, however, our methods of adding the new language (as described in section 5.3) have failed to produce any meaningful output when decoding to or from Portuguese.

Frozen Softmax For the *frozen softmax* model we use a regular softmax output layer while freezing the embedding layers in training. This includes the encoder embedding layer, as well as both of the decoder embedding layers. This freezing ensures that the word embeddings stay in the shared embedding space. We can as such trivially add new words to our vocabulary and align new languages into this shared embedding space. Additionally our word vectors stay normalized, which is desirable as it helps generalize towards unseen word vectors since those are normalized as well.

Continuous Output Due to the fact that training the output embeddings in the continuous output approach would lead the embedding vectors to converge towards the trivial 0-vector solution, it limits our ability to train the word embeddings. As we also share our embedding parameters between the encoder and the decoder, the encoder embeddings have to remain fixed as well[5]. We are, however, not limited to completely letting the embeddings vectors remain fixed.

[3] `https://github.com/Sachin19/seq2seq-con`

[4] (Klein et al., 2017, `https://github.com/OpenNMT/OpenNMT-py`)

[5] In our experiments we observe that sharing parameters yields us better results than training the encoder embeddings with decoupled embedding parameters

As a means of adaptation to our task as well as our data we can train the input layer embeddings, while ignoring the gradient of the output layer. Due to the constant covariance shift from the lower layers, this approach has bad convergence properties and only lends itself for very limited amount of adaptation.

Subword-Unit Baseline In order to test the viability of our basesystem – namely a full word-unit translation system with pre-trained word embeddings – we also train a standard subword-unit translation system. This model uses BPE trained on the multilingual corpus as it is commonly used in state-of-the-art translation systems. This reduces the vocabulary size from the $355,459$ words in our full word-unit vocabulary to $36,150$ BPE units.

5.2.1. Results

Since Kumar and Tsvetkov (2018) do not (yet) provide a way to perform beam search for their continuous output approach, we use greedy argmax decoding for the sake of comparability. All of the following results are thus shown for a beam size of 1. The resulting BLEU scores for our three variants, as well as the BPE baseline are shown in table 5.4. Table 5.3 shows a comparison of the BLEU scores averaged over all of the languages. In line with our expectations the continuous output system is outperformed by the softmax models, on average yielding around 2.5 BLEU less. Somewhat surprisingly the softmax model with entirely frozen embeddings performs on par with its non-frozen counterpart, even slightly outperforming it on the development data. This suggests that the cross-lingual embeddings are very well suited for the translation task, despite them being trained on an unrelated task on purely monolingual data.

model	dev	Δ	test	Δ
BPE baseline	26.10		27.73	
softmax	24.56	-1.54	26.65	-1.08
frozen softmax	24.96	-1.14	26.60	-1.13
continuous	22.27	-3.83	24.11	-3.62

Table 5.3.: Comparison of average BLEU scores for the different variants of the multilingual base system

Due to our large multilingual vocabulary sizes of around $500,000$, we are forced to use relatively small batch sizes of around $1,500$ words and a larger amount of batches per update for our softmax models. Since the training complexity and memory consumption are independent of the vocabulary size, the continuous output approach is especially well suited for this multilingual full-word-unit scenario. Training times for the continuous output models thus result in around one fifth of the training times for the softmax models with frozen embeddings, and around one sixth of the training times for the regular softmax models. Furthermore this opens up the possibility to add massive amounts of languages.

lng	de	en	es	fr	it
de		32.6	22.9	24.3	19.6
en	27.2		35.5	34.5	27.8
es	22.8	41.8		31.2	25.9
fr	19.8	33.2	26.2		23.4
it	20.2	31.4	26.6	27.8	

(a) BPE baseline model

lng	de	en	es	fr	it
de		30.6	21.6	22.9	18.7
en	26.3		34.5	33.6	27.0
es	22.1	40.3		30.0	25.0
fr	19.2	32.3	25.3		22.5
it	18.9	30.4	25.4	26.3	

(b) regular CLWE softmax model

lng	de	en	es	fr	it
de		30.6	21.4	23.0	18.5
en	25.9		34.6	33.1	26.5
es	21.8	40.8		29.8	25.1
fr	19.1	32.5	25.3		22.8
it	18.9	30.2	25.7	26.7	

(c) frozen softmax model

lng	de	en	es	fr	it
de		29.1	20.4	18.3	15.2
en	21.1		33.8	30.1	23.4
es	17.5	38.8		27.0	22.3
fr	16.0	31.1	24.4		20.6
it	15.3	29.0	24.3	24.6	

(d) continuous output model

Table 5.4.: The multilingual basesystem test scores for each individual language pair. Every line represents the source language, while columns represent the target language.

