IMPROVING NEURAL MACHINE TRANSLATION FOR MORPHOLOGICALLY RICH LANGUAGES

by

Raja Gunasekaran

B.Tech., Madras Institute of Technology, Anna University, 2012

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

April 2018

Abstract

Machine Translation aims to provide a seamless communication and interaction, thereby overcoming human language barriers. Recently, Neural Machine Translation (NMT) approaches have been very successful and achieve state-of-the-art performance in many language pairs. NMT systems consist of millions of neurons that are optimised to learn the input-output mapping between the source and the target languages. However, these systems produce poor translation quality under low-resource conditions and are unable to handle a large vocabulary particularly for languages with rich morphology such as Turkish, Tamil and German.

In this project, we present a source vocabulary expansion technique to handle the problem of translating rare and unknown words by incorporating morphological information in the words. The effectiveness of the proposed technique is demonstrated by translating from two morphologically rich languages to English. Using this technique, we achieve a performance gain of approximately 2 BLEU points for both German → English and Turkish → English.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human languages are very diverse and are different from each other in many aspects. There are around 6000 - 8000 languages that are currently spoken in the world. This count varies based on the definition of a *language* (Evans and Levinson, 2009). This diversity also creates a barrier in communication and interaction. Machine Translation is a promising field that can be used to overcome the human language barrier. With the recent technological advancements in communication, there is an increasing need for seamless communication and content assimilation across languages.

Machine Translation (MT) is the process of translating text automatically from one natural language to another (Russell and Norvig, 2002). The development of MT systems can be broadly classified into three: rule based approach, statistical approach, and neural network based approach. Starting from the 1980s until recently, Statistical Machine Translation (SMT) approaches like phrase-based translation models (Och, 2002; Koehn et al., 2003) gave promising results and dominated the field of machine translation. They were widely adopted and used in most of the translation engines.

Neural Machine Translation (NMT) is a recent approach to MT using neural networks. Kalchbrenner and Blunsom (2013) proposed the first, successful end-to-end system using encoder-decoder architecture for MT. This led to rapid development of more complex encoder-decoder models by Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015); Vaswani et al. (2017), each with additional functionality and improved performance. In the *Conference on Machine Translation (WMT 2015)*, only one purely neural network based MT system was submitted, and it was outperformed by a statistical MT system. In the 2017 WMT conference, almost all the systems were NMT systems (Koehn, 2017). As a result of this rapid development, popular translation engines like Google NMT, Microsoft Translator and Systran adopted NMT as a base technology for their translation system.

NMT requires a parallel corpus of source language and target language sentence pairs. These systems learn vector representation of the input words called *word embeddings*. It is a way of representing words in a language using d-dimensional *word vector* $\vec{w} \in \mathbb{R}^d$. The word vectors capture essential information about the words such as semantics and morphology. In word vectors generated using continous bag of words(CBOW) models, a simple arithmetic on them can be used to answer analogies like *man is to king as woman is to* X as shown below.

$$\vec{man} - \vec{woman} + \vec{king} \approx \vec{queen}$$

$$\vec{walking} - \vec{walk} + \vec{stop} \approx \vec{stopping}$$

NMT systems will use the information captured in the word vectors and learn an input-output mapping, from a source language to the target language. The word vectors from the input sequence passed into a neural network are first mapped to a fixed length vector as shown in Figure 1.1. From this fixed length vector, the target language sentence is generated word by word. As words occur more often, they get semantically more accurate vector representations and are translated more accurately. Because of this, NMT requires a large training dataset to learn the mapping from an input (source) language to the output (target) language.

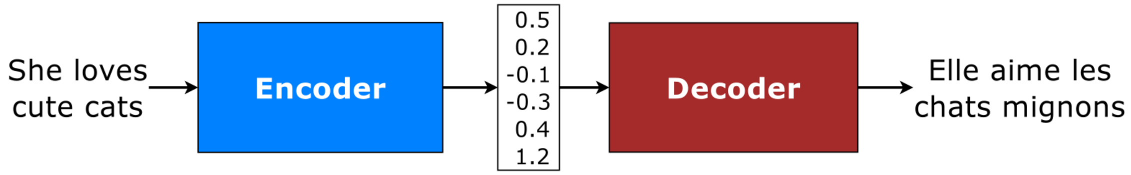

Figure 1.1: General architecture of NMT systems. Figure taken from Luong (2016)

1.1 Motivation

Although NMT systems have shown promising results in the past few years, there are a number of challenges that these systems face. Some of the major challenges are:

poor translation quality under low-resource conditions, poor translation of out-of-domain data, and inability to handle a large vocabulary (Koehn and Knowles, 2017). These challenges are usually more pronounced in languages with rich morphology.

1.1.1 Morphologically rich, low-resource languages

Morphologically rich languages such as Turkish, Tamil, German, Finnish etc., encode more information like gender, tense, number, etc. in a word as shown in Table 1.1. These languages also have a very large vocabulary since there can be large number of word forms per lexeme. Morphologically poor languages like English rely on word order (syntax) and context to convey this information.

Turkish	**English**
duy(-mak)	*(to) sense*
duygu	*sensation*
duygusal	*sensitive*
duygusallaş(-mak)	*(to) become sensitive*
duygusallaştırılmış	*the one who has been made sensitive*
duygusallaştırılamamış	*the one who could not have been made sensitive*

Table 1.1: Turkish - English Translation (Ataman et al., 2017)

Although benefits of NMT have been realized in high resource languages such as English and French, NMT is still a poor choice for languages, where parallel data for training is scarce. Neural methods learn poorly from low amount of data and hence, require a lot of data to perform well.

1.1.2 Research Problem

The primary focus of this project is to improve the quality of machine translation for morphologically rich languages under low resource settings. NMT systems use a limited vocabulary, from 30,000 to 80,000 words, to control the computational complexity during training. While this vocabulary size is large enough for languages like English, the performance of these models suffer in morphologically rich languages. To overcome this, Sennrich et al. (2015) proposed a system that can reduce the vocabulary size by splitting the words into common subwords. While this approach is sometimes sufficient, the words are not split at morphological boundaries.

Many inflected forms of words, as shown in Table 1.1, are usually scarce in the training dataset. Hence, the semantic and morphological information in the words is not captured in their word vectors. Addressing this problem can help us to improve the quality of machine translation for low-resource, morphologically rich languages.

In this project, we present a source language vocabulary expansion technique for handling a large vocabulary in NMT. This technique is particularly useful for low-resource, morphologically rich languages. The vocabulary is expanded based on morphological analysis of the Out of Vocabulary (OOV) words. For this purpose, we use a word embedding model based on sub-word units from Bojanowski et al. (2017).

To demonstrate the effectiveness of this approach, we present experimental evaluations on Turkish→English and German→English translation task. For comparison, we use global attention based NMT from Luong et al. (2015b) as a baseline.

1.2 Report outline

The project report is organized as follows. Chapter 2 gives a background on recurrent neural networks and historical approaches for machine translation. In Chapter 3, related NMT systems that handle unknown words and their techniques are presented. In Chapter 4, we discuss our proposed vocabulary expansion technique and its architecture in detail. In Chapter 5, we report on our comparison study of different techniques to handle OOV words. In Chapter 6, we present an evaluation of the proposed work on machine translation tasks for two language pairs. Finally, in Chapter 7, we conclude the project report and provide future directions to extend the work carried out in this project.

Chapter 2

Background

This section provides a background on two main topics of the project: Recurrent Neural Networks (RNN) and Machine Translation (MT). We begin by looking into neural networks and why RNNs are well suited for MT. Then, we will look into some of the historical and current approaches for machine translation.

2.1 Neural Networks

Rojas (2013) define neural networks as "distributed, adaptive, generally nonlinear learning machines built from many different processing elements". They can approximate a function by learning the input-output mapping. Neural networks like feed-forward multi-layer perceptrons, shown in Figure 2.1, can approximate any (continuous) function to an arbitrary accuracy if the number of hidden neurons are large enough (Hornik et al., 1989). These networks can be trained using the backpropogation algorithm (Rumelhart et al., 1988) by updating the weights and biases based on an objective function. As the training progresses, the network updates its weights in a way that its predicted output moves closer to the ground truth[1]. A one layer feed-forward neural network can be written as follows:

$$NN_{MLP1}(x) = g(xW^1 + b^1)W^2 + b^2 \tag{2.1}$$

where g is any non linear activation function, x is an input vector, W^1, W^2, b^1 and b^2 are the weights and biases for the network.

Unlike other machine learning approaches where input features have to be hand-engineered, neural networks can learn the required features from the training data.

