11-731: Machine Translation And Seq2Seq Models

Assignment 2 Report

Suhail Barot (sbarot) & Ayush Pareek (apareek)

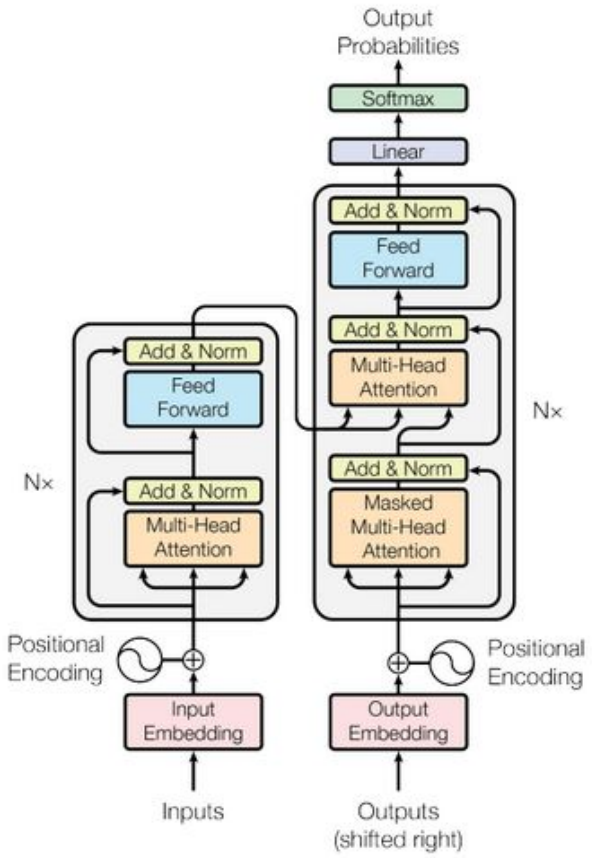
GitHub Repo: https://github.com/ayushoriginal/MT-Assignment-2

Goal

The goal for this assignment was to translate a given corpus of text from English to 3 low resource African languages using a seq2seq model with a focus on maximizing the BLEU score metric on the test set.

Baseline

The baseline that we are using is the implementation of a Transformer [1]. More details can be found in the related paper.



Advanced Model

We hypothesize that the reason why scores of NMT models drop significantly compared to SMT is because of (1) lack of sufficient data and (2) architecture-adaptability to low-resource settings. Hence, we try to focus on improving our system using these two broad ideas.

1. Architecture-centric Methodology- i.e. improving the model architecture to get better results
2. Data-centric Methodology- i.e. augmenting the dataset in various ways to improve the amount of knowledge being fed into the model

1. Architecture-centric Methodology

1) Adding a Lexical Model

There is research indicating that since the learning of Neural Machine Translation takes place in a latent continuous space, it rewards words that seem natural in context but might not be accurate or reflect what was actually said in the source sentence [2,3,4]. Hence, inspired from [5], we add a lexical model on top of a Bi-LSTM based seq2seq model with Attention (from first assignment) and evaluate the scores obtained). In this model, we train a feed-forward network on top of the NMT model such that this model takes in the source embeddings along with the attention weights. To this weighted combination, we add a skip connection to finally compute the probability distribution using softmax. Formally,

