Expanding the World of Machine Translation: English to Tigrinya - እንቋዕ ብደሓን መጻእኩም

Team members: Anne DeForge, Hannah Abraham, Natsnet Demoz

Abstract

Machine translation (MT) is an important and complex area of study within the world of natural language processing. We focus here on a particularly challenging aspect of NLP, low resource language translation. Our intention is to build an MT model that can translate from English to Tigrinya, which is a language that is currently very underrepresented in both NLP research and in online text. To approach this problem, we experimented with several different transformer models, including leveraging pre-trained transformer models including BERT, M2M100, and T5 in a sequence to sequence architecture to learn the MT task. We compared these results to a baseline LSTM model.

Introduction

The purpose of this project is to create a neural model which can successfully translate from English to Tigrinya. Tigrinya is a language spoken in Eritrea and Ethiopia. One thing that makes translation between English and Tigrinya difficult is that there are different stop characters. There are 7 million Tigrinya speakers worldwide, so translation models and available data are much harder to find compared to English, making this a low resource language. Low resource language implementation of machine translation is important because it can make faster, easier translation available to a wider range of people, decreasing the linguistic divide that exists in technology [1, 2].

Many developments have been made in machine translation, which provide us with the high quality translations available through several different models today. However, these quality translations are a result of training on millions, sometimes billions of data points. For low-resource languages like Tigrinya, there are only a few thousand data points to utilize, so they aren't included in the widely used pre-trained machine translation models. So our goal is to bridge the gap that exists in low resource languages such as Tirgrina.

---

[1] "Transformers for Low-Resource Languages: Is Féidir Linn!" https://arxiv.org/html/2403.01985v1

[2] "Tigrinya Neural Machine Translation with Transfer Learning for Humanitarian Response" https://arxiv.org/abs/2003.11523

Background

Recent research shows that using the transformer encoder/decoder architecture with attention has better potential than potential CNN and RNN architecture to produce quality translations for low-resource languages[1]. In addition, there are multilingual pre-trained transformer models that are trained on over 100 languages. Through transfer learning, these large pretrained transformers can be fine-tuned using the low-resource language data to generate translations.

In "Neural Machine Translation For Low Resource Languages", an mBART model that was pre-trained on 25 languages was fine-tuned with high-resource languages Tamil, Hindi, Vietnamese, and Farsi, and low-resource languages Sinhala, Nepali, Pashto and Khmer. The results showed that with transfer learning, there was an improvement in the BLEU scores for the three out of the four low-resource languages compared to the baseline mBART model[3]. This research motivated our experiments with pre-trained versions of the T5 transformer model to utilize embeddings from other similar languages to improve Tigrinya translation.

In "Beyond English-Centric Multilingual Machine Translation", the M2M100 model is introduced as a multilingual model that improves translation on non-English centric languages. The results show an improvement of over 10 in BLEU score compared to english-centric multilingual models[4]. This paper motivated our experiments with the M2M100 for its potential in translating non-English symbols.

Similar specialized low-resource language experiments have been conducted. For example, in "Towards Better Chinese-Centric Neural Machine Translation For Low-Resource Languages", improvements to mBART and M2M100 were used to pioneer research on low-resource translation for non-English centric languages. Their results for a Chinese-centric translation model surpass baseline models of mBART and M2M100 that were trained on english-centric languages[5]. This encouraged us to move forward with the Tigrinya translation task and use models that include non-English centric languages in pre-training.

In addition to these research papers on low-resource machine translation, we found research specific to Tigrinya. The paper "Tigrinya Neural Machine Translation With Transfer Learning For Humanitarian Response", details an 8-head, 6 layer transformer model that was trained on a mixed dataset of Ge'ez, Amharic, and Tigrinya sentences, and then fine-tuned on only Tigrinya sentences. The results of the experiment show a 1.3 point improvement in BLEU score compared to a baseline trained only on Tigrinya[2]. This experiment specifically for Tigrinya translation influenced our experiment design to focus on the translation potential using multilingual pre-training.

---

[3] "Neural Machine Translation For Low Resource Languages"
https://arxiv.org/pdf/2304.07869

[4] "Beyond English-Centric Multilingual Machine Translation"
https://arxiv.org/pdf/2010.11125

[5] "Towards Better Chinese-Centric Neural Machine Translation For Low-Resource Languages"
https://www.sciencedirect.com/science/article/abs/pii/S0885230823000852

Approach

Data

Online, there is a very limited amount of Tigrinya writing, and it is even more difficult to find a properly labeled parallel corpus. The best available parallel corpus that we were able to identify for training was an English to Tigrinya bible. The numbering of the verses provided a structured way to make sure the translated sentences were aligned to each other in the data. We first performed initial training of our models on a 1000 sentence corpus from the Book of Genesis. We later tested our models on a 5000 sentence corpus. We additionally identified a couple hundred translated sentences provided by the UN. We hoped that this expansion of the dataset would result in better generalization of the model because it could learn a different type of sentence with more modern language and vocabulary.

