STATE-OF-THE-ART METHODS FOR IMPROVING LOW-RESOURCE NEURAL MACHINE TRANSLATION

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF THE JOHNS HOPKINS UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Tanay Agarwal
December 2018

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

(Philipp Koehn) Principal Adviser

Preface

For translating between high-resource languages such as French, Spanish, German, and English, Neural Machine Translation (NMT) performs close to human quality because there is an abundance of parallel data that the models can learn from. However, NMT does not perform so well when it comes to low-resource languages for which parallel data are scarce. Although low-resource NMT has recently become a focus in the field of machine translation research, progress has been slow and there has not been much effort to consolidate well-performing methods. In this thesis, we explore two popular families of methods that have been used to tackle the low-resource problem: transfer learning and data augmentation.

We explore the use of state-of-the-art methods for high-resource NMT that have crossovers and applications in low-resource scenarios. Specifically, we demonstrate the high efficacy of the Transformer architecture, domain adaptation, multilingual training, bidirectional training, and backtranslation for low-resource NMT. Furthermore, we build a system that leverages all of these concepts together to maximize the BLEU score performance on Swahili-English and Tagalog-English data. Additionally, we present an analysis on the rare word translation problem and demonstrate that our systems improve out-of-vocabulary word translation rates.

Contents

Chapter 1

Introduction

1.1 Motivation

Machine translation (MT) — the task of automatically converting a sentence from one language to another without human guidance — is one of the most interesting problems in natural language processing. Although it might seem trivial since humans can easily translate between the languages they speak, it is in fact a very difficult problem for computers. The task has many implications in the overarching field of artificial intelligence, since translating a sentence between languages requires some form of understanding of its underlying meaning. Over the past two decades, we have seen tremendous progress in machine translation. For much of its history, MT has been treated as a statistical problem with data-driven approaches. Statistical Machine Translation (SMT) was effective up to a certain point in the presence of large training corpora. Over the last few years, however, the field has made a sharp turn towards the use of neural networks. Neural MT has proven to be significantly more effective than Statistical MT and is now considered the state-of-the-art approach.

Neural MT uses carefully designed neural networks to perform translation end-to-end. The network is trained using standard machine learning methods and once trained, it should be able to effectively translate any sentence from the source language to target language. Although Neural MT has shown much promise, one of its greatest limitations is the need for large amounts of parallel training data. This requirement is characteristic of neural networks in general, and it is especially limiting in machine translation because of the difficulty of data collection.

For translating between high-resource languages such as French, Spanish, German, and English, NMT performs close to human quality because there is an abundance of parallel data that the models can learn from. However, when it comes to low-resource languages such as Swahili and Tagalog, performance is dismal and often worse than SMT. The problem of low-resource neural machine translation is highly relevant for many reasons. A big application of MT is the translation

of obscure texts into English. These obscure texts are rarely in high resource Romance languages but rather in Urdu, Turkish, etc. Furthermore, being able to translate effectively with lower amounts of training data brings us closer to artificial intelligence, since humans do not require seeing millions of sentences when learning a new language. Improving neural network performance in data-scarce environments also has crossover implications in general neural network theory and other related machine learning applications.

For these reasons, there has been rising interest in low-resource NMT over the last few years. Despite there being much effort dedicated towards this task, breakthroughs have been sparse and rarely combined with other effective methods. In this thesis, we aim to present some state-of-the-art methods that, when combined, are able to far surpass previous achievements in low-resource scenarios.

1.2 Thesis Objectives

The aim of this thesis is to explicitly compare and combine different transfer learning and data augmentation techniques to see which methods are better in low-resource scenarios. We explore combinations of multiple approaches to see how translation quality can be maximized.

Specifically, this thesis explores the use of the state-of-the-art Transformer architecture, which has not yet been demonstrated in academic papers to be effective in low-resource scenarios. We present our experiment results and demonstrate that, with the right hyperparameter settings, the Transformer architecture does indeed offer significant improvements over the widely-used recurrent architecture.

We also explore the use of transfer learning to leverage information from high-resource language pairs in order to improve low-resource translation. In particular, we compare multilingual training and domain adaptation. In both these techniques, the NMT model is trained on low-resource as well as high-resource language pairs.

This thesis also explores the use of bidirectional models that learn to translate both directions between the source and target languages. We demonstrate that learning low-resource translation bidirectionally improves the translation quality in both directions.

Last but not least, we combine the above methods with back-translation, a long-studied data augmentation method used in low-resource scenarios. We present a technique to improve the back-translation quality itself using the above methods, as well as how to use the back-translation to improve the desired low-resource translation task.

Chapter 2

Background

This chapter provides the background needed to properly understand the rest of the thesis.