One class of neural network model called Recurrent Nerual Network (RNN) (Elman, 1990) is particularly well suited for machine translation. In natural languages, the input is a sequence of words of arbitrary length based on some structure properties of the language. RNNs allow for representing an input sequence of arbitrary length as fixed-sized vectors based on its structural properties. A simple RNN (Goldberg, 2016) can be defined as follows.

$$h_{1:n}, y_{1:n} = RNN(h_0, x_{1:n}) \tag{2.2}$$

$$h_i = R(h_{i-1}, x_i; \theta) \tag{2.3}$$

$$y_i = O(h_i; \theta) \tag{2.4}$$

[1] ground truth - reference output provided by direct observation as opposed to inference

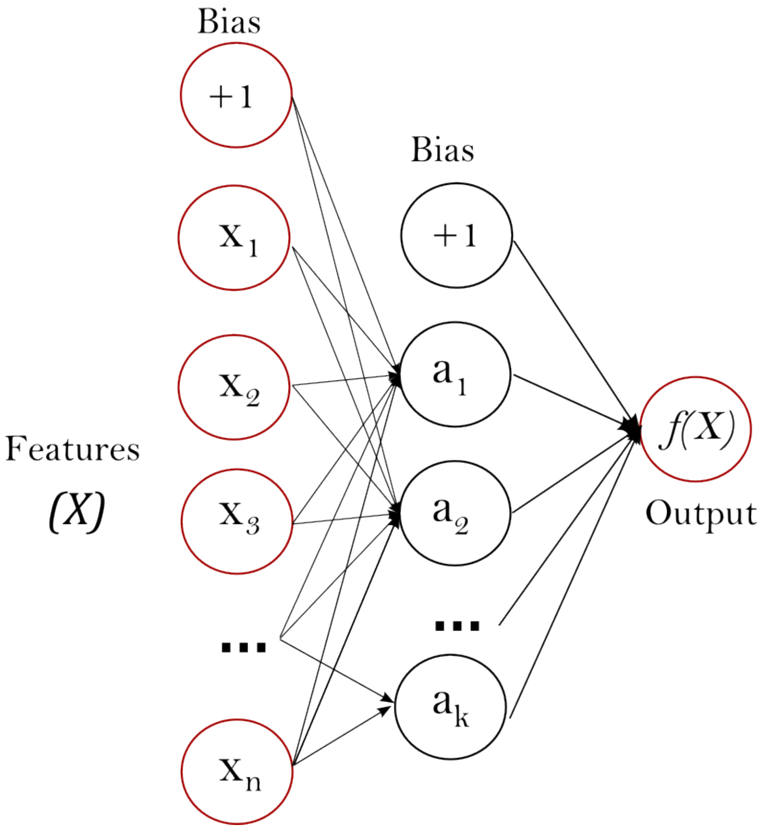

Figure 2.1: Multilayer perceptron network with one hidden layer. Figure taken from Pedregosa et al. (2011)

.

$$x_i \in \mathbb{R}^{d_{in}}, y_i \in \mathbb{R}^{d_{out}}, h_i \in \mathbb{R}^{f(d_{out})}$$

where $x_{1:n}$ is the input vector, $y_{1:n}$ is the output vector and h_i is the state vector at time-step i. R is a non linear function applied over current input x_i and previous hidden state s_{i-1}. O is an additional function applied over the current hidden state to generate output vector. Parameters θ are shared across the network. A simple RNN uses *sigmoid* or *tanh* as the non linear function in the neural units. Special kinds of neural units like Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber,

1997) or Gated Recurrent Units (GRU) (Cho et al., 2014) can also be used. A graphical representation of the same network is shown in Figure 2.2

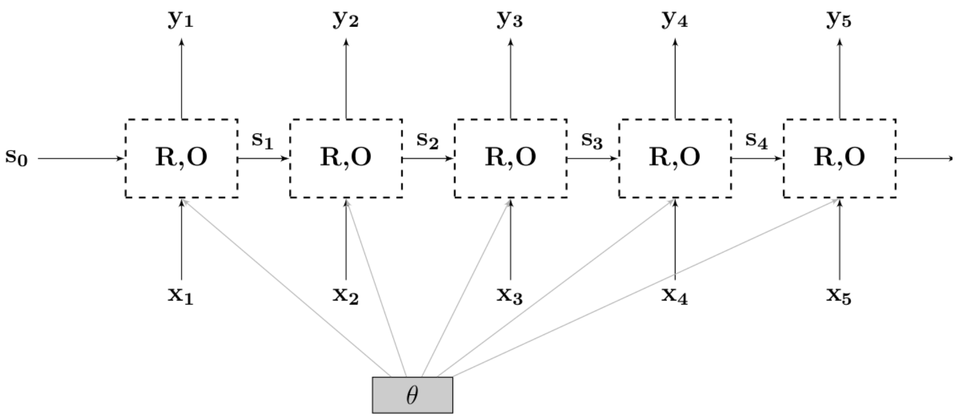

Figure 2.2: Graphical representation of RNN. Figure taken from Goldberg (2016)

.

The input for these neural networks is a sequence of vectors not words. NMT systems use word embeddings to represent input words. It is a way of representing words in a vocabulary as vectors in a higher dimensional space. These representations have been good at capturing semantic and syntactic regularities in languages (Mikolov et al., 2013). When used as input, these models have also been shown to improve the performance of many NLP systems in tasks such as *MT* (Vaswani et al., 2017; Sennrich et al., 2015), *sentiment classification* (Kumar et al., 2016), and *part of speech tagging* (Kumar et al., 2016). The vector representation for words in these models primarily depend their co-occurrence count with other words within a window size. These vectors capture syntactic, semantic and morphological properties of the words.

One of the major challenges for many embedding models is their inability to handle

Out Of Vocabulary (OOV) words. This problem is more pronounced in languages with rich morphology. A word in morphologically rich languages encodes more information (such as gender, number, tense) as compared to morphologically poor languages which rely on word order and context. Recently, many models have been proposed to solve this problem. Sun et al. (2016) proposed a method to integrate both external contexts and internal morphemes[2] to learn better word embeddings especially for rare and unknown words. Bojanowski et al. (2017) incorporate subword information like character n-grams[3] for learning word embeddings during training. Overall, character embedding models, where vector representations for every character or character n-grams are learned, have be shown to generate good word embeddings for rare and unknown words.

2.2 Statistical Machine Translation

Machine translation is a task of translating a source language sentence F to the target language sentence E using computing resources. It can effectively remove language barriers between humans, allowing assimilation of content from different languages. This potential of MT has led to substantial amount of research since the advent of digital computing. There have been many approaches to the problem such as rule-based MT, phrase-based MT, NMT, etc. In the early days, rule based approaches

[2] Morphemes are the smallest meaningful units in a language.

[3] A character n-gram is a sequence of n characters from a word

that used dictionaries, grammar and pre-defined rules to translate text were explored until the ALPAC report in 1966 (ALPAC, 1966). The ALPAC report showed that post-editing machine translation was not cheaper or faster than human translation.

In the late 1980s, following the success of statistical method on speech recognition, IBM research (Brown et al., 1993) modelled the problem of translation as a statistical optimization problem. Many SMT approaches such as word-based models (Brown et al., 1993), phrase-based models (Koehn et al., 2003; Marcu and Wong, 2002), hierarchical phrase-based models (Chiang, 2007) and syntax based models Galley et al. (2004, 2006) were proposed. The goal of all the SMT systems is to maximize the probability of target sentence f given the source language sentence e

$$\underset{f}{argmax}\ P(f|e) = \underset{f}{argmax}(P(f) \times P(e|f)) \tag{2.5}$$

where $P(f)$ is a language model and $P(e|f)$ is a translation model. Language model and translation model are sub-components of SMT systems. Language models learn to assign probability to a sequence of words in a language. They assign higher probability to sentences that are more likely to occur in the language and hence a measure of fluency in the language. Language modelling is central to many tasks in NLP including MT.

Translation models learn the mapping between source and target language words or phrases. They measure word level translation accuracy between source and target sentences. These models were built by analyzing monolingual, bilingual corpus and learning their probability distribution. The rise of digital text resources like parallel corpora and an increase in computing power and storage, fuelled the growth of SMT systems. Although SMT systems are robust to noisy data, they required fine tuning for many components such as language model, reordering model, and translation model for each language pairs. They also require large amount of data and do not handle long range dependencies well.

2.3 Neural Machine Translation

An NMT system is a neural network that models the conditional probability $p(y|x)$ of generating a target language y sentence given the source language sentence x. Generally, any NMT system consists of two components i) an *encoder* that computes a sentence representation vector from the source sentence, and ii) a *decoder* that transforms the vector representation to a target language sentence. RNNs are a common choice of network for both encoders and decoders as they process input in a sequential fashion. Convolution neural networks have also been used especially as an encoder (Kalchbrenner and Blunsom, 2013).