5.3. Adding a new Language

As the actual contribution of our work we will next extend our previously trained multilingual base system by a new language. This new language will in the following again be denoted as ℓ_{new}. As the first step after training the multilingual basesystem we

1. from our ℓ_{new} monolingual training corpus D_{new} create the ℓ_{new} vocabulary $\mathcal{V}_{\text{new}}$
2. from $\mathcal{V}_{\text{new}}$ create the ℓ_{new} embeddings E_{new} using our ℓ_{new} fastText model
3. using our bilingual embedding alignment method train the mapping into the shared embedding space by aligning E_{new} to our pivot embeddings E_{pivot}

The following experiments are all conducted using $\mathcal{V}_{\text{new}}$ and E_{new} on the source or target language side, as required in training and decoding.

5.3.1. Blind Decoding

As our first experiment we to decode to ℓ_{new} without any sort of additional training or adaptation. We therefore simply swap out the source or target side embeddings and decode from or to ℓ_{new}.

Target Side With our method of using the *BOS* token to indicate the desired target language, the model learns in training to associate each of its target languages with the matching BOS token. At this point, however, the model has never seen even a single sentence from ℓ_{new} and as such doesn't know to associate its matching token BOS_{new} with ℓ_{new}. Simply decoding using BOS_{new} thus results in a sentence closely resembling in meaning the input sentence, but intermixed with the vocabulary from all of its target languages

```
Ich möchte commencer <unk> by asking du pensar back when you were
bambini, playing con cerramientos.
```

To force the system to decode to ℓ_{new} we thus remove any words not from ℓ_{new} from the target vocabulary.

lng	continuous	frozen
de→pt	8.4	4.8
en→pt	16.2	7.0
es→pt	13.3	5.8
fr→pt	11.6	5.6
it→pt	10.7	5.1
average	12.0	5.7

Table 5.5.: BLEU scores for decoding to Portuguese without any additional training

Using our previously trained basesystems to decode to Portuguese results in an average BLEU score of 6.2 for the continuous output system, and 3.0 for the frozen softmax model. We suspect the main cause for the inferior performance of the softmax model to be the missing bias for

the new language, which the continuous output system does not require. Both models often get stuck in a decoding loop, always outputting the same word until the maximum sentence length is reached. We therefore add a postprocessing step, removing these duplicate words. For both models this almost doubles the BLEU score, while achieving up to even 16.2 BLEU on English-Portuguese. The resulting scores including this postprocessing step are displayed in table 5.5.

Source Side Decoding with ℓ_{new} as a source language doesn't require any additional effort. We simply expand the vocabulary with the aligned word vectors and decode sentences from ℓ_{new}. As seen in table 5.6 we can achieve BLEU scores of up to 36.4 on Portuguese. While the average of 28.3 BLEU on the test set is significantly lower, it is still a considerably high score, considering the model has never seen even a single Portuguese sentence. On a more distant source language such as Russian, we achieve much lower scores. Despite of the fact that the basesystem has never even seen a single sentence from any Slavic language the results are, however, still intelligible, achieving scores of up to 13.1 BLEU. Appendix section A.2 lists some example sentences for Portuguese-English, as well as Russian-English.

lng	de	en	es	fr	it	avg
pt (dev)	19.1	35.7	27.6	25.1	22.6	26.0
pt (test)	20.6	36.4	30.7	29.3	24.4	28.3
ru (dev)	10.0	13.1				11.5
ru (test)	11.1	12.8				11.9
lng	de	en	es	fr	it	avg
pt (dev)	15.6	33.6	26.1	22.2	20.1	23.5
pt (test)	16.2	34.3	28.5	26.4	22.0	25.5
ru (dev)	8.0	12.7				10.3
ru (test)	8.7	11.7				10.2