2.1 Machine Translation Evaluation

As in any other machine learning task, machine translation research is highly dependent on reliable evaluation. It is important to use evaluation techniques that are system-independent in order to be able to compare effects across models. The flexibility of language presents an obvious challenge in this regard.

The major issue in MT evaluation is the variance in correctness. For a single input sentence, there can be several different translations, all of which can be correct. These differences come not only from subtleties in interpretation, but also just different preferences in syntax and grammar [Knight and Marcu, 2004].

The industry-standard evaluation process is simple. Generally, a fixed test set of source language sentences is translated by MT systems into a set of target language sentences. The generated sentences are then assessed in some predetermined manner. In this way, the performance of the MT systems can be measured and compared on a common set of sentences [Leusch, 2005].

2.1.1 Human Evaluation

One of the most effective ways of performing MT evaluation is by using human assessors. Once an MT system generates target language sentences, these can be manually evaluated and scored by humans who judge for different aspects such as diction, fluency, grammar, etc.

The Workshop on Machine Translation (WMT) is one of the conferences that uses human evaluators to compare MT systems. For each source sentence, the evaluators receive the reference translation as well as five generated (candidate) translations. The scoring of the MT systems is

determined based on the evaluators' rankings of these candidate sentences [Bojar, 2011].

Human evaluation is usually not feasible due to its time consuming nature. It is unrealistic for human assessors to evaluate large samples of translated sentences in a timely fashion. For this reason, it is generally desirable to use some form of automatic evaluation.

2.1.2 Automatic Evaluation (BLEU)

Research has shown that certain automatic evaluation techniques exhibit high correlation with human evaluations, which makes them particularly desirable for this task [Zhang, 2004]. One such metric is BLEU (Bilingual Evaluation Understudy), which scores a candidate translation against a reference sentence [Papineni, 2001].

BLEU measures how well the output of an MT system overlaps with a reference sentence using n-gram co-occurrence statistics. The BLEU score of a candidate translation is evaluated by two factors concerning its precision as well as its length. In our case, precision refers to the proportion of correct (matching) n-grams in the candidate. For example, uni-gram precision refers to the number of words from the candidate that appear in the reference, divided by the total number of words in the candidate.

However, the standard n-gram precision is often insufficient in measuring accuracy. Consider the following example:

Candidate: how how how how how.

Reference: how are you today?

In this case, the uni-gram precision is 100%, but it's clear to see that the candidate is not a suitable translation. This issue motivates BLEU, which uses a modified n-gram precision measure that eliminates a word in the reference once it is matched to a word in the candidate. Using this measure, the uni-gram precision for our example becomes 20% because the "how" in the reference only matches with the first "how" in the candidate.

BLEU also includes a length penalty which aims to penalize short candidates. Although this may seem like a one-sided penalty, it is actually effective because long candidates are inherently penalized by the modified n-gram precision. The penalty α is calculated as follows (c is the length of the candidate corpus and r is the length of the reference corpus):

$$\alpha = 1 \text{ if } c > r \tag{2.1}$$

$$\alpha = e^{1-\frac{r}{c}} \text{ if } c \leq r \tag{2.2}$$

The final BLEU score is computed as follows (where p_n is the modified n-gram precision):

$$BLEU = \alpha \exp(\sum_{n=1}^{N} \omega_n \log p_n) \tag{2.3}$$

We use the BLEU metric as our evaluation method for the experiments in this thesis.

2.2 Neural Machine Translation

Deep neural networks are known to eliminate the need for extensive feature engineering as they are able to inherently learn several layers of abstraction through their hidden layers. The design choices for the network architecture play a big role when employing these networks for solving machine learning tasks. Neural networks have recently become crucial in the field of machine translation and research efforts are now almost solely focused on NMT. The focus for NMT is to design an end-to-end trainable model that requires no pre-designed features.

Recurrent neural networks (RNN) have been a staple in NMT, as they are capable of maintaining their hidden states while processing sentences sequentially. This allows them to function as memory states that can model long-term dependencies without any special tampering [Bahdanau and Bengio, 2014]. The use of convolutional neural networks (CNN) for NMT has also seen a rise in the last few years. Facebook's AI Research group showed that fixed-size CNNs can in fact be used to translate variable-size sentences instead of RNNs. Furthermore, they demonstrated higher BLEU scores as well as significantly faster training time for CNNs [Gehring et al., 2017] [Chen et al., 2018]. Recently, Google published a new architecture, the Transformer, that is based solely on attention mechanisms. This architecture has been shown to be superior to all previous state-of-the-art architectures in terms of training time and BLEU performance for high-resource languages [Vaswani, 2017]. The Transformer will be our architecture of choice for this thesis, as we demonstrate that it is superior to RNNs and CNNs for low-resource languages.

