11-731 Assignment 2 Report

Zehao Guan, Liang Wu, Sean Zhang
{zehaog, liangwu, xiaoronz}@andrew.cmu.edu

Our code is available at https://github.com/zach96guan/11-731_hw2_NMT_further.

1 Introduction

Assignment 2 looks at three low-resource languages from Africa: Afrikaans (af), Northern Sotho (nso), and Xitsonga (ts). We are going to translate from English into these three different languages. Compared to the high-resource languages, these languages do not have a large parallel corpus available, which makes training one NMT system difficult due to the data-hungry nature of neural models. Our team explored different approaches to improve upon a basic transformer model. By using back translation, transfer learning, and hyperparameters tuning, we were able to achieve an obvious BLEU score increase over the baseline for each language.

2 Baseline

2.1 Dataset

The training set consists of 47106 parallel sentences in the en-af pair, 24197 sentences in the en-nso pair, and 187577 sentences in the en-ts pair. For each language, we were also given 3000 parallel sentences as the development set, and 3000 English sentences as the test set.

2.2 Model

The baseline model is a simple transformer model, as described in [1]. It has 4 layers, with a 4-head attention in each layer. It uses an embedding size of 512 and a hidden size of 512. Some other important hyperparameters include a learning rate of 4×10^{-2} and a dropout rate of 0.1.

3 Approaches

After surveying several relevant papers on low-resource translation, we decided on the following **three** directions for improvement of the baseline model.

3.1 Data Cleaning and Augmentation

To augment the training dataset, we leverage some of the monolingual data available online. We obtained the Wikipedia database dumps for all three languages from Wikimedia Downloads. Each dump is an `.xml` file that contains all articles in that wikipedia as well as metadata.

We first applied [3] to extract the texts from the `.xml` files. The extractor removes the metadata, as well as the formatting, urls, and citations from the articles.

Then we apply our own cleaning algorithm, which uses regex to further clean up the texts. We remove empty lines and remaining html formatting lines. More importantly, we removed all symbols that are not part of the alphabet, numbers, or punctuation marks. We note that each language has a sightly different alphabet set. We then run `sentencepiece` on the text files. This gives us 1142591 lines of *af* texts, 38582 lines of *nso* texts, and 5387 lines of *ts* texts.

After that, we apply back translation [5] to the monolingual data: using the baseline transformer model, we translate the clean Wikipedia text files back into English. We then augment the original training set with the new parallel sentences.

We also notice two problems with this approach. The first one is that decoding English sentences takes a relatively long time, and we do not have time to fully back translate all sentences into English, especially for the Afrikaans Wikipedia texts. We also observe that Wikipedia texts are not entirely similar to the type of texts in the training set and the development set. These observations motivate us to consider cleaning the Wikipedia texts so that we only choose the sentences similar to the original data set.

We use the idea outlined in [2], which was also mentioned in Marcin Junczys-Dowmunt's lecture. The idea is: we train a language model on an in-domain corpus I, in this case it is our original training sets. We train a second language model on a non-domain specific corpus N, in this case it is the Wikipedia texts. Define $H_I(s), H_N(s)$ be the per-word cross-entropy of sentence s according to the language models on I, N, respectively.

We then rank all sentences in N according to the score

$$H_I(s) - H_N(s).$$

We select only the sentences with low scores. In this case, we choose a threshold of 2. This gives us 4532 out of 1142591 sentences from *af* Wikipedia, 903 out of 38582 sentences from *nso* Wikipedia, and 2479 out of 5387 sentences from the *ts* Wikipedia. We then back translate these sentences and add them to the training set.

For training the language models, we use a recurrent neural network with LSTM based on the model in Assignment 1. Instead of having an encoder and a decoder, we only have an encoder. We feed in the words one by one, and we compute the loss with respect to the next word in the sentences. This is implemented in `wiki/languageModel.py`.

3.2 Transfer Learning

Recent research has shown the positive result of using data from high-resource languages to improve low-resource translation. We try to follow the practice in [9] by using data from a similar language to train along with the original parallel data. We are different from [9] in that in our case it is one-to-many translation instead of many-to-one. We hope that by sharing the training data, we could improve the result on the low-resource pair.

We found that Dutch is closely related to Afrikaans, and they are even mutually intelligible. So we start our experiment with this pair. The English-Dutch parallel data is obtained from the

European Parliament Proceedings Parallel Corpus 1996-2011. To simulate the setting in [9], we took 100k sentences from the dataset. Like [8], we prepend a symbol like "⟨2af⟩" to indicate the language we want to translate into. We concatenate the two datasets. To compensate for the difference in the amount of data between the English-Dutch pair and the English-Afrikaans pair, we oversample the latter dataset, and generate a vocabulary size of 12,000 using BPE. Oversampling is a practice suggested by [8], where they claim that oversampling can benefit the smaller dataset.

However, the are not any languages similar to the other two we have, so we try the transfer learning approach by concatenating the Xitsonga and Northern Sotho datasets, because they are both Bantu languages, and then transfer the model to Northern Sotho dataset. However, since the vocabularies are less similar between these two languages, we use BPE with vocabulary size of 16,000.

3.3 Hyperparameters Tuning

As stated in the papers [10], low-resource NMT system is very sensitive to the hyperparameters such as dropout rate, BPE vocab size, batch size, learning rate. A systematic way of tuning hyperparameters can greatly improve the performance of translation on low-resource settings. When tuning these parameters, we should also consider the size of dataset, which is significant for determining the tuning direction.