Table 5.6.: Scores for decoding from Portuguese and Russian as the new language with either the frozen softmax model (top) the continuous output model (bottom)

5.3.2. (Denoising) Autoencoder

As our next step we evaluate our two methods of teaching the model decoder to translate to ℓ_{new}, as well as their combination. Table 5.8 shows the results for decoding to Portuguese and to Russian using these methods. All of the employed methods result in substantial gains over the *blind* decoding strategy. The method of teaching the ℓ_{new} syntax to the decoder via denoising autoencoding (*denoise*) results in slightly lower scores than the raw training of the decoder mapping from the encoder sentence space to the ℓ_{new} sentences (*frozen encoder*). The difference, however, is essentially negligible. When training to denoise via word order shuffling, we found – in accordance with Lample et al. (2017) – that displacing every word by up to $n = 3$ positions yields the bests results. Additionally we experiment with adding additional noise via word dropout in the source sentence. While a dropout rate of $p_{wd} = 0.1$ yields us the best results, we were not

able to observe any noticeable gains over pure reordering noise. However, while performance on Portuguese target data is very impressive, we find the Russian scores to be disappointingly low. Using both methods in conjunction, e.g. freezing the encoder parameters while training to denoise gives us gains of around 2 BLEU points for Portuguese and around 1.5 BLEU for Russian. This brings the translation Portuguese up to an average of 22.9 BLEU, and up to a maximum of 28.2 BLEU on English-Portuguese. Together with our postprocessing method of removing duplicate words from the output data (see section 5.3.1) we are also able to bring the Russian translation scores up to an average of 8.4 BLEU. Table 5.7 shows the resulting BLEU scores for the individual language pairs using these combined methods. Language pairs for which we did not have test data readily available are left blank. The Portuguese and Russian continuous output models trained for 1,000 and 1,250 iterations respectively – taking roughly 15 minutes on a *GTX 1080 Ti* GPU. Note that we now decode using beam search for the frozen softmax model.

	frozen		continuous	
lng	dev	test	dev	test
de→pt	16.8	17.0	16.7	16.9
en→pt	28.2	28.1	27.4	28.2
es→pt	24.5	27.1	23.6	26.4
fr→pt	19.3	21.5	18.4	21.7
it→pt	20.3	20.8	19.5	20.3
average	21.8	22.9	21.1	22.7

	frozen		continuous	
lng	dev	test	dev	test
de→ru	7.5	8.1	6.8	8.1
en→ru	8.6	8.7	7.1	8.0
average	8.0	8.4	7.0	8.1

Table 5.7.: BLEU scores for decoding to new languages after training in denoising autoencoding with a frozen encoder and postprocessed output data

method	$*\rightarrow pt$	$*\rightarrow ru$
denoising autoencoder	20.8	6.5
frozen encoder	21.1	6.6
+ denoise	22.9	7.9
+ postprocessing	22.9	8.4

Table 5.8.: A comparison of average test scores for different variations in training and decoding of the denoising autoencoder method. The comparison shows that freezing the encoder in training and removing duplicate words from the output results in substantial gains in translation quality. While only scores for the frozen softmax model are listed, the scores for the continuous output model behave in very much the same way.

We believe the significant improvement for the frozen encoder approach when adding noise to be an indication that the encoder representation still retains some leftover language specific word-level information. In this case the noiseless model might find it easier to reconstruct the original sentence, thus learning less about the new language syntax in the process. While this is an undesirable quality, this is to be expected considering just the fact that the encoder output sequence is equal in length to the input sentence.

5.3.3. Backtranslation

Without any additional steps, decoding blindly our basesystem already produces very impressive results. As a reminder: English-Portuguese translations go up to 36.4 BLEU (refer to section 5.3.1). In our final experiment we make use of this for backtranslation. Table 5.9 shows the translation scores after training on backtranslated data for each of the language pairs. For each of the variants the best performing model is listed, that is in case of the softmax variant the model with frozen encoder, while the continuous output model performs better with trainable encoder parameters. A detailed comparison can be found in table 5.10. Translation scores reach up to 34.6 BLEU, namely on the English to Portuguese language pair. From the comparison in table 5.10 we can also see that the Portuguese backtranslation scores are almost on par with our supervised model trained on bilingual data, falling short by just 0.8 BLEU. Appendix section A.2 lists some example sentences for English-Portuguese, as well as English-Russian.