Initially, neural networks were used as a component in phrase based systems to score the quality of translation (Schwenk, 2012) and to provide additional features to SMT systems (Zou et al., 2013). Kalchbrenner and Blunsom (2013) proposed the first end to end approach for NMT using convolution neural networks as the encoder and RNN as the decoder. Their network suffered from the problem of vanishing gradients where the network was unable to capture long range dependencies. To overcome this problem, more sophisticated activation functions such as LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014) are used. Sutskever et al. (2014) and Cho et al. (2014) demonstrated that these gated units can handle long range dependencies better than a simple RNN (Elman, 1990). An encoder-decoder network with LSTM RNNs is shown in Figure 2.3. The boxes in the figure are LSTM RNN units. Mathematically, an LSTM RNN unit is defined in Goldberg (2016) as follows:

$$s_j = R_{LSTM}(s_{j-1}, x_j) = [c_j : h_j] \tag{2.6}$$

$$c_j = c_{j-1} \odot f + g \odot i \tag{2.7}$$

$$h_j = tanh(c_j) \odot o \tag{2.8}$$

$$i = \sigma(x_j W^{xi} + h_{j-1} W^{hi}) \tag{2.9}$$

$$f = \sigma(x_j W^{xf} + h_{j-1} W^{hf}) \tag{2.10}$$

$$o = \sigma(x_j W^{xo} + h_{j-1} W^{ho}) \tag{2.11}$$

$$g = tanh(x_j W^{xg} + h_{j-1} W^{hg}) \tag{2.12}$$

$$y_j = O_{LSTM}(s_j) \tag{2.13}$$

$$s_i \in \mathbb{R}^{2 \cdot d_h}, x_i \in \mathbb{R}^{d_x}, [c_j, h_j, i, f, o, g] \in \mathbb{R}^{d_h}, W^{xo} \in \mathbb{R}^{d_x x d_h}, W^{ho} \in \mathbb{R}^{d_h x d_h}$$

where x_j, y_j and h_j are the input vector, output vector and the hidden vector repectively. $\odot$ is component-wise product. i, f and o are input, forget and output gates respectively. g is the update candidate.

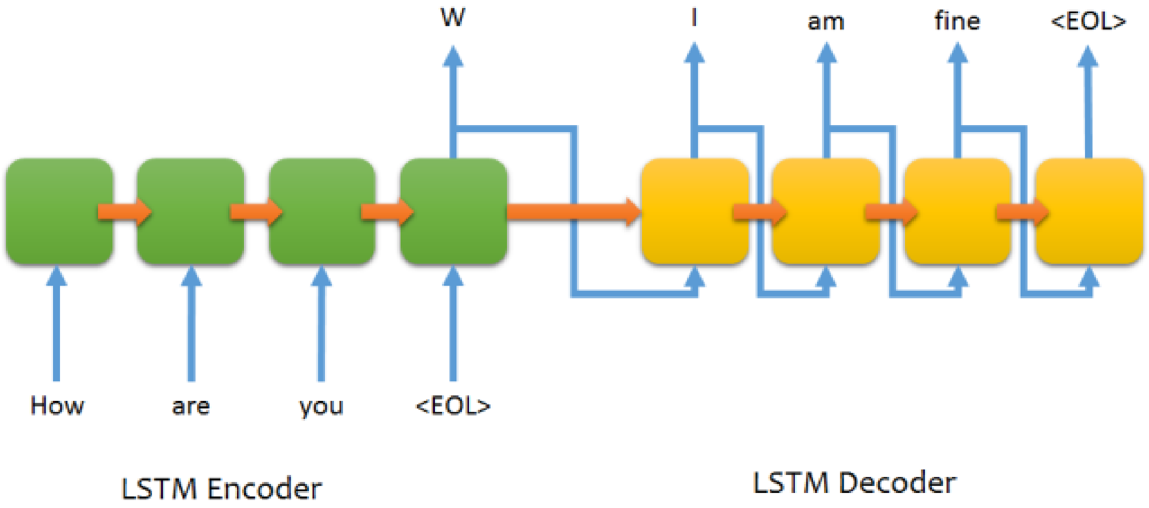

Figure 2.3: RNN with LSTM encoder and decoder. Figure taken from Rahman (2017)

.

Attention based Encoder-Decoder models

These simple encoder-decoder networks summarize the source sentence in a fixed length context vector. Bahdanau et al. (2015) noted that these networks were inadequate to represent long sentences due to the fixed dimension of the context vector. While this problem could theoretically be solved by increasing the dimension of the context vector, the computing power and memory required to train such a network sets an upper limit even today.

Bahdanau et al. (2015) proposed an attention based encoder-decoder architecture to mitigate this problem. This allowed for the encoder to produce better sentence representation for longer sentences which in turn improved the translation quality. In their additive approach, a single feed forward neural networks that can learn to assign different weights to the hidden layer vectors was used. These weighted sum of the hidden layer vectors $h_{1:n}$ called context vector c_i is calculated each time decoder generates a new word as shown in Figure 2.4.

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j \tag{2.14}$$

$$\alpha_{ij} = \frac{\hat{a}_{ij}}{\sum_j \hat{a}_{ij}} \tag{2.15}$$

$$\hat{a}_{ij} = att(s_i, h_j) \tag{2.16}$$

where $att(s_i, h_j)$ is an attention function that calculates the weights for each encoder hidden state $h_{1:n}$, for a given decoder state s_i. They also used bi-directional RNN which reads the sentence from both directions. The state vector from both direction right to left $\overleftarrow{h_j}$ and left to right $\overrightarrow{h_j}$ is concatenated for each word. The attention mechanism is applied over this concatenated hidden state vector $h_j = [\overleftarrow{h_j}; \overrightarrow{h_j}]$. The whole network is trained with negative log-likelihood as the objective function using stochastic gradient descent.

Luong et al. (2015b) proposed two simpler but effective variations of attention mechanism namely: 1) a global attention model where all words in the source sentence are attended, and 2) a local attention model where only a subset of the source words are attended at a time. In the global attention mechanism, which is very similar to one proposed in Bahdanau et al. (2015), they did away with bi-directional, concatenated state vector $h_j = [\overleftarrow{h_j}; \overrightarrow{h_j}]$. They simplified the computation path to make it run faster.

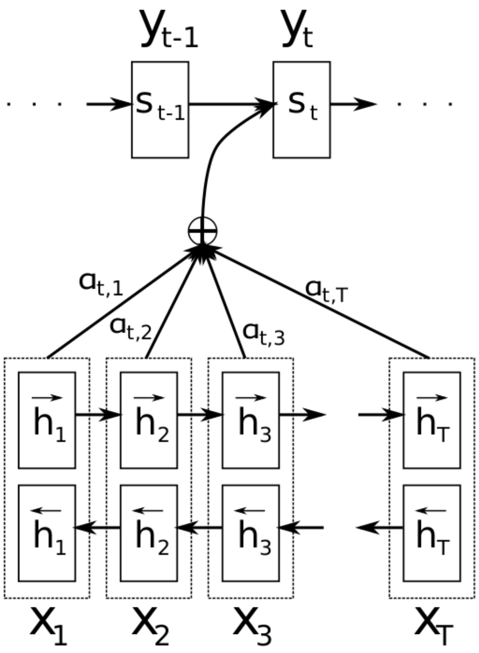

Figure 2.4: Attention based encoder. Figure taken from Bahdanau et al. (2015)

.

Their model produced state-of-the-art results in WMT'14 and WMT'15 for English to German translation.

In this project, we use the global attention based encoder-decoder model proposed in Luong et al. (2015b) as the baseline NMT system for evaluation. The models discussed so far cannot handle words that are not in their vocabulary. In the next chapter, we will look at some of techniques that allow us to handle OOV words.

Chapter 3

Handling Rare and Unknown Words

NMT systems typically have a vocabulary size of 30,000 to 80,000 words. Until recently, these models were incapable of translating rare and unknown words (Luong et al., 2015a). As the vocabulary size of the network grows, the complexity and the number of parameters to tune increases rapidly. Pioneering works in NMT (Sutskever et al., 2014; Bahdanau et al., 2015) observe that sentences with rare and unknown words often produced poor translations when compared to sentences with many frequently used words. Since the network has seen these words only a few times, they don't get a vector representation that can capture their semantics and morphology.