Neural machine translation employs what is called an encoder-decoder framework in order to learn feature representations of the data. The encoder's job is to convert the input tokens (encodings of input words) into continuous representations (embeddings). In the case of recurrent architectures, these embeddings are then fed into the RNN units to produce a hidden state representation of the input sentence. The job of the decoder, then, is to convert the hidden state representation into the continuous representations of the target language, which then get converted into the target words. This is achieved in tandem with an attention mechanism that controls the weighting of different positions in the input sentence when determining translations [Bahdanau and Bengio, 2014].

The Transformer architecture uses an encoder-decoder framework that is made up entirely of stacked attention mechanisms and fully connected layers. The encoder is composed of several identical layers, each of which has two sublayers: one multi-head self-attention mechanism, and one position-wise fully-connected feed-forward network. The decoder is also composed of several identical layers, each of which has three sublayers: the first two are the same as those found in the encoder, and the third is another multi-head self-attention layer that performs over the output of the encoder stack. For details regarding the Transformer architecture, refer to the Google publication "Attention

is All You Need" [Vaswani, 2017].

In this thesis, we use the Transformer architecture with hyperparameters suggested by the Sockeye toolkit.

2.3 Out-of-Vocabulary Words

Since machine translation systems use pre-defined mappings from input words to vectors, they are only able to operate on pre-determined closed vocabularies. Generally, vocabulary sizes tend to range from 30,000 to 70,000 words depending on available data and computational resources. This presents a challenge because translation is an open vocabulary problem, and it is very possible for test data to contain words that were not present in the training data. This is highly relevant to neural MT because we want the model to perform and train well end-to-end without the need for explicit handling of out-of-vocabulary (OOV) words [Sennrich et al., 2016b].

The most basic method of dealing with OOV words for NMT was to preprocess the test data and identify the OOV words before running the model. The model would then insert a marker in the outputs instead of attempting to translate the words into the target language. A postprocessor would then replace the markers by looking up the OOV words in a dictionary. This method, although popular at first, was shown to have many complications.

Byte pair encoding (BPE) is the method of choice that has been adopted by the research community to solve this problem. It is possible to split words into subword units that build up the meaning of the entire word. Applying BPE to all the data allows the NMT system to train on subwords rather than whole words, thus enabling it to potentially identify the meanings of OOV words in the test data. We use BPE on all our training and testing data in this thesis [Sennrich et al., 2016b].

Researchers have also looked into potential modifications to NMT decoder architectures in order to improve the modeling of rare word distributions. The task of a decoder is to return a probability distribution on possible word translations, and it has been seen that the functional distributions learned across the training data tend to disproportionately favor common words over rare words. Decoder-side weight normalization is a method that has been shown to combat this issue [Nguyen and Chiang, 2018]. We incorporate weight normalization into our Transformer architecture to improve translation of rare words.

Chapter 3

Transfer Learning

3.1 Multilingual Training

Multilingual NMT has long been a task of interest for the research community. Although the task of translating from one source language to a target language is itself difficult and interesting, it is even more intriguing (and relevant to artificial intelligence) to have a model that can translate between many languages. It is more applicable to how humans understand and use language, as we can obtain meaning and translate between multiple languages with relative ease.

Multilingual NMT seems particularly attractive for low-resource MT because the data sparsity gives rise to the need for having other sources of information that the model can learn from. Training a multilingual model that incorporates a high-resource language provides such opportunities. For example, we can consider Swahili and French as our available source languages and English as our target language. Since the encoder architecture creates a representation of the source data in a latent space, it is possible to include multiple languages on the source side that can contribute towards a shared latent space [Chu, 2017]. Since there is only one target language, the decoder side can be used to translate from the shared space to the target with no modifications. The most intuitive way of facilitating this is to simply throw all the data together and train as per usual [Johnson et al., 2017]. In other words, the training data would be a concatenation of French-English parallel text and Swahili-English parallel texts.

Since the sizes of the low-resource and high-resource corpora are so different, it makes sense to over-sample the low-resource data (in other words, copy every sentence multiple times) to scale it in accordance with the size of the high-resource corpora. In our case, we ensure that the number of Swahili-English sentences in the training data is around the same as the number of French-English sentences.

3.2 Domain Adaptation

Domain adaptation is a new line of research within the NMT community and has been gaining traction over the last few years. Loosely speaking, one can consider the domain of an MT task to be the types of content, field of knowledge, and/or style of publications that the system should be able to translate. Training data is in itself scarce, and even more-so when it must all fall into a particular domain. It is unrealistic to build a well-performing system with such scarce in-domain parallel texts.