	frozen		continuous	
lng	dev	test	dev	test
de→pt	21.1	21.3	19.7	20.1
en→pt	34.1	34.6	32.6	33.1
es→pt	28.7	32.3	27.0	30.9
fr→pt	22.7	26.4	21.3	24.9
it→pt	24.6	25.4	22.7	24.2
average	26.3	28.0	24.6	26.7

	frozen		continuous	
lng	dev	test	dev	test
de→ru	12.4	13.6	10.6	11.5
en→ru	15.1	13.9	12.6	12.0
average	13.7	13.8	11.6	11.8

Table 5.9.: BLEU scores for decoding to the new languages after training on backtranslated data

	frozen		continuous	
method	$*\rightarrow pt$	$*\rightarrow ru$	$*\rightarrow pt$	$*\rightarrow ru$
backtranslation	27.4	13.2	26.7	11.8
+ frozen encoder	28.0	13.8	26.7	11.4
supervised baseline	28.8	16.2	27.5	13.7

Table 5.10.: A comparison of average test scores for decoding to Portuguese and Russian. The models are *a)* trained on backtranslated data without frozen encoder, *b)* with frozen encoder, and *c)* trained on bilingual data in a supervised fashion.

We note that training the Russian system on *en→ru* data only also improves the performance on *de→ru*, in our experiments sometimes even outperforming the performance on the language pair it is trained on. We see this as an indication that this training mainly affects the decoder language model for ℓ_{new}. This furthermore means that the decoder, when decoding to Russian, is able to use a very similar amount of information from the encoder representation of the source sentence. This either means that the encoder representation of English and German are very similar in quality, being able to extract the same amount of meaning. Alternatively it means that the decoder does not learn to use the full amount of information encoded into the representation

of the English source sentence, meaning the decoder is lacking in translation adequacy while leaning towards better fluency.

5.3.4. Summary

In our experiments to add a new language we have employed four levels of supervisedness: in blind decoding we employ no additional data at all, in autoencoding we strictly employ monolingual data, while in backtranslation it is synthetic bilingual data, and finally real bilingual data for the baseline model. Table 5.11 shows a summarising comparison of all of the methods. As expected decoding with ℓ_{new} on the source-side is the easiest. Here even blind decoding comes close to the baseline model for Portuguese achieving up to 36.4 BLEU. We believe this result to provide a resounding yes as an answer to the question of whether the universal encoder learns a well generalized language representation. Applied to Russian the method also delivers impressive results considering the setting. Translation quality is expected to be worse on Russian considering it is linguistically far more distant to the base languages than Portuguese. Additionally its morphological richness and more than double vocabulary size further makes translation more difficult. We, however, believe that the lower quality of the embedding alignment plays a big role in the poorer performance on Russian, thus leaving room for improvement.

In our experiments to adapt the universal decoder to a new target language we compare between our autoencoding approach and the backtranslation approach. Our results for the autoencoding approach also clearly show that translation from other languages can – to some degree – be learned just by learning to translate from Portuguese to Portuguese. While this works well as a proof-of-concept, the backtranslation approach, however, far outmatches the autoencoding approach. This suggests that, while the encoder can blindly extract impressive amounts of information from Portuguese source sentences, the resulting encoder representation is still far worse in quality than for sentences of known languages, e.g. English. This is the case even more so for Russian.

source-side method	pt-de	pt-en	pt-es	pt-fr	pt-it	ru-de	ru-en	∅ pt	∅ ru
blind decoding	20.6	36.4	30.7	29.3	24.4	11.1	12.8	28.3	11.9
supervised	22.2	39.7	33.0	31.9	26.6	15.2	20.9	30.7	18.1
target-side method	de-pt	en-pt	es-pt	fr-pt	it-pt	de-ru	en-ru	∅ pt	∅ ru
blind decoding	8.4	16.2	13.3	11.6	10.7	1.1	1.7	12.0	1.4
autoencoder	17.0	28.1	27.1	21.5	20.8	8.1	8.7	22.9	8.4
backtranslation	21.3	34.6	32.3	26.4	25.4	13.6	13.9	28.0	13.8
supervised	21.9	35.8	32.9	27.1	26.2	15.1	17.2	28.8	16.2

Table 5.11.: A summary of our evaluation scores for translating from or to either Portuguese or Russian. The top half lists evaluation with the new language on the encoder side and the decoder side in the bottom half. Aside from the target-side blind decoding method only softmax model scores are taken into consideration. *target-side blind decoding* lists the results for the continuous output model.