Luong et al. (2013) note that rare and unknown words are usually morphologically rich in nature in comparison to more frequently occurring words. Although the problem of handling unknown and rare words was focused heavily in the context of generating word representations, it was not addressed in any of the early works in NMT (Luong et al., 2015a). In this chapter, we present some of the general techniques for handling rare and unknown words. First, we will discuss how this problem is handled in the context of word embeddings. Then, we will explore NMT specific approaches to handle OOV words.

3.1 Vector representation for OOV words

Representing words as vectors has a long history in NLP. A good vector representation should capture the syntax and semantics of a word. It should mirror the linguistic relationship between the words in the vector space. Embedding models discussed in Bengio et al. (2003), Collobert and Weston (2008), Mikolov et al. (2013), Pennington et al. (2014), etc., learn word representations in a continuous vector space where semantically similar words occur closer to each other. Use of word representation, learned from these models, have been shown to improve performance across all NLP tasks (Kumar et al., 2016). These systems are usually trained on a large monolingual corpus with billions of sentences. Then, the learned word embeddings can be used to represent individual words in the downstream tasks like sentiment classification, text summarization, machine translation, etc.

These models work at word level and ignore the internal structure of the words. Only the words that the model has seen in the training data will get a vector representation. To get vector representations for OOV words, morphological and orthographic information can be used (Botha and Blunsom, 2014; Luong et al., 2013; Bhatia et al., 2016) while training. These models require an external morphological analyzer to segment the word which is not readily available for all languages. Soricut and Och (2015) proposed an unsupervised, language agnostic method to extract morphological rules and to build a morphological analyzer. In their approach morphological transformations can be captured and used to map OOV words to an in-vocabulary word.

An alternate approach is to get embeddings for characters or character n-grams by breaking down words (Bojanowski et al., 2017; Kim et al., 2016; Wieting et al., 2016). This allows us to get word embeddings for any words by convoluting over character embeddings. Among these models, Bojanowski et al. showed that, by incorporating character n-grams their models can outperform other word level or morphology based approaches. They learn representations for character n-grams and represent word as the sum of character n-grams.

For this project, we have used the work of Soricut and Och (2015) and Bojanowski et al. (2017) to represent OOV words. we have implemented and studied the morphological analyzer from Soricut and Och (2015) on a word similarity task and used the pretrained word representations from Bojanowski et al. (2017) for translation task.

3.2 Handling OOV words in NMT

Unlike Statistical approaches, NMT systems have a fixed vocabulary due to computational complexity of the model. This forces us to handle words outside this fixed vocabulary using some other techniques. Initially, almost all the NMT systems represent all the OOV words using an unknown token *unk*. Later, different approaches were introduced to address this problem. They can be broadly classified into two groups: Dictionary back-off models, and Subword models.

3.2.1 Dictionary back-off Models

Jean et al. (2014) proposed a method based on importance sampling to use a very large target vocabulary without increasing the complexity of the NMT system. In their attention based NMT, the attention weights were used to determine the alignment of unknown (*unk*) target words with the corresponding source word; usually a rare word, in the translation. Then a dictionary is used to replace the *unk* tokens in the target languages with the translation of the rare source word.

Luong et al. (2015a) proposed a similar model using an external aligner instead of using the attention mechanism. An external aligner was used to align words from the source sentence and the generated target sentence. During training, for all the unknown source words, the aligned target word is replaced with an unknown token *unk*. In their copy attention model, each unknown target word is assigned individual *unk* token based on their source word. The alignments between source words and target words are maintained. In their positional model, a pre-build dictionary is used to replace the unknown source words in the target side, based on the alignment. Both these approaches were effective and showed a better performance than the state-of-the-art in English-French and English-German language pairs. Choi et al. (2017) extended the work of Luong et al. (2015a) to include multiple positional unknown tokens for digits, proper nouns and acronyms instead of just one *unk* token.

3.2.2 Subword and character Models

One problem with these dictionary back-off methods is that there is not always 1-1 correspondence between words from different languages because of the variance in degree of morphological synthesis between languages.

Luong and Manning (2016) presented a open vocabulary NMT system based on word and character embedding models using RNN. In their hybrid approach, the network translates at word level and falls back to character components for rare words. The representation for rare words is computed using Recurrent Neural Network working on the character level. Their system is faster and easier to train unlike other NMT systems using purely character representations. They produced state-of-the-art for English-Czech translation on WMT'15 dataset.

Sennrich et al. (2015) proposed a system that works on subword level instead of word level like the previous models. The words are segmented into subword units using Byte Pair Encoding (BPE) (Gage, 1994). BPE is a data compression technique, where the most frequency character or character sequences are iteratively replaced with unused bytes. The vocabulary of their NMT system comprises entirely of these subword units of different lengths.

They demonstrated that subword models achieve better accuracy in translating rare words. The model was able to generate new unseen words during testing time and improved English-German and English-Russian translation over other back-off dictionary models. One of the major contribution of Sennrich et al.'s paper was showing that the NMT systems are capable of achieving open vocabulary translation by modelling sub word units. This technique has been used widely to improve translation quality.

BPE does not split words based on their morphology. So the morphological information that the network already learned is not used. In the next chapter, we present how this information can be used to improve the quality of machine translation.

Chapter 4

Vocabulary Expansion for NMT

In this Chapter, we present our proposed approach for improving the quality of machine translation for morphologically rich languages. In Section 4.1, we present the challenges in translating a language with rich morphology. In Section 4.2, we describe in detail the proposed vocabulary expansion technique as a solution to the problem. All the components implemented for this study and their architectures are presented in Section 4.3. Finally, in Section 4.4, the implementation details of all the modules are discussed.

4.1 Problem

NMT systems have been very successful in achieving state-of-the-art performance for many language pairs such as English-French and English-German. However, these systems are not capable of translating rare and unknown words (Luong et al., 2015a).

All NMT systems have a fixed vocabulary of 30k-80k most frequently occurring words. So, words that do not occur or occur rarely in the training data will not get good vector representations. This results in poor translation performance on sentences with rare and unknown words (a.k.a OOV words) (Sutskever et al., 2014; Bahdanau et al., 2015).

This problem with translating OOV words is more pronounced in morphologically rich languages like Turkish, German, Tamil etc. There are two factors that primarily contribute to this:

- In morphologically rich languages, due to inflections, there are many possible *word forms* per lexeme as shown in Table 1.1. For example, even in a morphologically impoverished language like English, we have *run, running, ran, runs* which are all the forms of the same lexeme *run*. So, the vocabulary size in these languages is very high resulting in **more OOV words**.

- NMT requires a large amount of training data for higher performance. Many of these languages have a much **less data** available for training a translation system. For example, there are only 300k sentence pairs available for Turkish-English and approximately 200k sentence pairs available for Tamil-English.

To overcome these problems, we need a system that can handle OOV words effectively by understanding the morphology of the words. This system should be capable of mapping the OOV words to an in-vocabulary word using morphological transformations. So, to address this problem, we need to answer the following questions:

1. How do we get vector representations for OOV words, that captures the linguistic properties of a word?

2. Can we improve the performance of NMT systems using vector representations that exhibit linguistic regularities?

4.2 Proposed Vocabulary Expansion Technique

The main objective of this project is to improve the quality of machine translation for morphologically rich languages by incorporating morphological information. In particular, we focus on improving the translation quality under low-resource settings - when only a smaller amount of data is available for training. To address this problem in NMT systems, we propose and study a simple source vocabulary expansion technique that can translate any source word.

Word embedding models proposed in Mikolov et al. (2013); Pennington et al. (2014) have been shown to exhibit linguistic regularities in the form of semantics and morphology as shown in eq 4.1 and eq 4.3 (Soricut and Och, 2015).

$$\vec{man} - \vec{woman} + \vec{king} \approx \vec{queen} \tag{4.1}$$

$$\vec{stop} + (\vec{walking} - \vec{walk}) \approx \vec{stopping} \tag{4.2}$$

$$\vec{stop} + (\vec{suffix{:}ing}) \approx \vec{stopping} \tag{4.3}$$

Inspired by this observation, we hypothesize that fixed word embeddings can be used to expand the source side vocabulary in any NMT system. Instead of letting the NMT system learn word representation, we use a *fixed* pre-trained word representation during training. During training, the NMT system learns the mapping from source language word embeddings to target language word embeddings.

By fix the word embedding, we achieve the following benefits:

- We retain the linguistic regularities present in word representations and feed them to a translation system.

- We reduce the number of parameters to tune for NMT systems resulting in faster training.

During testing, for any OOV source language word, we generate a word representation which maintains the linguistic regularities. Consider this example: We have a vector $(\vec{suffix} : ed)$ that can transform all the present tense words to their corresponding past tense word in the vector space. If we encounter an OOV word like $\vec{enthralled}$, we can use the vector for the in-vocabulary word $\vec{enthrall}$ and $(\vec{suffix} : ed)$ to generate a meaningful vector representation. This vector can now be fed into the NMT system instead of mapping it to *unk* word. There have been many approaches proposed that make use of pre-trained word embeddings. But, to the best of our knowledge, vocabulary expansion based on fixed word embedding have not been studied.