Tokenization

There are not any pre-trained tokenizations that exist for Tigrinya, as a result, we trained our own tokenization and positional embeddings for the model. We split the data 80/20 for training and validation, the training data was used to create the reference vocabulary with which the tokenization was based upon. We tokenized only on vocabulary from the training split in order to prevent data leakage. One limitation of this is because we were working with a very small dataset, on the unseen data, there were often many unknown tokens to the model.

Vocabulary Size

Another challenge that we faced was that Tigrinya uses a lot more words compared to English. In the 5k corpus, the Tigrinya vocabulary size was approximately 14,000 compared to only 4,500 English words. This results in the model having to learn tens of thousands of tokens to predict the next word in the Tigrinya sentence with only thousands of sentences. We approached this challenge using pre-trained embeddings and techniques to prevent overfitting (see more on overfitting in the model fine-tuning).

Back Translation

We attempted to implement back translation for synthetic data augmentation. The cheapest model to train was the encoder-decoder with no pre-training. We attempted back translation by training two translations models, one that went from English to Tigrinya and the second that went from Tigrinya back to English. With the limited amount of sentences, there wasn't enough data for the model to learn high quality word swaps, so the back translated English sentences that were different from the original source were often grammatically invalid sentences or had other mistakes, like repeated tokens.

Modeling

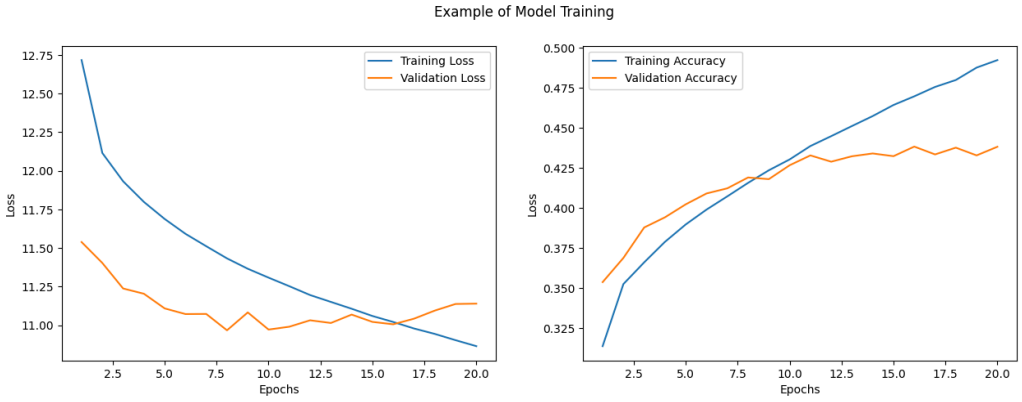
We used a sequence to sequence LSTM model as our baseline to compare several transformer models to. The transformer architectures that we trained included:

- Transformer encoder-decoder with no-pretraining (Trns_np)
- Encoder-decoder with BERT encoder embeddings
- Google's mT5 including English and Amharic embeddings (base and large checkpoints)
- M2M100 with English and Amharic embeddings

We identified Amharic as the best language to fine tune Tigrinya on, as it is a close relative of Tigrinya. There is about 40% lexical similarity between the two languages.[6] Amharic is also spoken in Ethiopia and they share a similar alphabet, this language was in the pre-training corpus of both the M2M100 and T5.

Fine-tuning Hyperparameters

With lots of parameters in the encoder and decoder to learn and not many sentences, one thing that is likely to happen is overfitting. For the model without pre-trained embeddings and embeddings only in the decoder, it was difficult for the model to generalize to the holdout set. To combat this, we implemented a dropout layer in the decoder as well as regularization in order to improve the model performance on unseen data; larger learning rates were also necessary to prevent overfitting. Beyond preventing overfitting, tuning the embedding dimensions and intermediate dimensions did not result in much change in model performance. Below is a model training progression from the model without pre-training, as shown in the plot after 15 epochs, the model begins to overfit.



Using pre-trained Amharic also improved the model's ability to learn more accurate associations between English and Tigrinya sentences without overfitting the training data. We were thus able to tune the batch size and learning rate to improve the performance of these models.