In recent years, researchers have experimented with ways of combining in-domain data with out-of-domain data, which is typically much more easily obtained in large quantities. A refinement approach that has shown much promise involves first training the model entirely on the out-of-domain data. Then, the converged parameters can be used to initialize a new copy of the model, which can then be trained entirely on the in-domain data [Chu, 2017]. An important part of this approach is the pre-determination of the number of epochs to train in both phases. Typically, out-of-domain training is limited to a small number of epochs relative to the number of in-domain epochs. This method has been shown to be very effective in practice.

Research in domain adaptation has been limited to high-resource languages. It requires the availability of in-domain as well as large amounts of out-of-domain parallel texts. By definition, these are unavailable for low-resource languages since there is a scarcity of overall data in the first place. However, domain adaptation methods can still prove promising for our purposes. The refinement method described above leverages what the model learns from a large corpus and finds a way to apply that towards the in-domain data. A similar concept can be applied to low-resource languages.

Over the last two years, there have been some attempts at tailoring domain adaptation towards low-resource languages by using a high-resource language pair as the out-of-domain data and considering the low-resource pair to be in-domain. In this form of continued training, researchers have experimented with freezing certain decoder parameters, such as the target embeddings [Zoph et al., 2016]. One can build on top of this simple method by incorporating byte pair encoding in order to exploit morphological similarities between the high-resource and low-resource language pairs [Nguyen and Chiang, 2017]. Further improvements to this approach have been made by incorporate more complex recurrent architectures and by further experimenting with freezing and unfreezing certain embeddings [Feng et al., 2017].

For our purposes, we consider the simplest domain adaptation approach described above and demonstrate its effectiveness with the Transformer architecture, with no modification of parameters between training phases. Consider the source languages Swahili (low-resource) and French (high-resource), for example. We can take Swahili to be in-domain and French to be out-of-domain. Then, the refinement approach can work by first training a parent model on French-English data and then initializing the Swahili-English child model training using exactly all those parameters.

It is not very intuitive why this might work at first glance, but there are a few lines of thought. Since the target language is the same in both scenarios, it's possible that BLEU score gains might come from increased target fluency (from the decoder simply seeing more target data) [Zoph et al., 2016]. It is also possible that the decoder obtains a better understanding of target-side "meaning" of words because of potential similarities in the encoder representations of different source languages [Nguyen and Chiang, 2017]. A justification of this refinement approach is that in low-resource scenarios, we need a stronger prior distribution over the data. By first training on a high-resource pair, we are obtaining initial parameters for the child model that must be more informed than random initialization. It is likely the case that the anchor point learned in the parameter space through the high-resource language contains similarities with the desired low-resource parameters.

The multilingual approach and domain adaptation method seem very similar to each other. In both techniques, we train on high-resource as well as low-resource language pairs in order to improve low-resource performance. The multilingual (with scaled low-resource data) and domain adaptation approaches could theoretically have the same number of epoch steps through each sentence in our corpora. However, the domain adaptation method separates the high-resource and low-resource training, whereas the multilingual approach has a back-and-forth aspect where the model has to continuously switch between language-pairs during training. Intuitively, this suggests a kind of regularization where we are preventing the model from over-fitting to the low-resource pair by routinely switching to the high-resource pair and deterring the parameter trajectory [Johnson et al., 2017]. In this thesis, we present results comparing both these transfer learning techniques.

Chapter 4

Data Augmentation

Data augmentation has long been a topic of interest in the machine learning community. Data is the center point of statistical modeling, and it stands to reason that the more usable data we possess, the better models we can produce. The concept of augmenting data in the context of machine translation is quite difficult because of the complexity of natural language. Regardless, it has been studied extensively for high-resource languages, and in recent years the community has also explored some applications on low-resource languages.

One of the simplest proposed methods in recent years has been the idea of copying target-side monolingual data and simply using it on the source-side as well. Basically, one takes a parallel corpus — for example Swahili-English — and concatenates it with an English-English parallel text created by simply copying the target-side sentences of the Swahili-English corpus. Despite its simplicity, this method has been shown to be moderately effective [Currey et al., 2017].

More complex methods have also been proposed. In particular, one such approach targets the translation of low-frequency rare words by augmenting the training data to contain these words in synthetic contexts. It incorporates language modeling to use rare words in new contexts so as to expand the domain observed by the NMT system [Fadaee et al., 2017].

Like any other family of methods, data augmentation has also been taken to the extremes of ambition by researchers interested in unsupervised learning. For example, one studied method claims to enable NMT for low-resource languages that do not have any parallel corpora at all. They use a combination of transliteration and language modeling to leverage a high-resource language with similarities to the low-resource language in order to generate completely synthetic parallel data [Karakanta1 et al., 2018].

Below, we describe the methods that we will explore in this thesis.

