Language Identification with Contextual Translation

Alex Shah

12/4/17

1 Abstract

This project's goal is to combine different state of the art neural network architectures to make a translator that detects the input language and translates with contextual awareness. The neural networks required to accomplish this task are a CNN classifier and tensor2tensor's transformer model for translation. Initially, this project sought to use English and Spanish languages, however due to dataset difficulties, training was done with a French dataset. It remains the case that any language with an appropriately processed dataset can be trained with these models.

2 Introduction

Providing a software implementation for translation does not require the entire computation to occur on device, but rather through training ahead of time. In this way, your cell phone does not need a desktop PC's compute power to leverage neural networks and the advances such technologies have made in machine translation. However, training the network does require a large amount of compute power and time.

In order to minimize the amount of time required, we can make training more efficient by training both translation directions simultaneously. By using encoder/decoder as a basis for our model, languages and context are broken down into mathematical representations that the network uses to translate any language to any other. We do not need to train translation from language A to language B and vice versa, but can learn each language independently. By utilizing language agnostic training, we can improve the speed at which we train. We can train any language in the same time, without needing to retrain or cross train various languages when a new language is added. This concept applies to both the classifier and translation.

3 Background

3.1 Language Classification

Classifying a given input is handled by a Convolutional Neural Network (CNN). It is crucial to quickly determine the input language. Identifying the

language is an intermediary step before translation and incorrectly determining the input language will create incorrect translations. Therefore it is important that the classification network is able to quickly but accurately detect the input language. A CNN is accurate with very few epochs of training. In as little as 1 epoch the validation accuracy is over 99% for inputs of 70 characters (K.M. 2016).

3.2 Contextually Aware Translation

Strides have been made using deep learning models for Machine Translation (MT). MT research grows ever more accurate in contextually aware translation. Sequence to Sequence networks (Seq2Seq) are built like an encoder/decoder. The encoded input text is examined to determine the decoded output in the target language. Adding an attention mechanism is used to share information from previous input steps to better decode output. This is more efficient than bidirectionally sharing input and output values and is also more accurate than methods that require less computation such as reversing input strings. Specifically, attention shares memory between the encoder steps and decoder to produce more contextually aware output. The encoder mechanisms in NMT models are used to build a thought vector from a given input. This vector captures context which is then decoded into the target language translation. Furthermore, an attention mechanism coupled with dropout determines how much context is too little or too much. This enables the translation to capture and translate advanced syntactic structures such as gender agreement and other "long range dependencies" without overfit or poor context capture (Luong, 2015).

Neural Machine Translation (NMT) advances even further with Google's tensor2tensor library,. T2T is a specifically derived set of Seq2Seq and RNN based models focusing on accuracy and efficiency of training translation models. Models like Transformer have achieved some of the best accuracies in NMT (Luong, 2015).

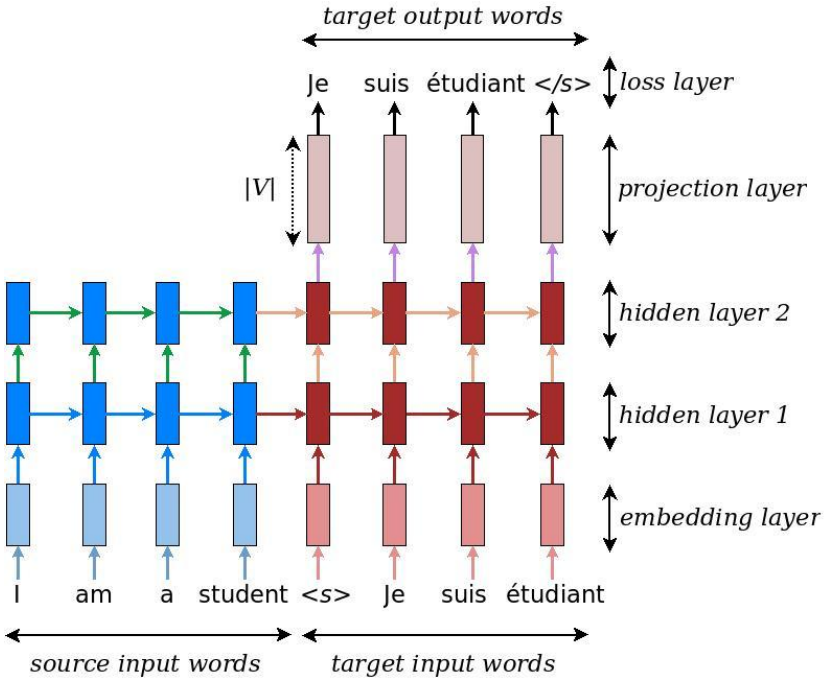

Figure 1: Sequence to Sequence Model (Luong, 2017)