To generate these morphological transformations, we experiment with two models:

- Language agnostic, unsupervised morphological transformation approach from Soricut and Och (2015)

- Word representation using subword information from Bojanowski et al. (2017)

From our experiments, discussed in next chapter, we find that Bojanowski et al.'s approach generates a more accurate vector OOV word. We use this model in our final NMT system. To study the effectiveness of the proposed approach, we implement this technique on a baseline model and report the performance gains. For baseline NMT systems, we implement the global-attention based encoder-decoder RNN model from Luong et al. (2015b).

4.3 Project Components

In this section, we discuss the baseline NMT systems and the morphological analyzer implemented for this project. The architecture of these two systems is explained in detail below.

4.3.1 Global Attention Based Model

Simple encoder-decoder NMT models (Sutskever et al., 2014; Cho et al., 2014) do not translate long sentences very well. To overcome this problem, attention-based NMTs are used. Their networks learn word alignments between the source and target language sentence. Luong et al. (2015b) proposed two simple but effective attention based NMT models as an alternative to the attention model from Bahdanau et al. (2015).

Similar to Bahdanau et al.'s approach, their *global attention* model focuses on all the source words for every target word. Their *local attention* model focuses only on a small subset of the source words. For this project, we use the global attention model as the baseline.

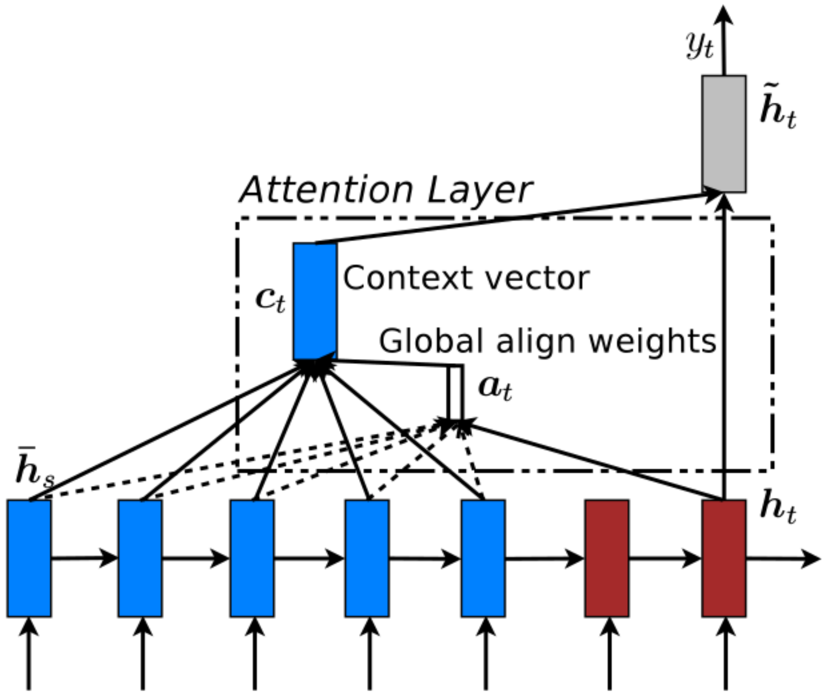

Figure 4.1: Attention based encoder. Figure taken from Luong et al. (2015b)

.

Figure 4.1 presents a visual representation of the global attention model. The context vector c_t is calculated based on alignment weights a_t for every target word. To calculate attention weights, they experimented with three different scoring functions as shown in eq 4.4. They noted that with the global attention model, a simple *dot* product between encoder and decoder hidden layer vectors performs the best. We use this model with *dot product* as the scoring function.

$$score(\overrightarrow{h_j}, s_i) = \begin{cases} \overrightarrow{h_j}^\top s_i & dot \\ \overrightarrow{h_j}^\top \mathbf{W}_a s_i & general \\ v_a^\top \mathbf{W}_a [\overrightarrow{h_j}, s_i] & concat \end{cases} \tag{4.4}$$

4.3.2 Unsupervised Morphological Analyzer

Soricut and Och (2015) proposed a language-agnostic, heuristic method to capture morphological transformations by exploiting regularities present in word embeddings (Mikolov et al., 2013). Their method automatically deduces morphological rules and transformations from word vectors. These morphological transformations can be represented as vectors in the same embedding space. During testing, OOV words can be mapped into the same vector space using the learned morphological transformations. In this algorithm, the morphological rules are learned as follows.

1. Extract candidate morphological rules like *('suffix', 'ies', 'y')* (replace suffix *ies* with *y* from the word *treaties* to get *treaty*) from every possible word pairs in vocabulary V.

2. Then, evaluate the quality of rules in the pre-trained embedding space by calculating the hit-rate for other word pairs.

3. Generate morphological transformation from the above candidate rules and build a cyclic multi-graph, where words form the nodes and edges form the

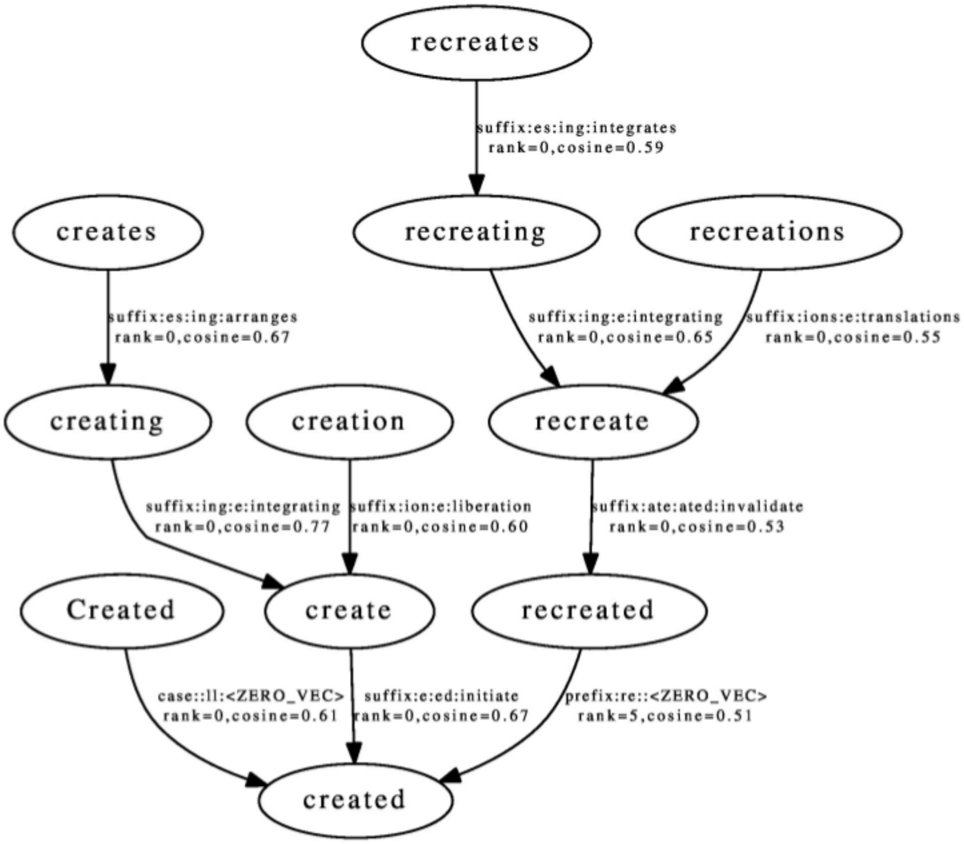

Figure 4.2: A part of normalized, directed graph with morphological mapping. Figure taken from Soricut and Och (2015)

.

morphological transformations. The edges between the words are weighted using cosine distance between their vectors and their ranks $(rank, cosine)$.

4. From the above graph, build a normalized acyclic graph with 1-1 morphological mapping between words. This graph is normalized by mapping low-frequency words to high-frequency words as shown in Figure 4.2. In this graph, all the low-frequency words in the dataset *(recreates, recreations, creates)* are mapped

to higher frequency words *(create, created)*.

5. Use the final graph and the learned morphological transformations to map the rare/out of vocabulary words in the same vector space.

Using this approach, if the word *unassertiveness* occurs in the source sentence and is not found the vocabulary of the word embedding model, we would still be able to get a meaningful vector representation for the word. Traditionally, any word not in the vocabulary is mapped to the token *unk*. Using this approach, we can learn vector representation for morphological transformations like *(prefix,un,ϵ)* and *(suffix,ϵ,ness)*. These morphological rules and their vector representation can then be used to map the OOV word *unassertiveness* to *assertive* and get a good vector representation.