4.1 Bidirectional Translation

As discussed earlier, the task of translating from one source language to one target language is relatively basic and does not completely capture the human experience with language. In particular, it makes much more sense for a model to be able to translate in both directions between two languages. For humans, translating from language A to B and from language B to A are virtually the same difficulty if one speaks both languages. This type of bidirectional NMT has recently become a topic of research in the community.

This bidirectional translation can be enabled without any changes to the model architecture. The simplest way is to just concatenate the original source-target language pair with a flipped version of itself, namely the target-language pair [Niu et al., 2018]. For example, we would concatenate our Swahili-English parallel data with English-Swahili parallel data, and then train the model as usual. In addition to the concatenation, we must also attach a specifier token to the start of every sentence in the new training corpus that indicates the target language. For example, our Swahili-English sentences would be prepended with "<2en>" and our English-Swahili sentences would be prepended with "<2sw>", so that the decoder can learn which language it must output. The test sets must also be prepended accordingly. This approach has been shown to be effective in training a bidirectional model.

The approach described above also addresses the data scarcity issue of low-resource languages. Since parallel texts are expensive to obtain for low-resource languages, we need to find ways of augmenting the training procedure and leveraging information from other data. This was the motivation for the multilingual training discussed earlier, as we enlarged the training corpus by including a high-resource language pair. This bidirectional training achieves the same goal, as we are increasing the training corpus size by concatenating the flipped data direction. Although all the sentences are still the same, the data affects the model training differently because the encoder and decoder are observing new data.

In this thesis, we present results demonstrating the effectiveness of bidirectional models in improving low-resource BLEU scores.

4.2 Backtranslation

A commonly used approach for incorporating monolingual data into training is to backtranslate target-side monolingual data into the source language [Sennrich et al., 2016a]. The synthetic source sentences and authentic target sentences can be paired to create parallel data that can be added to the training corpus.

Extensive research has been done on the efficacy and intricacies of backtranslation in NMT. Specifically, researchers have looked into the actual syntactical and grammatical effects of synthetic training data on output translation quality. One of the important issues when it comes to applying

backtranslation is the specification of the ratio of synthetic to authentic training data. It has been shown that adding more synthetic data keeps increasing BLEU performance, although the law of diminishing returns applies [Poncelas et al., 2018]. So, in this thesis we attempt to use as much synthetic data as is feasible given our computational resources.

The combination of backtranslation and bidirectional training is very intriguing for low-resource NMT. NMT systems trained end-to-end lack a direct avenue for incorporating monolingual data. Backtranslation introduces the significant cost of first building a reverse system. The bidirectional approach allows us to use a single architecture throughout training and costs significantly less than training two separate unidirectional systems [Niu et al., 2018]. The resulting model can be used to backtranslate target monolingual data into the source language, which can then be used to further augment the training corpus for our overall system.

In this thesis, we use bidirectional training to improve the use of backtranslation. We train a bidirectional Swahili-English model, for example, and and use this to backtranslate monolingual English data. The quality of backtranslations generated with the bidirectional model should be better than those generated by a unidirectional English-Swahili model because the bidirectional model has more data to fine-tune its parameters. This property holds for both directions, which is why the use of bidirectional models is so enticing. In this way, one can use an iterative bootstrap approach to keep improving translations: namely, one can train a bidirectional model, then generate backtranslations, and then use this data to train a new bidirectional model, and so on and so forth until BLEU scores taper out to convergence [Niu et al., 2018]. We present results demonstrating the effectiveness of backtranslation on low-resource performance when coupled with bidirectional models.

Chapter 5

Experiments

This section lays out the details and results of all experiments conducted for this thesis. First, we describe the data and tools that were used. Then, we describe the methods tested and experimental results.

5.1 Data, Tools, and Resources

Data

For French-English parallel training data, we use the Europarl 2017 corpus. For Swahili-English and Tagalog-English parallel data, we use a combination of the MATERIAL data sets and crawled sentences from the web. For testing, we use the MATERIAL "analysis" sets. We also use monolingual English data crawled from the web. Training data sizes can be found in the table below (L1 refers to the left language and L2 refers to the right language in each pairing).

Language Pair	Sentences	Words (L1)	Words (L2)
French-English	2,007,723	51,388,643	50,196,035
Swahili-English	210,286	3,994,813	4,492,349
Tagalog-English	201,324	4,113,272	3,820,804
English (Monolingual)	499,402	11,550,197	N/A

Testing data sizes can be found in the table below (L1 refers to the left language and L2 refers to the right language in each pairing).

Language Pair	Sentences	Words (L1)	Words (L2)
Swahili-English	6,733	136,393	163,969
Tagalog-English	7,903	143,585	145,405

Toolkits

We conduct all experiments using the Sockeye NMT toolkit developed by Amazon AWS Labs [Hieber et al., 2018]. It is available for open-source use through Github. The toolkit provides sequence-to-sequence framework with state-of-the-art implementations for recurrent, convolutional, and Transformer architectures.