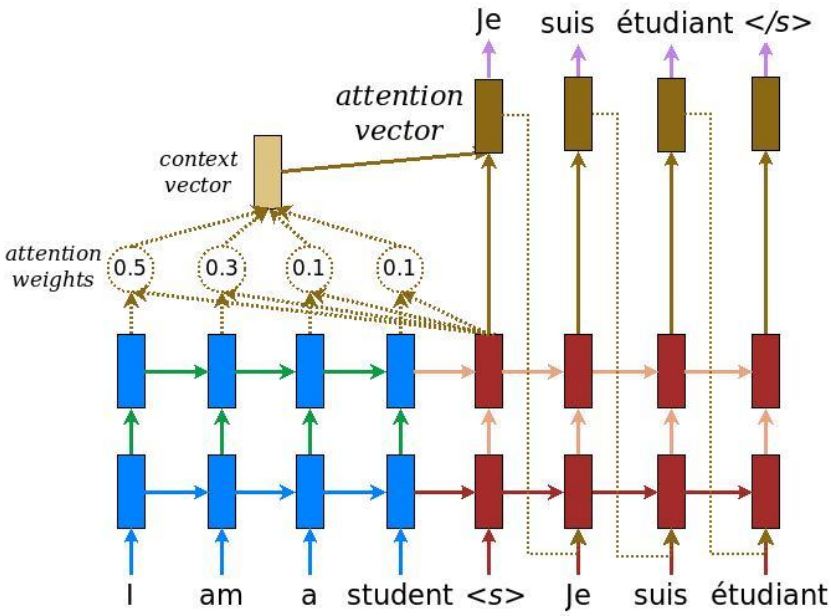

Figure 2: Sequence to Sequence Model with Attention Mechanism (Luong, 2017)

4 Methodology

In this project there are two networks. A Convolutional Neural Network is used as a language classifier, to detect whether a given input is English or French. Next, a tensor2tensor network, specifically using Google's Transformer model, translates the given input to the opposite language. For example a given input is identified as English and translated to French. And vice versa.

In order to improve classifier accuracy, each language's specific characteristics are identified manually ahead of time. For example French has accented characters which allows the CNN to easily identify input characters that will only appear in a certain language (K.M. 2016).

Transformer is the latest from Google's research into efficient and accurate multipurpose models. Transformer is able to work with a range of inputs such as images, text, and audio with a focus on scalability and efficient training. As such, transformer is able to achieve accuracy greater than previous models. In particular, Transformer employs an RNN with attention, and a specifically designed encoder/decoder. An encoder forms a representation of the input language without needing to train translation from one language to another. The decoder translates from representations of languages to human readable output accurately and quickly (Britz, 2017).

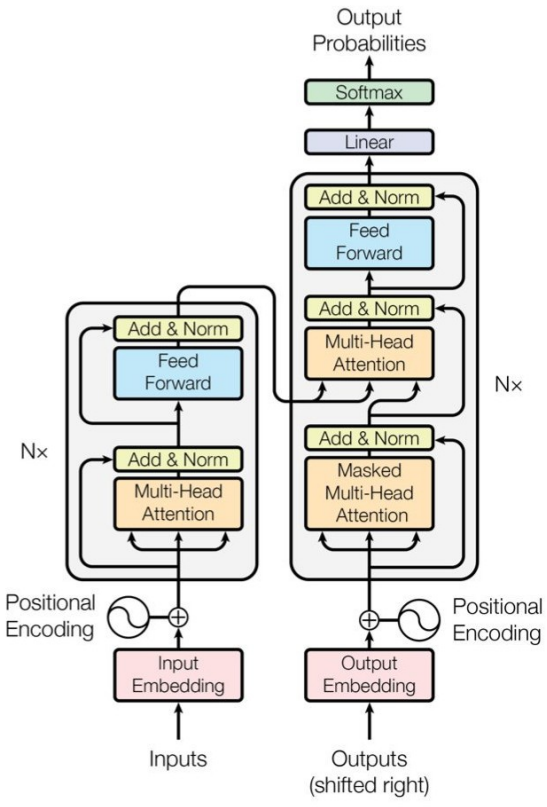

Figure 3: Transformer Model architecture

5 Experiments

Datasets in Spanish were difficult to find, but there are many datasets in other languages. For training, the classifier and translation models were trained with the WMT17-small8k dataset in English and French. In order to train translation, the model needs to study the target language and human corrected translations. It is not necessary to train the model with the input language and human corrected translations of the target language, but rather any human corrected language. In this way a trained model can train on multiple languages without need for datasets to overlap languages, this enables us to add a new language at a later date.

The dataset is WMT's English-French small 8k vocab. This enabled both the translator and classifier to train using the same dataset. The small vocab size was necessary to train faster with a single GPU.

The classifier's Keras Sequential model was able to reach near perfect validation accuracy in the first epoch. The test accuracy after 5 epochs was 0.99, effectively not able to improve from further training.