4.4 Implementation Details

In this section, we will present the implementation details of *attention based NMT* and *morphological analyzer* for this project.

The baseline NMT system was implemented from scratch in *pytorch*, a deep learning framework. Later, we switched to *OpenNMT-py*[1] for the fine-tuned performance gains it offers. OpenNMT-py is a collaborative research-friendly framework focusing specifically on machine translation.

We implemented the morphological analyzer purely in Python. We used *gensim*[2]

[1] https://github.com/OpenNMT/OpenNMT-py

[2] https://github.com/RaRe-Technologies/gensim

to calculate word similarities and their rank in the vector space. For building and storing graphs, we used *networkx*[3] graph library. Finding the nearest neighbour in 300 dimension vector space, with 3M candidates is expensive. Since we do this calculation many time for each word, it can takes days to run on a standard computer. To speed up computation, we used k-d tree based Approximate Nearest Neighbour (ANN) algorithms from *annoy*[4] library under low precision settings.

[3] https://networkx.github.io

[4] https://github.com/spotify/annoy

Chapter 5

Vector Representations for OOV words

In this chapter, we compare different approaches to generate vector representations for OOV words. Soricut and Och's approach is implemented and compared with other existing approaches. The goal of this study is to compare various techniques and use the best performing model in NMT. The models reported in this study are not specific to machine translation. However, they are general methods used to generate vector representations for any word based on its morphology, or the subword units present in them.

In this experiment, we compare the performance of three popular pre-trained word embeddings models (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) and the implemented morphological transformation approach on standard word similarity dataset. We try to answer the following questions through this study.

- What are all the techniques to generate a vector representation for OOV words and their performance?
- How does the implemented morphological transformation technique compare against other approaches?
- Which technique would be ideal to use in NMT systems for handling OOV words?

5.1 Dataset and Evaluation Metric

We use the Stanford **Rare Word (RW) Similarity dataset** for English from Luong et al. (2013) to evaluate the vector representation of rare words. This dataset contains 2034 morphologically complex word pairs with similarity scores for each pair. They used Amazon mechanical turk to collect 10 human similarity rating on a scale of 0 to 10 for each word pair. The average of all the human judgement scores is taken as the similarity score. The words in RW dataset have a higher degree of English morphology compared to other word-similarity datasets. So, an evaluation on this dataset can be used as a measure of the ability to handle rare and unknown words.

To study the performance, we use **Spearman's rank correlation coefficient** ρ (Spearman, 1904). The value varies from 0 to 1, with 1 indicating perfect correlation. We use it to report the correlation of the cosine similarity values between the word vectors for each pair, and the human judgement scores.

5.2 Results

The performance of all the models on RW dataset is presented in Table 5.1. For skipgram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) models, we used the publicly available pre-trained word embeddings. On top of skipgram vectors and GloVe vectors, we induce morphological transformations and report their performance. For both the fastText model, we use their pre-trained model trained on Wikipedia dump[1].

For *fastText + subword* model, we use the pretrained sub-word model from (Bojanowski et al., 2017) to handle OOV words. We used this instead of *fastText + morph* for two reasons: a) *subword* model can handle any OOV word whereas the *morph* model cannot handle words for which the morphological transformation are not extracted from available words. b) *fastText + subword* model is one best performing model in RW dataset reported in the literature (Bojanowski et al., 2017).

From Table 5.1, it can be seen that inducing morphological transformations improves the Spearman correlation between human judgement and predicted cosine similarity. By using this method, vector representations for approximately 10% more words were obtained for SkipGram and GloVe. The high correlation indicates that the vector representations for OOV words (10%) are almost as good as vector repre-

[1] https://fasttext.cc/docs/en/pretrained-vectors.html

Word Embedding Models	**Spearman Correlation** $\rho \times 100$	**No of word pairs handled** (2034)
SkipGram	22.9	1825
SkipGram + morph	24.5 **(+1.6)**	1988
GloVe 6B	22.5	1782
GloVe 6B + morph	25.9 **(+3.4)**	1996
fastText	29.9	1966
fastText + subword	36.76	2034

Table 5.1: Performance of pre-trained and morphological transformation induced word embeddings on RW dataset

sentations for the known words.

Despite the increase in performance by using morphological transformations, we were not able to map all the OOV words to an in-vocabulary word. We were able to generate word vectors for only 1996 out of 2034. Even the baseline fasttext model from Bojanowski et al. (2017) outperforms all the other models by a good margin. When OOV words are handled based on their subword information, we see approximately 7-point increase in the Spearman correlation ρ. Also, the *fasttext + subword* model is able to generate vector representation for all the words.

5.3 Implementation challenges

In our implementation of morphological induction, we were able to improve on the baseline by approximately 2 points. However, we were not able to replicate the results of the original paper. Soricut and Och used a 500-dimension skipgram model as their baseline model. They report a 6-point increase in Spearman ρ over their baseline model. There are a few differences in the implementation from the original model, that may have hindered the algorithm from generating better morphological transformations.

In our implementation, the morphological rules were extracted only from 100,000 most frequent words, from a vocabulary of 3 million words. This was done to allow quick turn around time for implementation and experimentation of the algorithm. Common words are morphologically poorer as compared to rarer words. Extracting morphological rules from all the words in vocabulary may result in further increase in the correlation.

To find the nearest neighbour for a given vector in the vector space, we used a k-d tree based nearest neighbour search. We used 100 trees to build the indexing for all the vectors. This allows for faster querying at the cost of precision. Building the index using more number of trees, say 300, will allow for higher precision at the cost of querying time.

Summary

From this experiment, we observe that Bojanowski et al.'s approach generates a semantically more accurate vector representation for words in RW dataset, as compared to other models. The model can also generate accurate word representations for all possible words. This is a very important property that can aid translation for morphologically rich languages. Based on this result, we use the fasttext model in NMT system to handle OOV words which we will discuss in the next chapter.

Chapter 6

MT for Morphologically Rich Languages

In this chapter, we present how we use the fasttext model to expand source vocabulary and improve the quality of machine translation for morphologically rich languages. In Section 6.1, we describe the dataset used for the translation task. We present the pre-processing steps and training details in Section 6.2. In Section 6.3, we describe the evaluation metric used to report the quality of generated translations. The performance of the proposed approach is presented in the Section 6.4 and Section 6.5.

For the proposed approach, the translation task is performed on two language pairs: German → English and Turkish → English. Both these languages are morphologically rich and well studied in previous works (Bahdanau et al., 2015; Luong et al., 2015b; Sennrich et al., 2015; Gulcehre et al., 2015), particularly German - English translation. They also have a very large vocabulary making them ideal candidates for this study. German → English is high resource task in machine translation with 4.5M sentence pairs available for training. But, Turkish → English is low resource task with around 160,000 sentence pairs available for training. For the experiments in this study, both of the languages are studied under low-resource settings; since the goal of this study is to improve translation performance for morphologically rich, low-resource languages.

6.1 Dataset

For both of the language pairs, we used the WIT^3 dataset (Cettolo et al., 2012) from IWSLT'14 machine translation track (Cettolo et al., 2014). The dataset consists of sentence aligned subtitles for TED and TEDx talks. WIT dataset contains 149,000 sentence pairs for Turkish - English and 160,000 sentence pairs for German - English. During testing time, the translation performance is reported on test data from the WIT^3 corpus. Additionally, German → English model will also be evaluated newstest2012 (3000 sentences) and newstest2013 (3000 sentences) dataset from ACL WMT'14. We do this as a measure of generalization, since the newstest data is from a different domain than training data.

In addition to above datasets, we used the pre-trained word embedding model[1] from Bojanowski et al. (2017) for English, German and Turkish. These word vectors

[1] https://fasttext.cc/docs/en/pretrained-vectors.html

	Turkish	English
No. of sentences	149k	
No. of words	3.3M	2.7M
Unique words	160k	48k

(a) Turkish - English

	German	English
No. of sentences	160k	
No. of words	3M	3.1M
Unique words	112k	49k

(b) German - English

Table 6.1: Training data statistics

are trained on Wikipedia data [2].

6.2 Training

The data was preprocessed using a data preparation script[3] from Ranzato et al. (2015). The Moses tokenizer is used to tokenize the sentences. All sentences are converted to lowercase and sentences with more than 80 words are removed. Detailed statistics of the parallel corpora used for training is given in Table 6.1. The source and target vocab size was limited to 50k for training.