The continuous output model is consistently outperformed by the softmax model by around 1 to 2 BLEU. The exception to this presents the target-side blind decoding, where due to the absence of a word level bias in the output layer the softmax model has difficulties decoding to new languages. The continuous output, however, has shown up to 6 times quicker training times due to larger batch sizes and quicker forward and backward passes.

Chapter 6.

Conclusion and Future Work

6.1. Conclusion

In this work we have looked into adding a new language to a previously trained multilingual NMT system in an unsupervised fashion. We explore the possibility of reusing the existing latent sentence representation, adding a language by merely learning the mapping between this latent representation and the target language space. We hope to see the model learning a generalized and language independent enough sentence representation that we can easily apply and adapt it to an unseen language. As part of our approach we explore the possibility to do multilingual NMT with pre-trained cross-lingual word embeddings. To then help our model map sentences from a yet unseen language to and from the model sentence representation space, we manually align pre-trained monolingual word embeddings for our new language into the shared cross-lingual embedding space. Using this technique alone allows us to decode from a yet entirely unseen source language in a process we call *blind decoding*. By blindly decoding from Portuguese using a basesystem containing multiple Romance languages we achieve scores of up to 36.4 BLEU for Portuguese-English. We believe that this result clearly demonstrates the feasibility of applying the learned sentence encoding to an unseen language and shows that it is indeed language independent enough to generalize to an entirely unseen language. While this works significantly worse on a more distant language, by using this model – which has never seen even a single sentence from any Slavic language – we are still able to achieve up to 13 BLEU for Russian-English. Furthermore, by applying this blind decoding technique on the target side we have been able to achieve up to 16.2 BLEU when decoding to an unseen Portuguese. To this end, employing a recently proposed approach, we have used a continuous output representation as replacement for the softmax output layer. While we found this approach to mostly underperform the traditional softmax approach, we still believe this approach to be promising.

In an attempt to train the mapping from our sentence representation to a new target language we use our model as an autoencoder. By training to translate from Portuguese to Portuguese while freezing the encoder we have achieved up to 26 BLEU for English-Portuguese. The addition artificial noise to the source-side to let the model learn the correct word order gains us an additional 2 BLEU for a total of 28.1 BLEU. When training to translate to Russian we achieve up to 8.7 BLEU for English-Russian.

Lastly we have explored a more practical approach of learning the new language by training the system on backtranslated data. To this end we exploit our model's ability to produce high quality translations from an unseen source-side language to generate the synthetic data. The training on the synthetic data has yielded the scores of up to 34.6 BLEU, again on English-Portuguese, attaining near parity with a model trained on real bilingual data. Translating to Russian yields at most 13.9 BLEU for English-Russian. Considering the low English-Russian baseline score of 17.2 BLEU we suspect the overall low Russian scores to partly be an issue with the low quality Russian word embedding alignment.

6.2. Future Work

In the future we would like to explore the question of how the composition and number of languages in the base system affects the ability to perform transfer learning on a new language. Since transfer learning is largely related to generalization ability, we would like to know whether seeing a wide variety of different languages will help the translation system with new languages – especially more distant ones.

In order to further improve the already impressive performance on source-side unseen languages we would like to explore various methods for the adaptation of the encoder. While the focus of this work lies mainly on adapting the decoder to a new language, the improvement of the source-side would especially benefit the backtranslation results for more distant languages. A possible approach to this would be iterative refinement of the encoder under the utilization of a language model. Alternatively we would like to try and employ the generating NMT model itself in its rescoring capacity to iteratively select the best sentences from one round of backtranslation and subsequently train the NMT system on these.

As an alternative to teaching the new language to the decoder via autoencoding, we would like to explore the possibility of using generative adversarial training on top of a translation system with a continuous output representation.

