Neural Machine Translation: English-Afrikaans

Jean Lucien Randrianantenaina
Applied Mathematics, Machine Learning and Artificial Intelligence
Stellenbosch University

***Abstract*—**

I. Introduction

II. Neural Machine Translation

Neural Machine Translation (NMT) is at the heart of many language translation systems in the current era. We will briefly present the main idea behind this machine learning concept.

Most of these models are based on the encoder-decoder architecture. This kind of architecture is usually introduced as a neural network that learns the identity function, e.g., reproduces its input. However, it can be used in various tasks like image processing, computer vision, and Natural Language Processing (NLP).

In the case of NMT, the architecture learns how to translate a language A to a language B. Figure **??** illustrates a high-level representation of such an architecture.

Usually, we use Recurrent Neural Networks (RNN) to create these models, also known as seq2seq (sequence-to-sequence) models. A particularity of these neurons is to take their output as the next input. The three most well-known RNNs are the Vanilla RNN, Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM). However, these architectures seem to be hard to train and computationally slow due to their recurrence.

One way researchers have found to improve the performance of these architectures is the attention mechanism, which aims to conserve the importance of each output of the encoder when decoding its input. This technique was primarily used with RNNs but was later used with classical neural networks (without recurrence), giving rise to what we know as self-attention. The main idea behind this is to rely only on the output of the layers using some weighting system to produce the next layers or outputs.

Lastly, attention and self-attention yielded the so-called transformer block, which is part of the state-of-the-art in many machine learning models today.

III. Methodology

In this task, we will focus on translating English to Afrikaans using a small dataset from engineering assement corpus from Stellenbosch University.

To achieve that, we will implement from scratch a vanilla RNN, LSTM, GRU, and GRU with attention. Then, we will use the pretrained `opus-mt-en-af` model. The goal is to show if we are able to implement such models and performing comparison as well.

A. Dataset

The provided dataset is very small and may not have much vocabulary. To increase word coverage, we included the English-Afrikaans dataset from the Tatoeba project. However, our main dataset focus is on the engineering domain, and adding this dataset will increase common words only, which is one of the reasons to use a pretrained model in the second part.

To clean the dataset, we tried as much as possible to detect the LaTeX environment and compactify it as single words (removing spaces), add spaces on delimiter characters such as "(),[],{}" to treat them as a single character, finally removing extra spaces. We do not change the case and do not remove special character. This may limit our model, but that will be more close to a real world task (Opter option may be possible for the case, but it will be for a future work).

B. From Scratch Implementation

As mentioned earlier, our from scratch model will have the parameter listyed in the table I. The only things that will differentiate them will be the architecture that we use.

Parameter	Value
Embedding Size	256
Hidden Size	1024
Number of Layers	2

TABLE I: Hyperparameters used in the from-scratch models.

Each the model will be trained at most with 50 epochs, using a Adam as optimiser and the CrossEntropyLoss function. After taining weevaluate these model on the test set, combined with the augmentaion and the provided data set only.

C. Using a Pre-trained Model

Why use a pre-trained model? We can save a lot of time and money by using a pre-trained model, as most of the hard work has already been done. Usually, pre-trained models are trained on a huge amount of data, therefore they contain a lot of knowledge. In this work, we want to translate engineering assessments from English to Afrikaans with a small amount of data. To have a more general model, we use the pre-trained model that already translates English to Afrikaans.

This model was trained on public datasets, which may not contain scientific and technical words, so by retraining it on a new dataset, we perform a domain shifting/adaptaion.

The model is based on the MARIAN model, which is an open source model also, which use transformer block. it use subword units. So to tokenize our data we use the one that is provided with the model.

The data is prepared in the same way as the from scratch model. We train the model with Adam in first palce and AdamW (As it is my first time to see this one) in the secnd place, in order to compare which one give us the best results.

In order to see if our model improved after the fine tunning process, we evaluate on the validation set before and after the trainig.

Concering the loss function, it is provided directly with the model, and we were not able to find a paper or documnetation that explictily show the loss function.

IV. Results and discussion

V. Conclusion

References

[1] Herman Kamper. *NLP817*. https://www.kamperh.com/nlp817/, 2022–2024.

[2] Open Parallel Corpora https://opus.nlpl.eu/

[3] Helsinki NLP https://blogs.helsinki.fi/language-technology/, https://huggingface.co/Helsinki-NLP