In all our models, the embedding layer dimension is 300. The number hidden layers in encoder and decoder is set to 2, each layer with 500 dimensions. We trained all our models for 20 epochs, using Adam optimizer and used the best performing model based on accuracy and perplexity for evaluation. The batch size was fixed to

[2] https://dumps.wikimedia.org/

[3] https://github.com/facebookresearch/MIXER/blob/master/prepareData.sh

128. As suggested Luong et al. (2015b), we use a dropout probability of 0.2 for our LSTMs. We trained our models on Nvidia Tesla P100 GPU, from Compute Canada[4], where we achieved a speed of 5k target words per second.

6.3 Evaluation

To be comparable with other existing NMT systems, we have evaluated the models using standard BLEU score metric from Papineni et al. (2002). BLEU is one of the most popular and widely used automatic MT evaluation metric. The score varies from 0 to 100 and higher scores denote better translation. It reports the correlation of machine translations with human translations measured using the degree of n-gram overlap. As a baseline model, the attention model from Luong et al. (2015b) was used. Then, the proposed OOV word handling approach will be added to the baseline model, and the translation performance will be studied.

During the testing time, the vocabulary of the proposed model is expanded dynamically. To do this, we first create a vocabulary of all the words in the source language sentences. For each word in the vocabulary, we use pre-trained embedding model from Bojanowski et al. (2017) to generate vector representation. In this model, each word w is represented as bag of character n-grams. For example, the word *where* will be represented using character n-grams <*wh, whe, her, ere, re*> and the special sequence <*where*>. The vector representation of the word w is the sum of vector

[4] www.computecanada.ca

representations of its n-grams. Mathematically, this can be defined as follows:

$$s(w, c) = \sum_{g \in G_w} z_g^T v_c. \tag{6.1}$$

where w is the input word and c is the context word. z_g and v_c are the vector representations for each n-gram and context word respectively. G_w is the set of all n-grams appearing in the word w. Then, the embedding matrix of the NMT system is concatenated with the vector representations the OOV words obtained using the above model.

6.4 German → English Translation

Table 6.2 and Table 6.3 present comparisons of proposed approach with the baseline model from Luong et al. (2015b) on German → English translation. The results demonstrate that the proposed vocabulary expansion significantly improves the translation performance, particularly for out of domain data. The performance gain is noted separately for adding fixed word embedding and for adding expanded vocabulary.

As reported in Table 6.2, for in-domain test data, the performance gains for the proposed approach are in the range of approximately 2-2.8 BLEU points. As a measure of generalization, we also studied the performance on out of domain data. On the test data from WMT'14, the performance gains are approximately 3 BLEU points. We notice that the performance gain is higher in out of domain data, making this approach very suitable for low resource languages.

System	**BLEU score on IWSLT test data**		
	tst2010	tst2011	tst2012
Global Attention Model	27.09 ± 0.96	30.32 ± 0.97	27.11 ± 0.91
Global Attention Model + pre-trained word embeddings	27.40 ± 0.95	31.25 ± 0.99	26.80 ± 0.89
Global Attention Model + pre-trained word embeddings + expanded vocabulary	**29.04 ± 0.90**	**33.18 ± 1.03**	**28.23 ± 0.88**

Table 6.2: German → English Translation performance on **in-domain test data** from IWSLT'14 with 95% confidence interval

Two possible explanations for the huge performance gap between in-domain data and out of domain data are:

- The model was trained in spoken language domain from IWSLT. However, the test data from WMT is news data (written language). Written languages are generally more complicated than spoken languages.
- Out of Vocabulary words (both source and target) are higher in WMT test data in comparision to IWSLT test data.

System	**BLEU on WMT test data**	
	news2012	news2013
Global Attention Model	10.85 ± 0.37	13.09 ± 0.43
Global Attention Model + pre-trained word embeddings	12.75 ± 0.44	14.67 ± 0.44
Global Attention Model + pre-trained word embeddings + expanded vocabulary	**13.77 ± 0.45**	**16.01 ± 0.5**

Table 6.3: German → English Translation performance on **Out of domain test data** from WMT'14 with 95% confidence interval

It has already been shown in the literature that using pre-trained word embedding helps in faster convergence and improved performance (Garcia et al., 2015; Delbrouck et al., 2017). This is shown in the small increase of the BLEU score in the second column of the Table 6.2 and Table 6.3. By fixing the word representations, we retain the linguistics properties captured in the word embedding model. This helps in translating OOV words.

In the *Global attention model* and *Global attention model + fixed word embeddings*, the vocabulary size of the network is fixed at 50,000. When using the expanded vocabulary model, the vector representations for all the unknown words are generated using fastText (Bojanowski et al., 2017). Because of this, the out of vocabulary words which are semantically similar or morphologically related to an in-vocabulary word gets translated correctly. In the literature, we found only Bahar et al. (2017) reported performance on the same test data as ours. They reported a performance of 37.3 BLEU points on tst2010. The difference in performance of their model and our model can be attributed to the size of training data. They used 2.1M parallel sentence pairs for training while we used only 160k.

6.5 Turkish → English Translation

To study the suitability of the approach to a morphologically more complex but low resource language, we ran the experiments on Turkish → English translation. Table 6.4 presents the performance of the proposed approach and the baseline attention model. The results demonstrate that the translation performance significantly improved compared to the baseline attention model. We report performance only on in-domain data from IWSLT'14. To the best of my knowledge, no other test dataset is available to test the out of domain performance.

Similar to German → English, when we use pre-trained word embeddings alone, the performance improves by 1.0 - 1.5 BLEU points. The source vocabulary expansion technique gives another 1.2 BLEU points improvement. Overall, we achieve performance gain of 1.6 - 2.3 BLEU points by expanding the vocabulary of the source language.

System	BLEU		
	tst2010	tst2011	tst2012
Global Attention Model	15.60 ± 0.72	15.79 ± 0.83	16.02 ± 0.73
Global Attention Model + pre-trained word embeddings	16.16 ± 0.72	17.11 ± 0.86	17.27 ± 0.75
Global Attention Model + pre-trained word embeddings + expanded	**17.21 ± 0.75**	**18.31 ± 0.90**	**18.41 ± 0.75**

Table 6.4: BLEU Score: Turkish → English on IWSLT'14 data

The difference in performance between German → English and Turkish → English can be attributed to the low resource availability of Turkish.

- The source vocabulary size for German is 110,000 whereas the source vocabulary size for Turkish is 160,000. This makes it harder for the baseline model to capture learn the input-output mapping.

- When compared with German → English, the low performance of *baseline + pre-trained word vector* and *expanded vocabulary* model can be attributed to the lower quality of the pre-trained word vector for Turkish. The word vector for German was trained on 2.2 million Wikipedia articles whereas the Turkish word vectors were trained on only 100,000 Wikipedia articles.

Transformer model

This vocabulary expansion technique is independent of any underlying architecture. We can use this technique for any word based NMT systems. To study the effectiveness to the vocabulary expansion technique using a different word based NMT system, we ran the same experiment on the transformer model proposed in Vaswani et al. (2017). For all the transformer model, we used the same parameters as the original paper. The word vector and RNN size was set to 512 with 6 layers. Adam optimizer with a learning rate of 2 and learning rate decay was used. The result of the experiments are shown in Table 6.5.

System	**BLEU**		
	tst2010	tst2011	tst2012
Transformer	12.66 ± 0.66	13.51 ± 0.75	13.27 ± 0.66
Transformer + pre-trained word embeddings	17.03 ± 0.86	18.08 ± 0.86	18.86 ± 0.77
Transformer + pre-trained word embeddings + expanded vocabulary	**18.58 ± 0.79**	**19.76 ± 0.92**	**20.27 ± 0.80**

Table 6.5: BLEU Score: Turkish → English on IWSLT'14 data

Compared to the baseline transformer model, we see a significant increase of approximately 4-5 BLEU points when we used pre-trained word embeddings from fasttext. The vocabulary expansion technique further increases the performance by approximately 1.5 BLEU points.

In the literature, we find that only Sennrich et al. (2016) and Gulcehre et al. (2015) reported results on the same test data as ours. In both the papers, they used twice the amount of training data as we did. Gulcehre et al. (2015) integrated language model training on monolingual data into an NMT system. Sennrich et al. (2016) proposed a technique to improve the performance of NMT by constructing synthetic training data obtained using back-translation. Sennrich et al. (2016) reported a performance of 21.2 and 21.1 BLEU points on *tst2011* and *tst2012* respectively. We found that the performance of our model and Gulcehre et al. are very close in *tst2011* and *tst2012*. But Sennrich et al.'s model outperforms our model only by approximately 1 BLEU points despite being trained on twice the amount of training data.