Bibliography

N. Arivazhagan, A. Bapna, O. Firat, R. Aharoni, M. Johnson, and W. Macherey. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091, 2019. URL http://arxiv.org/abs/1903.07091. 15

M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017. URL http://arxiv.org/abs/1710.11041. 7, 20

M. Artetxe, G. Labaka, and E. Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019, 2018a. 15

M. Artetxe, G. Labaka, and E. Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, 2018b. 15

M. Artetxe, G. Labaka, and E. Agirre. An effective approach to unsupervised machine translation. *CoRR*, abs/1902.01313, 2019. URL http://arxiv.org/abs/1902.01313. 7, 20

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473. 12

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X. 11, 12, 24

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL https://www.aclweb.org/anthology/J90-2002. 5

M. Cettolo, C. Girardi, and M. Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May 2012. 31

K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078. 11, 12

C. Chu and R. Wang. A survey of domain adaptation for neural machine translation. *CoRR*, abs/1806.00258, 2018. URL http://arxiv.org/abs/1806.00258. 17

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017. 2, 15, 16, 20

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 17

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805. 17

C. Escolano, M. R. Costa-jussà, and J. A. R. Fonollosa. Towards interlingua neural machine translation. *CoRR*, abs/1905.06831, 2019. URL http://arxiv.org/abs/1905.06831. 9, 20

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980. 10

M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer. Constant-time machine translation with conditional masked language models. *CoRR*, abs/1904.09324, 2019. URL http://arxiv.org/abs/1904.09324. 12

T. Ha, J. Niehues, and A. H. Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798, 2016. URL http://arxiv.org/abs/1611.04798. 1, 15, 24

T. Ha, J. Niehues, and A. H. Waibel. Effective strategies in zero-shot neural machine translation. *CoRR*, abs/1711.07893, 2017. URL http://arxiv.org/abs/1711.07893. 15

T.-L. Ha, J. Niehues, M. Sperber, N. Q. Pham, and A. Waibel. KIT-multi: A translation-oriented multilingual embedding corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1616. 16

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735. 10

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558, 2016. URL http://arxiv.org/abs/1611.04558. 1, 9, 15, 21

A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, and E. Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. 2, 15, 24, 29

M. Kauers, S. Vogel, C. Fügen, and A. Waibel. Interlingua based statistical machine translation. In J. H. L. Hansen and B. L. Pellom, editors, *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*. ISCA, 2002. URL http://www.isca-speech.org/archive/icslp_2002/i02_1909.html. 9

Y. Kim, Y. Gao, and H. Ney. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. *CoRR*, abs/1905.05475, 2019a. URL http://arxiv.org/abs/1905.05475. 19, 20, 32

Y. Kim, J. Geng, and H. Ney. Improving unsupervised word-by-word translation with language model and denoising autoencoder. *CoRR*, abs/1901.01590, 2019b. URL http://arxiv.org/abs/1901.01590. 21, 30

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980. 7

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017. doi: 10.18653/v1/P17-4012. URL https://doi.org/10.18653/v1/P17-4012. 32

P. Koehn. Neural machine translation. *CoRR*, abs/1709.07809, 2017. URL http://arxiv.org/abs/1709.07809. 5

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://www.aclweb.org/anthology/N03-1017. 5

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. 17

S. Kumar and Y. Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. *CoRR*, abs/1812.04616, 2018. URL http://arxiv.org/abs/1812.04616. 3, 17, 22, 24, 32, 33

G. Lample and A. Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291. 7

G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017. 7, 20, 21, 27, 36

G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755, 2018. URL http://arxiv.org/abs/1804.07755. 20

Y. Lecun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. ISSN 0899-7667. 11

L. S. Levin, D. Gates, A. Lavie, and A. Waibel. An interlingua based on domain actions for machine translation of task-oriented dialogues. In *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference, Sydney Convention Centre, Sydney, Australia, 30th November - 4th December 1998*. ISCA, 1998. URL http://www.isca-speech.org/archive/icslp_1998/i98_0999.html. 9

Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017. URL http://arxiv.org/abs/1703.03130. 13