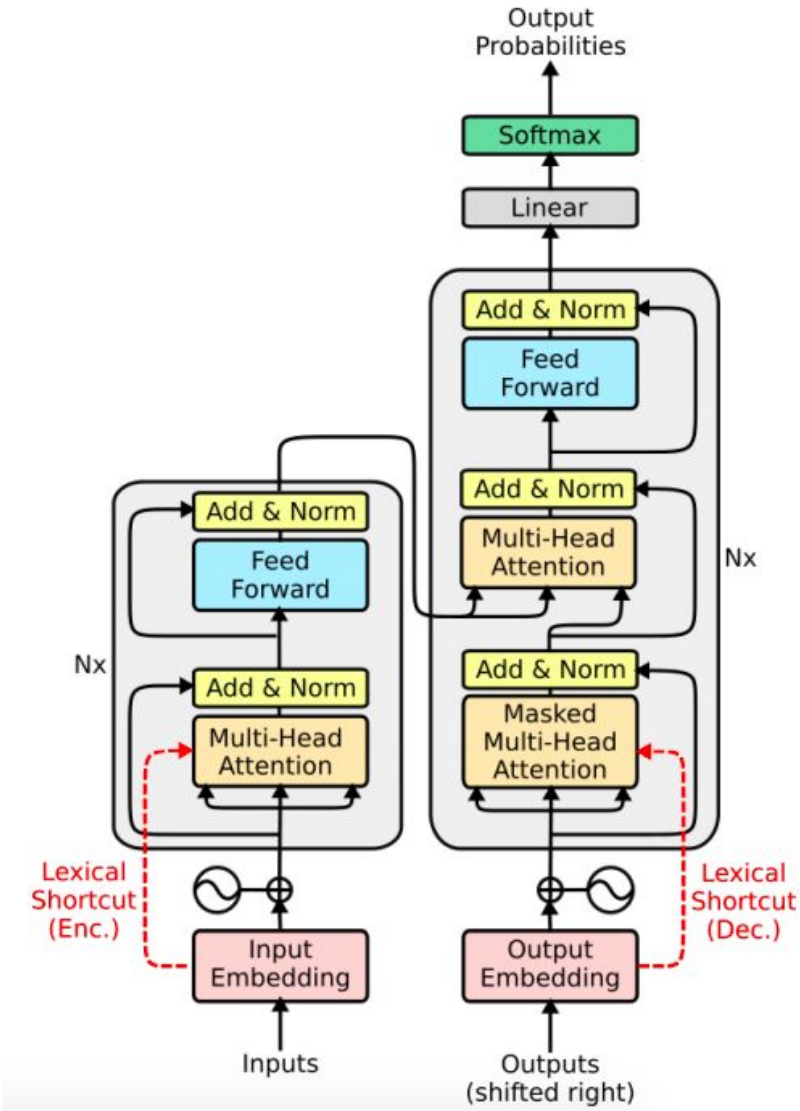
$$f_t^l = \tanh \sum_s a_t(s) f_s$$

$$h_t^l = \tanh(W f_t^l) + f_t^l$$

$$p(y_t | y_{<t}, x) = \text{softmax}(W^o h_t^o + b^o + W^l h_t^l + b^l)$$

2) Transformer with Lexical Connections

In the previous approach, we explored using lexical connections to improve translation quality for low-resource settings. However, we were using it on a attention-based Seq2Seq model rather than the superior Transformer model [1] which are known to be better in lexical disambiguation and capturing long-term dependencies [6]. Hence, we try to extend this idea to transformers by applying the skip connections to the self-attention region of the encoder and decoder as suggested by the implementation of [7]. The overall architecture of this approach is shown in the figure below.



This is similar to using gated units with a skip-connection architecture in order to control the amount of information being passed through the network as suggested by [8]. Similar to the first approach, we use attention to create parameterized connections. In this case, we introduce gated connections between the embedding and the self-attention sub-layer in the encoder and decoder. The embeddings are first projected to the respective latent space by multiplying them with layer-specific weights. We also want to ensure that the network can learn how much importance it wants to give to these lexical features. Hence we add a Gating mechanism similar to [9]. Formally,

$$
\begin{aligned}
K_l^{SC} &= W_l^{K^{SC}} E \\
V_l^{SC} &= W_l^{V^{SC}} E \\
K_l &= W_l^K H_{l-1} \\
V_l &= W_l^V H_{l-1}
\end{aligned}
$$

Where the weights on embeddings (**E**) are used to project them to the appropriate latent space. Also,

$$
\begin{aligned}
r_l^K &= \text{sigmoid}(K_l^{SC} + K_l + b_l^K) \\
r_l^V &= \text{sigmoid}(V_l^{SC} + V_l + b_l^V)
\end{aligned}
$$

$$
\begin{aligned}
K_l' &= r_l^K \odot K_l^{SC} + (1 - r_l^K) \odot K_l \\
V_l' &= r_l^V \odot V_l^{SC} + (1 - r_l^V) \odot V_l
\end{aligned}
$$

Represents the equations inspired by [9] to enable the model to control the significance of lexical features

3) Improving Hyperparameter Configurations

The biggest constraint in hyperparameter optimization is the training time of these large models. To alleviate this issue we take use training upto a limited number of epochs as a heuristic in estimating the performance over a larger number of epochs. We also review the experiments of various researchers to build an intuition of what good hyperparameters should look like for

low-resource setting. Particularly, we make the following observations about hyperparameter tuning on a standard seq2seq model with attention-

1. Number of Layers- For high-resource MT, more number of layers seem to be useful but [5] suggest using smaller and fewer layers for the low-resource setting.
2. Batch Size- For the high-resource setting, a higher batch-size has been suggested [10,11]. We experiment with a lower batch size. The idea is to check if getting more updates is worth it at the cost of noise introduced by smaller batch size.
3. Dropout- We experiment with a higher dropout as suggested by [12]

2. Data-centric Methodology

Using Data from related languages

Several researchers have explored ways to use parallel datasets of related languages to increase the performance by pre-training jointly learn representations [13,14,15,16,17,18,19]

For the given language, we find the following related languages and their corresponding parallel datasets.