For preprocessing, we use scripts contained within the Moses Decoder toolkit, which is available for open-source use through Github [Koehn, 2007]. Moses is a SMT system but its preprocessing scripts can be used regardless of model. In addition, we use the BPE-related scripts contained within the Subword-NMT toolkit, available on Github [Sennrich et al., 2016b].

For evaluation, we use detokenized case-sensitive BLEU. The evaluation script can be found in the OpenNMT-py toolkit, available on Github.

Computational Resources

All experiments for this thesis were performed on the Center for Language and Speech Processing machine cluster run by The Johns Hopkins University. All training and testing was performed using one NVIDIA Tesla K40 GPU.

5.2 Methods and Results

Preprocessing

All the raw data outlined in the previous section were preprocessed prior to experimentation. The following steps were used:

First, we tokenize all individual language files (including validation and testing). We use "simple" tokenization for our purposes, which applies the most basic form of separation to words. Specifically, commas, periods, other punctuations, and special characters are detached from preceding words using white space.

Second, we clean all our tokenized training parallel texts on the basis of length. Specifically, we use a length cut-off of 80 tokens and a ratio cut-off of 1-to-9. For example, we go through all French-English parallel sentences and discard those where either the source or target has more than 80 tokens.

Third, we train a truecase model on a dump of all of our cleaned training data (meaning we throw all the French, Swahili, Tagalog, and English cleaned training sentences into one file). We then apply the truecase model to all our language files separately (including validation and testing).

Fourth, we train a BPE model on a dump of all our truecased training data (meaning we throw all the French, Swahili, Tagalog, and English truecased training sentences into one file). We then apply the BPE model to all our language files separately (including validation and testing).

Approaches and Results

In this section, we walk through the specific setup of all the approaches and the results. We use Swahili as our low-resource source language in the explanations. The experiments for Tagalog follow in the exact same manner, and the overall aggregation of all results can be found at the end of the section.

Our first set of experiments focuses on the application of different architectures for low-resource languages. We train and test three different models — recurrent, convolutional, and Transformer — on our Swahili-English data. For the recurrent and convolutional architectures, we use Sockeye default parameters. For the Transformer architecture, we use Sockeye recommended parameters along with weight normalization. The detokenized case-sensitive BLEU scores on the test set are:

	RNN	**CNN**	**Transformer**
sw-en	24.73	24.84	26.59

The Transformer model clearly performs the best, thus demonstrating the efficacy of the state-of-the-art architecture on low-resource languages. For the rest of the experiments, we proceed with the Transformer architecture.

One of the goals of this thesis is to present comparisons between the two transfer learning methods described earlier: multilingual training and domain adaptation. To this end, we run these two methods with our French-English data as the high-resource components. Specifically, we concatenate the French-English corpus with the Swahili-English corpus (oversampled 5x) for the multilingual training (labeled ML in the table below). For domain adaptation (labeled DA), we first train the Transformer architecture on just French-English for 5 epochs, and then continue training on just Swahili-English. We also ensemble the models resulting from the two transfer methods because this should improve translations. The baseline column is the "baseline" relative to the transfer learning methods, meaning it represents the generic method of just training on the low-resource component of the data (in our case, the Swahili-English data).

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	26.59	28.96	29.65	31.32

Clearly, we see that the transfer learning methods significantly improve performance. The domain adaptation method seems to be working slightly better than multilingual training, and ensembling them together produces even more improvements.

Continuing with the theme of transfer learning, we now look towards incorporating additional data. Specifically, we can augment the low-resource component of our data with Tagalog-English. It stands to reason that if the addition of French-English data can benefit Swahili-English performance, then adding some Tagalog-English might also prove beneficial. We run the same experiments as earlier but now the low-resource component is Swahili-English concatenated with Tagalog-English.

This means that the baseline model runs regular training on this data, the "multilingual" model concatenates French-English with Swahili-English and Tagalog-English, and the domain adaptation model first trains on just French-English and then continues training on Swahili-English and Tagalog-English. The results below show that adding Tagalog-English data improves BLEU scores across the board.

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	26.59	28.96	29.65	31.32
sw-en + tl-en	28.13	30.48	29.89	32.08

Next, we explore the effect of bidirectional training on low-resource performance. We do this by letting our low-resource component be a concatenation of Swahili-English and English-Swahili. We then test baseline, multilingual (French-English and Swahili-English and English-Swahili), domain adaptation (first French-English then Swahili-English and English-Swahili), and ensemble models as before. One key addition is that we now test both directions: namely, we translate our Swahili test data as well as our English test data using our models. This is reflected in the "rev" row below (note that the BLEU scores for this row cannot be directly compared to the forward direction scores).