Y. Lu, P. Keung, F. Ladhak, V. Bhardwaj, S. Zhang, and J. Sun. A neural interlingua for multilingual machine translation. *CoRR*, abs/1804.08198, 2018. URL http://arxiv.org/abs/1804.08198. 9, 21

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a. URL http://arxiv.org/abs/1301.3781. 11, 12

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013b. URL http://arxiv.org/abs/1310.4546. 11, 12

T. Mitamura, E. Nyberg, and J. G. Carbonell. An efficient interlingua translation system for multi-lingual document production, 1991. URL https://kilthub.cmu.edu/articles/journal_contribution/An_Efficient_Interlingua_Translation_System_for_Multi-lingual_Document_Production/6621035/1. 9

G. Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *CoRR*, abs/1703.01619, 2017. URL http://arxiv.org/abs/1703.01619. 5

G. Neubig and J. Hu. Rapid adaptation of neural machine translation to new languages. *CoRR*, abs/1808.04189, 2018. URL http://arxiv.org/abs/1808.04189. 2, 19

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040. 9

N. Pham, J. Niehues, T. Ha, and A. Waibel. Improving zero-shot translation with language-independent constraints. *CoRR*, abs/1906.08584, 2019. URL http://arxiv.org/abs/1906.08584. 2, 9, 21

Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig. When and why are pre-trained word embeddings useful for neural machine translation? *CoRR*, abs/1804.06323, 2018. URL http://arxiv.org/abs/1804.06323. 11, 19

S. Ruder, A. Søgaard, and I. Vulić. Unsupervised cross-lingual representation learning. In *Proceedings of ACL 2019, Tutorial Abstracts*, pages 31–38, 2019. 15

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015. URL http://arxiv.org/abs/1508.07909. 17

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009. 7

P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. *CoRR*, abs/1803.02155, 2018. URL http://arxiv.org/abs/1803.02155. 31

A. Siddhant, A. Bapna, Y. Cao, O. Firat, M. Chen, S. Kudugunta, N. Arivazhagan, and Y. Wu. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.252. URL https://www.aclweb.org/anthology/2020.acl-main.252. 20

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215. 11, 12

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762. 11, 13, 14

A. Waibel. Phoneme recognition using time-delay neural networks. *Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE)*, December 1987. 10

A. Waibel. Modular construction of time-delay neural networks for speech recognition. *Neural Comput.*, 1(1):39–46, 1989. doi: 10.1162/neco.1989.1.1.39. URL https://doi.org/10.1162/neco.1989.1.1.39. 17

A. Waibel, H. Sawai, and K. Shikano. Modularity and scaling in large phonemic neural networks. *IEEE Trans. Acoust. Speech Signal Process.*, 37(12):1888–1898, 1989. doi: 10.1109/29.45535. URL https://doi.org/10.1109/29.45535. 17

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901. 17

Appendix A.

Appendix

A.1. Training Parameters

```
general settings:
 -layers 6
 -rnn_size 300
 -word_vec_size 300
 -transformer_ff 1200
 -heads 6
 -warmup_init_lr 1e-8
 -warmup_end_lr 0.0007
 -min_lr 1e-9
 -encoder_type transformer
 -decoder_type transformer
 -position_encoding
 -max_generator_batches 2
 -param_init_glorot
 -label_smoothing 0.1
 -param_init 0
 -share_embeddings
 -share_decoder_embeddings
 -generator_layer_norm
 -warmup_steps 4000
 -learning_rate 1
```

```
vmf model:
 -dropout 0.1
 -batch_size 8192
 -batch_type tokens
 -normalization tokens
 -accum_count 2
 -optim radam
 -adam_beta2 0.9995
 -decay_method linear
 -weight_decay 0.00001
 -max_grad_norm 5.0
 -lambda_vmf 0.2
 -generator_function
    continuous-linear
 -loss nllvmf
```

```
softmax model:
 -dropout 0.2
 -batch_size 1536
 -batch_type tokens
 -accum_count 6
 -optim adam
```

```
-adam_beta2 0.999
-decay_method noam
-max_grad_norm 25
```

A.2. Example Sentences

Table A.1.: Examples for blindly decoded Portuguese-English sentences.