6.6 Comparison with BPE

In this section, we compare the source vocabulary expansion approach with BPE approach proposed in Sennrich et al. (2015). BPE is particularly useful for morphologically rich language as it operates at subword level and capable of modeling open-vocabulary translation. Table 6.6 presents the performance of both the models when trained on IWSLT'14 dataset.

In in-domain data, BPE approach consistently outperforms our approach by approximately 0.5 - 1 BLEU points. For both the language pairs, BPE encoding reduced

	BLEU				
Dataset	tst2010	tst2011	tst2012	news2012	news2013
German → English					
Global Attention Model + pre-trained word embeddings + expanded vocabulary	29.04	33.18	28.23	**13.77**	**16.01**
Global Attention Model + BPE	**29.78**	**34.23**	**30.17**	4.78	5.32
Turkish → English					
Transformer + pre-trained word embeddings + expanded vocabulary	18.58	19.76	**20.27**	-	-
Transformer + BPE	**19.04**	**20.46**	20.00	-	-

Table 6.6: Comparision with BPE for both the language pairs

the source vocabulary approximately 7k - 8k ad target vocabulary to approximately 5k resulting in faster training time. But for out-of-domain data, BPE performed very poorly in comparison to our model. We suspect that the BPE code learned on IWSLT dataset did not fit the test data from newstest. In Table 6.7, we present some of translations from the baseline models and the vocabulary expanded model.

1	*Source*	sie wuchs zu einer zeit auf , in der konfuzianismus die soziale norm und der lokale mandarin die wichtigste person war.
	Target	she grew up at a time when confucianism was the social norm and the local mandarin was the person who mattered.
	Baseline	it grew up at a time when the social norm and the social norm was the most important person in retirement.
	+ fixed embedding	she grew up at a time when the social norm and the local chinese , the most important person was .
	+ expanded vocab	she grew up at a time in buddhism , the social norm , and the local speaker was the most important person .
2	*Source*	evolution bedeutet nicht zwangsläufig das längste leben zu bevorzugen.
	Target	evolution does not necessarily favor the longest-lived.
	Baseline	evolution doesn't necessarily mean the longest biggest life.
	+ fixed embedding	evolution doesn't necessarily mean the longest life to prefer.
	+ expanded vocab	evolution doesn't necessarily mean to prefer the longest life .

Table 6.7: **Sample translations -** for each example, the source sentence, target translation, baseline translations and vocabulary expanded translation are showed.

Summary

In this chapter, we presented an experimental evaluation of how the quality of machine translation for morphologically rich languages can be improved by expanding source language vocabulary. We achieve a performance gain of 2-3 BLEU points for both German → English and Turkish → English. We also note that we see a larger performance gain on out of domain data as compared to in-domain data. We also compared the performance with another popular approach for morphologically rich languages. Our model is competitive, but doesn't outperform BPE in in-domain data. In out-of-domain data, our model performs better than BPE.

Chapter 7

Conclusion and Future Work

7.1 Summary

Machine translation can aid in overcoming the human language barrier in communication. Current translation systems rely on large amounts of parallel corpora for training. But, most of the languages in the world have only a very small amount data available for training these system. Translating morphologically rich languages is more difficult as compared to morphologically poor languages.

In this project, our goal was to improve the quality of machine translation for morphologically rich, low-resource languages. We presented a vocabulary expansion technique that improved the machine translation from (German and Turkish) → English. We demonstrated that our approach improved the translation performance by approximately 2-3 BLEU points.

In our proposed approach, we use pre-trained, fixed word vectors as our input. By doing so, we separated the problem of handling Out of Vocabulary (OOV) words from NMT systems. Therefore, as long as the word embeddings capture the morphological and semantic information of the words and exhibit regularity in their mapping, the translation systems will be able to use that information to translate OOV words.

Our approach is independent of the underlying architecture of the base NMT system. This technique can be incorporated in any NMT systems that work at word level instead of character or sub-word level.

7.2 Future Directions

In our approach, we used a simple attention based NMT system from Luong et al. (2015b), as the underlying base system. The performance in the translation task is limited by the ability of the base system. In future, more experiments, with varying training data sizes can be done on more sophisticated models like purely attention based architecture or convolution neural network (CNN) can be done. Recently, Gehring et al. (2017) proposed a fully parallelizable CNN architecture for translation. Vaswani et al. (2017) proposed a simple network architecture without using any recurrence or convolution. These two networks can be a very good candidate for a base system.

Another promising direction to explore is the expansion of target vocabulary of NMT systems. If a similar improvement is observed in the translation quality, the training time of the network can be greatly reduced.

Appendix A

List of Abbreviations and Definitions

BLEU	Bilingual Evaluation Understudy
BPE	Byte Pair Encoding
CNN	Convolution Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MT	Machine Translation
NMT	Neural Machine Translation
OOV	Out Of Vocabulary words

RNN Recurrent Neural Network

SMT Statistical Machine Translation

Morpheme The smallest meaningful units in a language.

Word embedding A technique used to map and represent words in a language as vectors of real number, where words with similar meaning have similar representations.

Word vector A vector used to represent a word in the vector space. Also known as *Word representations*

Morphological rich languages Languages that encode large amount of information through morphology. They have a very large vocabulary size since there can large number of word forms per lexeme.

Bibliography

ALPAC (1966). *Language and Machines: Computers in Translation and Linguistics; a Report*, volume 1416. National Research Council (US). Automatic Language Processing Advisory Committee, National Academies.

Ataman, D., Negri, M., Turchi, M., and Federico, M. (2017). Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Bahar, P., Rosendahl, J., Rossenbach, N., and Ney, H. (2017). The rwth aachen machine translation systems for iwslt 2017. In *Int. Workshop on Spoken Language Translation*, pages 29–34.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *In International Conference on Learning Representations.*

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bhatia, P., Guthrie, R., and Eisenstein, J. (2016). Morphological priors for probabilistic neural word embeddings. *In Proceedings of Empirical Methods for Natural Language Processing (EMNLP).*

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *TACL'17.*

Botha, J. and Blunsom, P. (2014). Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning*, pages 1899–1907.

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

Cettolo, M., Christian, G., and Marcello, F. (2012). Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. (2014). Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam.*

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational linguistics*, 33(2):201–228.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Conference on Empirical Methods in Natural Language Processing.*

Choi, H., Cho, K., and Bengio, Y. (2017). Context-dependent word representation for neural machine translation. *Computer Speech & Language*, 45:149–160.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Delbrouck, J.-B., Dupont, S., and Seddati, O. (2017). Visually grounded word embeddings and richer visual features for improving multimodal neural machine translation. In *Proc. GLU 2017 International Workshop on Grounding Language Understanding*, pages 62–67.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Evans, N. and Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and brain sciences*, 32(5):429–448.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968. Association for Computational Linguistics.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule. Technical report, Columbia Univ New York Dept of Computer Science.

Garcia, E. M., España-Bonet, C., and Màrquez, L. (2015). Document-level machine translation with word vector models. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *CoRR, abs/1503.03535*, 15.

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

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. *In 53rd Annual Meeting of the Association for Computational Linguistics.*

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware neural language models. In *AAAI*, pages 2741–2749.

Koehn, P. (2017). Neural machine translation. *arXiv preprint arXiv:1709.07809*.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Luong, M.-T. (2016). *Neural Machine Translation*. PhD thesis, Stanford University.

Luong, M.-T. and Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. *CoRR abs/1604.00788*.

Luong, M.-T., Socher, R., and Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. *CoNLL-2013*, page 104.

Luong, M.-t., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W. (2015a). Addressing the rare word problem in neural machine translation. In *In ACL*. Citeseer.

Luong, T., Pham, H., and Manning, C. D. (2015b). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 133–139. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Och, F. J. (2002). *Statistical machine translation: from single-word models to alignment templates*. PhD thesis, Bibliothek der RWTH Aachen.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Rahman, F. (2017). seq2seq. `https://github.com/farizrahman4u/seq2seq`.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *International Conference on Learning Representations*.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Russell, S. J. and Norvig, P. (2002). Artificial intelligence: a modern approach (international edition).

Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Sennrich, R., Haddow, B., and Birch, A. (2016). 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)*, volume 1, pages 86–96.

Soricut, R. and Och, F. J. (2015). Unsupervised morphology induction using word embeddings. Proceedings of the North American Association for Computational Linguistics Conference (NAACL-2015).

Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

Sun, F., Guo, J., Lan, Y., Xu, J., and Cheng, X. (2016). Inside out: Two jointly predictive models for word representations and phrase representations. In *AAAI*, pages 2821–2827.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Charagram: Embedding words and sentences via character n-grams. *In Proceedings of Empirical Methods for Natural Language Processing (EMNLP).*

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.