[6] The Similarity and Mutual Intelligibility between Amharic and Tigrigna Varieties https://aclanthology.org/W17-1206.pdf

Results

Below is the results table with the best performance of each model architecture on both dataset sizes:

Model	Dataset Size	BLEU Score
LSTM (baseline)	1k	0.0315
Trns_np	1k	0.0923
Trns_np	5k	0.1501
BERT	1k	0.0980
BERT	5k	0.1374
mT5-base	1k	0.0170
mT5-base	5k	0.1863
mT5-large	1k	0.0499
mT5-large	5k	0.2804
M2M100	1k	0.3000
M2M100	5k	0.5196

Example output from best performing model:

```
English Input:
 You shall wash your clothes on the seventh day, and you shall be clean; and afterward you shall come into the camp.
Tigrinya Translation:
 ኣብታ ሳብዐይቲ መዓልቲ ኽኣ ክዳውንትኹም ሕጸቡ፡ ትነጽሁ ኽኣ። ድሕርቲ ሰፈር ትኣትዉ።
Predicted Translation:
 በታ ሳብዐይቲ መዓልቲ ኽኣ ክዳውንትኩም ሕጸቡ፡ ክትነጽህ ድማ ኢኹም። ድሕርቲ ናብ ሰፈር ኣትዩ።
```

Discussion

We found throughout our experiments that increasing the training set improved the model performance. A similar study on low-resource language MT found that a dataset of at least 16k resulted in BLEU scores of above 0.60.[1] We hypothesize that our model could similarly see this type of performance if we were to be able to expand the dataset.

The transformer model without any pre-trained embeddings did outperform the baseline LSTM, but overall the model performance was quite poor. The embeddings for this model were initialized using the keras token and position embedding, so it only had to learn the sentence patterns of the dataset. The embeddings were likely not capturing any canonical meaning of the dataset and instead just learning the patterns of sentences in the bible. However, it still struggled significantly with this task, as there were many parameters to learn to generate all the tokens accurately and too little data to train all of the transformer embeddings. The fact that the

embeddings with no pre-training were not capturing canonical meaning might also be a reason why adding BERT embeddings to only the encoder was not able to improve model performance. The model was attempting to connect an embedding with meaning to one that didn't have much meaning at all.

The models that utilized pre-trained embeddings in both the encoder and decoder were able to perform significantly better. The best model of the two was M2M100. This model was specifically designed to be used for low-resource languages of the same family group which was probably the reason why it did so well. Using the Amharic pre-trained embeddings in the M2M100 was extremely useful for model performance. One potential reason why the mT5 might have struggled is that by using a corpus of bible sentences, we were training on lots of archaic language. If the pre-training was on more modern data, then it would be much harder for the model to learn on our dataset. Despite these potential problems, both the mT5 and the M2M100 results support the existing research that shows that pre-training on language families can be useful for low-resource language machine translation tasks.

Conclusion

In this report, we demonstrate how the transformer architecture and pre-trained embeddings on similar languages can improve translation results for low-resource languages. Building on current research, we showed that BERT with embeddings, and pre-trained multilingual translation models like M2M100 can be fine tuned to work on languages with limited data like Tigrinya. These results encourage further research into low-resource machine translation so that language barriers can be overcome in travel, information exchange, humanitarian efforts, and connecting with people around the world.

Bibliography

1. Lankford, S. (2024, March 4). Transformers for Low-Resource Languages: Is Féidir Linn!. Retrieved from https://arxiv.org/html/2403.01985v1.
2. Öktem, Et Al. (2020, March 9). Tigrinya Neural Machine Translation with Transfer Learning for Humanitarian Response. Retrieved from https://arxiv.org/abs/2003.11523.
3. Goyle, Et Al. (2023, April 18). Neural Machine Translation For Low Resource Languages. Retrieved from https://arxiv.org/pdf/2304.07869.
4. Fan, Et Al. (2020, October 21). Beyond English-Centric Multilingual Machine Translation. Retrieved from https://arxiv.org/pdf/2010.11125.
5. Li, Et Al. (2023, April 10). Towards Better Chinese-Centric Neural Machine Translation For Low-Resource Languages. Retrieved from https://www.sciencedirect.com/science/article/pii/S0885230823000852**.**
6. Feleke, T. (2017, April 3). The Similarity and Mutual Intelligibility between Amharic and Tigrigna Varieties. Retrieved from https://aclanthology.org/W17-1206.pdf.