Source Language	Related Language	Dataset
Afrikaans	Dutch, Flemish	[20] [21]
Xitsonga	Tswa, Ronga	-
Northern Sotho	Setswana, sheKgalagari and siLozi	[22]

Based on a quick analysis, we found that there was significant vocabulary overlap between the Afrikaans dataset and the dutch dataset from [20]. This overlap was lesser for Northern Sotho and [22]. We use the following methods for improving the performance of NMT using data augmentation from related languages-

Method 1:

1. In this method, we first create a vocabulary of the Low-resource language from its parallel corpora
2. Then we sample those sentences from the High-resource language whose entities match with the vocabulary of the Low-resource language
3. We use the matching sentences of the High-resource language as additional data in our training.

Method 2:

1. In this method, we first create a parallel dictionary between low-resource and high-resource language
2. For Afrikaans, we use [23] and [24] to translate to English and then use Google Cloud API (Translation) [25] to convert english to Dutch. Thus, giving us a Afrikaas-Dutch dictionary of most common words.
3. For Northern Sotho, we crawl through the listing at [26] to convert common words to English and then convert words available in [27] to Setswana
4. Then we replace words in available high-resource language to their semantic equivalent using these dictionaries. This gives us additional parallel data from the High-resource language to augment the Low-resource language dataset.

Results

Quantitative Analysis

On running experiments, we get the following scores.

Quantitative Results

Model	Dataset	SacreBLEU

Transformer Baseline	Afrikaans	32.23
	Xitsonga	31.92
	Northern Sotho	17.29
Seq2Seq + Atten	Afrikaans	27.96
	Xitsonga	26.21
	Northern Sotho	14.82
Seq2Seq + Atten + Hyperparameter tuning	Afrikaans	28.31
	Xitsonga	27.87
	Northern Sotho	15.02
Seq2Seq + Atten + Hyperparameter tuning+ Lexical	Afrikaans	27.88
	Xitsonga	27.14
	Northern Sotho	14.81
Transformer + Lexical	Afrikaans	32.51
	Xitsonga	**34.39**
	Northern Sotho	18.15
Data Aug - Related Lang | (Method 1)	Afrikaans	34.71
	Northern Sotho	18.23
Data Aug - Related Lang | (Method 2)	Afrikaans	**36.04**
	Northern Sotho	**18.4**

Hyperparameter Tuning Details for Seq2Seq Model with attention

We experimented with lower-batch size, low number of layers as well as higher dropout. Our results obtained by performing grid-search on the Seq2Seq model with attention resulted in improvements over our original model

- Batch_Size- 16
- Dropout- 0.3 for dropping words; p = 0.5 for all other dropouts
- LSTM stacked Layers= 2 in both Encoder and Decoder
- Clip Gradient- 4.5
- Uniform Initialization- 0.1
- LR- Decay- 0.55

The lexical model when applied on top of the Seq2Seq model gave mixed results. The Lexical model on top of Transformer did improve the results. Data augmentation methods gave the best results.

Qualitative Analysis

We want to perform some qualitative analysis on the results obtained. Since the output is in low-resource language, it is difficult for us to evaluate the quality due to lack of knowledge of the target language. To alleviate this problem, we backtranslate the output of various systems using Google Translate to english and try to see which output retains the original sentence. Since, Google Translate only works for Afrikaans, we are only able to perform evaluation on that language.

Example 1
Original Sentence (English):

This licence is valid for only six months and can be issued at any driver's licence testing centre.

BackTranslate (Transformer +Lexical):

This licence is valid only and can be issued at any driver.

Backtranslate (Data Augmentation Model):

This license is valid for only six months and can be issued at any driver.

Analysis:

We hypothesize that our data augmentation model was useful in effectively translating words like month (maand) and six (ses) which were available in the vocabulary available in Data augmentation model.

—-

Example 2
Original Sentence (English):

This includes access to finance to initiate BEE deals and ownership, but also providing services and skills to manage and sustain BEE after the deals are made.

BackTranslate (Transformer +Lexical):

This includes access to finance to finance BEE and meal and ownership, as well as providing services and skills to management and sustainability to the threats made.