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	26.59	28.96	29.65	31.32
sw-en + tl-en	28.13	30.48	29.89	32.08
sw-en + en-sw	27.93	30.66	29.49	31.77

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en + en-sw (rev)	29.57	31.58	31.32	32.81

The bidirectional models indeed prove to be better than the most vanilla models (only Swahili-English), although they are worse than the models with both Swahili-English and Tagalog-English. So, the natural next exploration is to combine everything we have seen so far, by making the low-resource component a concatenation of Swahili-English, Tagalog-English, and English-Swahili. This will allow us to train a bidirectional model while also leveraging all the data in our possession. The results are below, following the same format as described earlier.

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	26.59	28.96	29.65	31.32
sw-en + tl-en	28.13	30.48	29.89	32.08
sw-en + en-sw	27.93	30.66	29.49	31.77
sw-en + en-sw + tl-en	29.55	31.18	30.30	32.13

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en + en-sw (rev)	29.57	31.58	31.32	32.81
sw-en + en-sw + tl-en (rev)	30.96	31.58	31.23	32.58

Clearly, using all the data at our disposal and training a bidirectional model at the same time produces the best results so far. The final component of the experiments is the incorporation of backtranslation. Using the bidirectional model with the best reverse direction BLEU, we backtranslate our monolingual English data to produce synthetic Swahili-English parallel data. Adding this synthetic data to the low-resource component of our experiments produces the following final results:

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	26.59	28.96	29.65	31.32
sw-en + tl-en	28.13	30.48	29.89	32.08
sw-en + en-sw	27.93	30.66	29.49	31.77
sw-en + en-sw + tl-en	29.55	31.18	30.30	32.13
sw-en + en-sw + tl-en + bt	30.72	32.01	31.22	32.92

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en + en-sw (rev)	29.57	31.58	31.32	32.81
sw-en + en-sw + tl-en (rev)	30.96	31.58	31.23	32.58
sw-en + en-sw + tl-en + bt (rev)	30.58	31.55	32.06	32.73

We obtain our best forward direction BLEU score using our hybrid model, which combines all the methods discussed in this thesis — data augmentation through bidirectional training and backtranslation, as well as ensembling of multilingual training and domain adaptation. We also perform all the experiments explained above for Tagalog-English in the exact same manner:

	RNN	**CNN**	**Transformer**
tl-en	27.24	27.63	30.72

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
tl-en	30.72	33.16	34.15	36.02
tl-en + sw-en	33.38	34.93	34.46	36.7
tl-en + en-tl	32.28	34.78	33.07	36.07
tl-en + en-tl + sw-en	33.83	35.55	35.01	36.35
tl-en + en-tl + sw-en + bt	34.77	35.31	34.91	37.18

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
tl-en + en-tl (rev)	25.73	26.12	26.65	27.57
tl-en + en-tl + sw-en (rev)	26.42	26.37	27.25	27.67
tl-en + en-tl + sw-en + bt (rev)	27.15	27.07	27.18	27.90

5.3 Rare Word Analysis

Neural MT is a difficult problem in low-resource settings for many reasons. Getting good BLEU performance using standard techniques is made even more difficult by out-of-vocabulary and rare words in the test set. In this section, we demonstrate that the combination of all the methods discussed in this thesis has a significant positive impact on OOV word translation.

To measure the impact on OOV word translation, we take the approach of directly checking whether OOV words were translated correctly in the test set. We do this by first identifying all the OOV words in our low-resource corpora. That is, we go through our Swahili-English parallel texts and identify all source words that exist in the test set but not in the training set. Then we train alignment models on all the Swahili-English data (training and test sets together) to identify the correct translations of the OOV words according to the reference. Then, we go through our models' generated translations and check if the correct OOV word translation occurs anywhere in the corresponding sentences. We record the percentage of OOV words that are translated correctly by each model. Although it is a relatively naive approach, this measure will show us whether the models are able to at least generate the correct word translation regardless of fluency.

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
sw-en	38.8	42.4	41.9	44.9
sw-en + tl-en	41.1	44.0	42.8	45.2
sw-en + en-sw	42.1	43.7	42.7	45.4
sw-en + en-sw + tl-en	43.3	45.5	44.5	46.1
sw-en + en-sw + tl-en + bt	46.2	47.3	46.4	48.0

	Baseline	**ML (fr)**	**DA (fr)**	**Ensemble**
tl-en	46.2	52.1	51.0	56.2
tl-en + sw-en	52.8	54.7	53.6	56.2
tl-en + en-tl	47.5	52.0	49.2	54.0
tl-en + en-tl + sw-en	53.9	55.3	53.7	56.2
tl-en + en-tl + sw-en + bt	56.1	56.1	56.1	57.1

We can see that the rate of OOV translation is pretty well correlated with our BLEU scores. Our transfer learning and data augmentation methods indeed improve OOV translation rates, and the best rates are obtained by our final combination systems.