The fish eat the phytoplankton.	The shrimp eat the phytoplankton.
It is really old, and it is tired.	It is really old, and it is tired.
I knew something that was done on the penguin movement, so I looked at Carlos.	I was imagining a "March of the Penguins" thing, so I looked at Miguel.
We can adjust that time machine in the way we want it.	We can set that time machine on anything we want.
But, you know, this is really a serious thing because this thing is a crap, and we spend billions of dollars on it.	But, you know, it is really a serious thing because this stuff is crap, and we spend billions of dollars on it.

Table A.2.: Examples for blindly decoded English-Portuguese sentences.

Existem três conceitos de felicidade que podemos utilizar, uma por que mas mesmo.	Há realmente dois conceitos distintos de felicidade que podem ser aplicados, uma para cada eu.
E mas então fazem uma estratégia.	E aí eles o executam.
E,, para utilizar isso, comecei à ver todos estes tipos de baterias que podem ser feitas 9/11 para carros, para computadores, para celulares, para lâmpadas, para tudo, e que porque percebi que todas baterias elétricos que esse mundo usa mas porque mas porque.	E para dimensionar esse problema, eu pesquisei todos os tipos de baterias que são feitas para carros, computadores, telefones, lanternas, para tudo ; e comparando isso ao montante de energia elétrica que o mundo usa, eu cheguei a conclusão que todas as baterias que fazemos agora poderiam armazenar menos de 10 minutos de toda a energia.
E e G4s Zinny veio me dizer, disse: "na minha experiência, à menos frequentemente dito mas mas ainda mas porque que não mas	E o Dr. Kean continuou: ele disse, "Em minha minha experiência, a menos que repetidamente dito de outra forma, e mesmo se dado um mínimo de apoio, se for deixada com seus próprios recursos, uma criança fará realizações".
Obrigada. Portanto para entender mais sobre sufiicente, sim...	Obrigado. Só para eu entender melhor sobre a Terrapower, certo.

Table A.3.: Examples for blindly decoded Russian-English sentences.

Some, their less, and business.	Some, fewer still, have come from business –
And within a couple of hours, he will land, take a car car, go to Long Beach and come to one of these wonderful TED dinners.	And he's going to land in a couple of hours, he's going to rent a car, and he's going to come to Long Beach, and he's going to attend one of these fabulous TED dinners tonight.
We, gourmets, not pragmatists.	We're not realists, us foodies;
Performing music and conversations about music transformed this person from mental <unk>, who lived in central streets of L.A., in <unk>, <unk>, amazing musician, <unk> in <unk>.	And through playing music and talking about music, this man had transformed from the paranoid, disturbed man that had just come from walking the streets of downtown Los Angeles to the charming, erudite, brilliant, Juilliard-trained musician.
This is going to be a epic romantic story, passionate film.	"It's going to be this epic romance, passionate film."

Table A.4.: Examples for English-Portuguese sentences after training on backtranslated data.

E pensei, "Uau. Eu estou tipo, vivendo em um filme de ficção científica.	E eu pensava, "Uau, estou vivendo um filme de ficção científica.
E o conjunto foi baseado como uma réplica exata nos moldes do navio.	E o set estava baseado em uma réplica exata dos desenhos do navio.
E o que é pior é o onde a dor estava no seu pico no final muito final.	A pior foi aquela em que a dor aumentou no final.
Falamos sobre isso %-%.	Nós falamos dele de modo ambivalente.
Isso é fantástico. Eu amo o Grande placebo.	Isso é fantástico. Eu adoro o grande placebo.

Table A.5.: Examples for English-Russian sentences after training on backtranslated data.

Это старое исследование.	Это одно давнее исследование.
Так вот, обычно большинство людей начинают направлять себя к задаче. Они говорят о нём, они выясняют, как он будет выглядеть, они <unk> для силы.	Обычно большинство людей начинают с того, что настраиваются на работу: говорят о задании, пытаясь прикинуть, как будет выглядеть результат, исподволь пытаются добиться влияния в группе.
В этом фильме. Успех висел в равновесии ли этот эффект сработает.	Успех фильма зависел от того, сработает ли этот эффект.
А потом что делать с этими отходами?	И, наконец, что делать с отходами?
Обычно, вы просто стоите, а некоторые идут дальше, некоторые нет.	Обычно жизнь приносит нам чудо или не приносит, а не мы за ним гонимся.