Backtranslate (Data Augmentation Model):

This includes access to finance to finance BEE and meal and ownership, as well as providing services and skills to management and sustainability for the deals made.

Analysis:

The first translation completely altered the meaning of the original sentence by inferring ‘threat’ as a translation. However, the presence of the word ‘deal’ in our vocabulary effectively preserved the meaning for data augmentation model.

References

[1]Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need.
[2]Arthur, Philip & Neubig, Graham & Nakamura, Satoshi. (2016). Incorporating Discrete Translation Lexicons into Neural Machine Translation. 1557-1567. 10.18653/v1/D16-1162.
[3]Wang, Yang & Wu, Lin. (2017). Multi-View Spectral Clustering via Structured Low-Rank Matrix Factorization. IEEE Transactions on Neural Networks and Learning Systems. PP. 10.1109/TNNLS.2017.2777489.
[4]Wu, Yonghui & Schuster, Mike & Chen, Zhifeng & Le, Quoc & Macherey, Wolfgang & Krikun, Maxim & Cao, Yuan & Gao, Qin & Macherey, Klaus & Klingner, Jeff & Shah, Apurva & Johnson, Melvin & Liu, Xiaobing & Kaiser, ukasz & Gouws, Stephan & Kato, Yoshikiyo & Kudo, Taku & Kazawa, Hideto & Dean, Jeffrey. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.
[5]Nguyen, Toan & Chiang, David. (2017). Improving Lexical Choice in Neural Machine Translation.
[6]Emelin, Denis & Titov, Ivan & Sennrich, Rico. (2019). Widening the Representation Bottleneck in Neural Machine Translation with Lexical Shortcuts.
[7]Emelin, Denis & Titov, Ivan & Sennrich, Rico. (2019). Widening the Representation Bottleneck in Neural Machine Translation with Lexical Shortcuts.
[8] Srivastava, Rupesh & Greff, Klaus & Schmidhuber, Jürgen. (2015). Highway Networks.
[9] Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y.. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.
[10] Murakami, Soichiro & Morishita, Makoto & Hirao, Tsutomu & Nagata, Masaaki. (2019). NTT's Machine Translation Systems for WMT19 Robustness Task.
[11] Masato Neishi∗ and Jin Sakuma∗ and Satoshi Tohda∗ and Shonosuke Ishiwatari,A Bag of Useful Tricks for Practical Neural Machine Translation: Embedding Layer Initialization and Large Batch Size
[12] Gal, Yarin. (2015). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks.
[13] Nakov, Preslav & Liu, Chang & Lu, Wei & Ng, Hwee. (2019). The NUS Statistical Machine Translation System for IWSLT 2009
[14] Zoph, Barret & Le, Quoc. (2016). Neural Architecture Search with Reinforcement Learning.
[15] Chen, Qiqing & Reisser, Julia & Cunsolo, Serena & Kwadijk, Christiaan & Kotterman, M.J.J. & Proietti, Maíra & Slat, Boyan & Ferrari, Francesco & Schwarz, Anna & Levivier,

Aurore & Yin, Daqiang & Hollert, Henner & Koelmans, Albert. (2017). Chen et al 2017. 10.6084/m9.figshare.5632264.v2.
[16] Nguyen, Toan & Chiang, David. (2018). Improving Lexical Choice in Neural Machine Translation. 334-343. 10.18653/v1/N18-1031
[17] Neubig, Graham & Hu, Junjie. (2018). Rapid Adaptation of Neural Machine Translation to New Languages. 875-880. 10.18653/v1/D18-1103
[18]Iyer, Srinivasan & Konstas, Ioannis & Cheung, Alvin & Zettlemoyer, Luke. (2016). Summarizing Source Code using a Neural Attention Model. 2073-2083. 10.18653/v1/P16-1195.
[19] Kocmi, Tom & Bojar, Ondřej. (2018). Trivial Transfer Learning for Low-Resource Neural Machine Translation. 244-252. 10.18653/v1/W18-6325.
[20] http://www.statmt.org/europarl/
[21] http://workshop2017.iwslt.org/
[22]http://rma.nwu.ac.za/SADiLaR_Redirect.html?url=http://hdl.handle.net/20.500.12185/404
[23]http://files.lib.byu.edu/family-history-library/research-outlines/Africa/SouthAfrica.pdf
[24] https://1000mostcommonwords.com/1000-most-common-afrikaans-words/
[25] https://cloud.google.com/translate/docs/intro-to-v3
[26] https://www.bilingo.co.za/sepedi-dictionary-2/
[27] https://www.bilingo.co.za/setswana-dictionary-2/