Chapter 6

Conclusion

This thesis focused on methods for improving low-resource neural machine translation. We discussed the background of automatic evaluation as well as the various architectures used in high-resource NMT. We also described various transfer learning and data augmentation methods that have been proven to be effective in low-resource scenarios. Of these methods, we experimented with multilingual training, domain adaptation, bidirectional training, and backtranslation, incorporating them all in a pipeline system for low-resource NMT.

Based on our experimental results, we see that our transfer learning and data augmentation significantly improves BLEU scores. When used individually, each method provides benefits over the baseline low-resource scores. The improvements are most significant, however, when the approaches are all combined. Using our combined approach, we are able to leverage all our low-resource data as well as our high-resource and monolingual texts. Furthermore, we demonstrated the positive effects of our approach on rare word translation rates. It is possible (and likely) that improved rare word translation contributes to our higher BLEU scores.

Standard methods tend to work poorly for low-resource NMT because there are not enough data to properly fit the large neural network architectures. In one way or another, all our methods add more data to our baseline training corpus. Despite the augmentations being in different languages and with unorthodox source-target configurations, the additional data improves scores without fail. There are several possible explanations for this. One potential reason is that the increased English data simply leads to more fluency on the decoder-side, and our source-side modifications are not making much of an impact. Another possible explanation is that our approaches are simply providing regularization to the models, as the original low-resource corpus is so small that it tends to induce over-fitting. The most desirable explanation for our success is that our additional data improves performance because the models are in fact able to leverage cross-lingual and bidirectional similarities just like humans. This is corroborated by the OOV translation results in the thesis. Future work in this area could focus on determining the accuracy of these explanations and investigating the true

cause of improved performance in low-resource NMT. Understanding these concepts would lead the community to a much better understand of machine translation as well as the functioning of neural networks and artificial intelligence in general.

Bibliography

[Bahdanau and Bengio, 2014] Bahdanau, D. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

[Bojar, 2011] Bojar, O. (2011). A grain of salt for the wmt manual evaluation.

[Chen et al., 2018] Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Schuster, M., and Chen, Z. (2018). The best of both worlds: Combining recent advances in neural machine translation.

[Chu, 2017] Chu, C. (2017). An empirical comparison of simple domain adaptation methods for neural machine translation.

[Currey et al., 2017] Currey, A., Barone, A. V. M., and Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation.

[Fadaee et al., 2017] Fadaee, M., Bisazza, A., and Monz, C. (2017). Data augmentation for low-resource neural machine translation.

[Feng et al., 2017] Feng, T., Li, M., and Chen, L. (2017). Low-resource neural machine translation with transfer learning.

[Gehring et al., 2017] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

[Hieber et al., 2018] Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A., and Post, M. (2018). Sockeye: A toolkit for neural machine translation.

[Johnson et al., 2017] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., and Thorat, N. (2017). GoogleâĂŹs multilingual neural machine translation system: Enabling zero-shot translation.

[Karakanta1 et al., 2018] Karakanta1, A., Dehdari, J., and van Genabith, J. (2018). Neural machine translation for low-resource languages without parallel corpora.

[Knight and Marcu, 2004] Knight, K. and Marcu, D. (2004). Machine translation in the year 2004.

[Koehn, 2007] Koehn, P. (2007). Moses: Open source toolkit for statistical machine translation.

[Leusch, 2005] Leusch, G. (2005). Evaluation measures in machine translation.

[Nguyen and Chiang, 2017] Nguyen, T. Q. and Chiang, D. (2017). Transfer-learning across low-resource, related languages for neural machine translation.

[Nguyen and Chiang, 2018] Nguyen, T. Q. and Chiang, D. (2018). Improving lexical choice in neural machine translation.

[Niu et al., 2018] Niu, X., Denkowski, M., and Carpuat, M. (2018). Bi-directional neural machine translation with synthetic parallel data.

[Papineni, 2001] Papineni, K. (2001). Bleu: A method for automatic evaluation of machine translation.

[Poncelas et al., 2018] Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G. M., and Passban, P. (2018). Investigating backtranslation in neural machine translation.

[Sennrich et al., 2016a] Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data.

[Sennrich et al., 2016b] Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units.

[Vaswani, 2017] Vaswani, A. (2017). Attention is all you need.

[Zhang, 2004] Zhang, Y. (2004). Interpreting bleu/nist scores: How much improvement do we need to have a better system?

[Zoph et al., 2016] Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer-learning for low-resource neural machine translation.