4 Results and Analysis

4.1 Baseline

After running the given baseline model, we obtained the following BLEU scores.

	en-af	en-nso	en-ts
dev	30.96	17.05	32.50

4.2 Our Experiment

After a series of experiments, we obtained the best BLEU scores as follows.

	en-af	en-nso	en-ts
dev	34.59	17.79	37.05

4.3 Analysis

4.3.1 Data Augmentation

For back translation using the baseline transformer model, we obtained the following BLEU scores on the development set.

	af-en	nso-en	ts-en
dev	27.63	16.25	25.57

We compare the BLEU scores for the baseline, using simple data augmentation and using data augmentation with cleaning.

We have the following table on the dataset size of the three languages.

	af-en	nso-en	ts-en
Original Training	47106	24197	187577
Wikipedia Text	1142591	38582	5387
Cleaned Wikipedia	4532	903	2479

We state and analyze the results by looking at each language separately.

For Afrikaans, due to the size of the Afrikaans Wikipedia, we did not perform back translation on the entire Wikipedia text file. However, the amount of clean Wikipedia sentences is small, with a size only 1/10-th of the original training set. Hence, data augmentation with cleaning did not have a large effect on the BLEU score, and the new score is within 1 standard deviation of the provided baseline score.

For Northern Sotho, we note a significant decrease in BLEU score when the entire Wikipedia set is added. Similar to the Afrikaans case, when only the clean texts are added, the BLEU score did not change much from the baseline.

For Xitsonga, we observe that the entire Wikipedia text file is small relative to the original training set, so augmenting data did not have a large influence on the overall dataset.

We further analyze possible reasons for the lack of improvement. One thing to note is the small size of the clean Wikipedia set. The size of the added data is much smaller than the original data in most cases, and it's natural to believe that the new data did not lead to a significant improvement. Perhaps the threshold of 2 in cross-entropy difference can be further increased, but upon inspecting some of the sentences with cross entropy difference < 2, we see that they are already rather different from the original training set. Some of them are in English, others are one-word sentences, a string of numbers, or a url. For example, we included the following sentences.

Northern Sotho	This village surrounded by agincourt and rolle together with Thulamahashe, under Bohlabela district, Mpumalanga province, South Africa.

Xitsonga	17, No.

Another observation we make is on the quality of the back translation. We note that despite a relatively high BLEU score in the development set, the back-translated Wikipedia texts suffer from various mistakes common to neural MT systems. For example, repeating words is very common. An example translation is shown in the table below.

Northern Sotho	Go tloga mowe aya Anglo Alpha Cement Rooderpoort, morago aya PSG Services - Johannesburg
Model	It is precluded to communicate to the cross-ordamination - after consulting on the ground - through interpreter - I'm concerned - I't go to cooperate - I't go to fetch to fetch to Johannesburg - I't go to fetch to fetch to fetch ... (101 words).

We conclude that using back translation on Wikipedia texts to augment the training set did not bring significant improvements to the overall BLEU score. The two possible reasons are (1) the Wikipedia texts are inherently different from the original training set, and not many sentences are left after performing data cleaning, and (2) the low quality of the back translation means that the new dataset does not improve the translation by a significant amount.

4.3.2 Transfer Learning

On the Dutch and Afrikaans training set, we first train the baseline model using the over-sampled, concatenated data, and get a BLEU score of 28.73 after 50 epochs. Then we fine-tune the model using the English-Afrikaans data only, and achieved a BLEU score of 34.59, a 3.63 improvement over the baseline. This shows that the added training samples from Dutch-English dataset help with the translation to Afrikaans.

The result for Afrikaans can be explained by the fact that Dutch and Afrikaans are similar. However, Xitsonga and Northern Sotho are not that similar. It turns out that the results are worse than the baseline. We suspect the reasons might be 1) that the vocabularies of the two languages are too different, so the model is having a hard time learning the difference given the limited data, and 2) the one-to-many nature of the model. By observing the output, it seems that there are many Xitsonga words mixed in the Northern Sotho output.

4.3.3 Hyperparameters Tuning

Through training model and tuning parameters for a long time, we find that using lower dropout rate (from 0.1 to 0.2), fewer samples per batch, reasonable learning rate scheduler (from 5×10^{-5} to 1×10^{-3}) are beneficial in low-resource settings, by which we improve the BLEU scores of *af* and *ts* languages. Especially using simple network structure (like decreasing number of layers to 3) will improve the BLEU score/perplexity performance of *nso* language translation.

5 Contributions

Zehao Guan worked on back-translation and hyperparameters tuning. Liang Wu worked on transfer learning. Sean Zhang worked on data augmentation via back translation as well as data cleaning using language models.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. *Attention Is All You Need*. In *NIPS*.

[2] Robert C. Moore and William Lewis. 2010. *Intelligent Selection of Language Model Training Data*. In *ACL*.

[3] Giuseppe Attardi. 2015. *WikiExtractor*. GitHub repository, https://github.com/attardi/wikiextractor

[4] Junczys-Dowmunt, Marcin. 2018. *Dual conditional cross-entropy filtering of noisy parallel corpora*.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. *Improving Neural Machine Translation Models with Monolingual Data*. In *ACL*.

[6] Bojanowski, Piotr, et al. 2017. *Enriching word vectors with subword information*. Transactions of the Association for Computational Linguistics 5: 135-146.

[7] Li, Junhui, et al. 2017. *Modeling source syntax for neural machine translation*.

[8] Johnson, Melvin, et al. 2017. *Google's multilingual neural machine translation system: Enabling zero-shot translation*. Transactions of the Association for Computational Linguistics 5: 339-351.

[9] Neubig, Graham, and Junjie Hu. 2018. *Rapid adaptation of neural machine translation to new languages*.

[10] Sennrich, Rico, and Biao Zhang. 2019. *Revisiting Low-Resource Neural Machine Translation: A Case Study*.

[11] Zoph, Barret, et al. 2016. *Transfer learning for low-resource neural machine translation*.