The translation net takes significantly longer to train. After 5 days, the validation estimated BLEU score was over 50. At the time of writing the estimated validation BLEU score is 51.232. Although the approximate scores showed great promise, training on a single GPU did not complete in time to run actual tests and t2t decoding could not function without reaching the target iterations. Unfortunately, training cannot be foreshortened or resumed. Therefore, while each component of the project proved to be highly accurate during training and validation steps, there was not enough time to train the translation network in time for a functional demonstration or a test version of the final product to be produced. On a single GPU (Nvidia GTX 980 Ti), the estimated time to complete training exceeds one month.

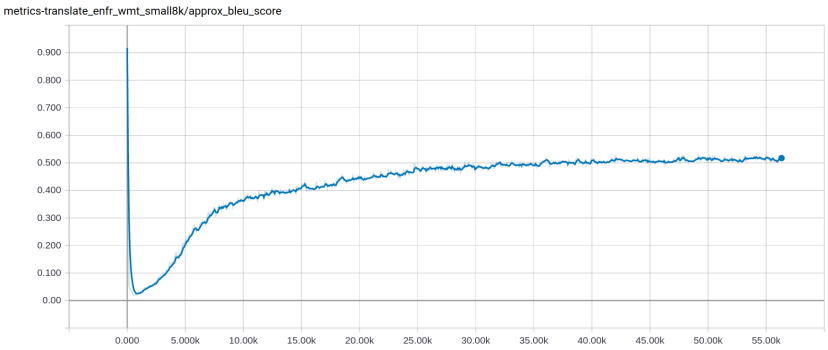

Figure 4: BLEU score after 5 days

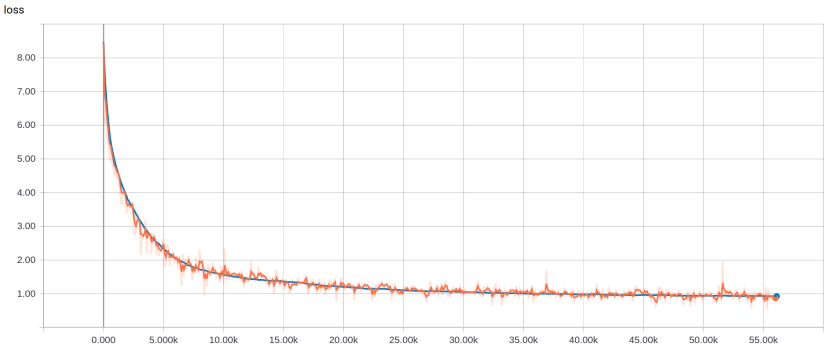

Figure 5: Loss after 5 days

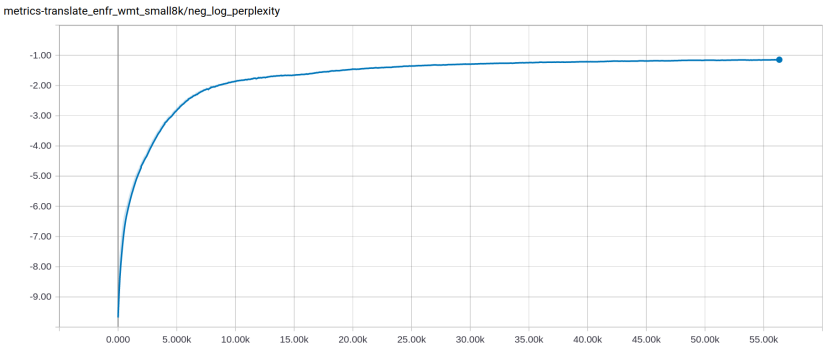

Figure 6: Negative Log Perplexity after 5 days

6 Conclusion

While the intention of this project was to create a finished executable, the research and training data gathered proves that the concept will work, even though training and testing was incomplete. From this project I have learned how to scale and efficiently modify basic neural network architectures, including some not so basic models, to work with particular data and concepts. The classifier works as intended to identify multiple languages. Tthe translation model will work once trained sufficiently. Training Transformer for 250,000 global steps proved to be a mistake, and what progress that was acheived could not be resumed.

7 References

7.1 Further Reading

https://github.com/tensorflow/nmt
https://github.com/google/seq2seq
https://github.com/tensorflow/tensor2tensor
http://www.nltk.org/
https://nlp.stanford.edu/projects/nmt/
https://research.googleblog.com/2017/07/building-your-own-neural-machine.html
https://sites.google.com/site/acl16nmt/
https://github.com/lmthang/thesis
https://google.github.io/seq2seq/data/
http://www.statmt.org/europarl/
https://conferences.unite.un.org/UNCorpus
http://www.statmt.org/wmt17/translation-task.html

7.2 Sources

Britz, Denny, et al. "Massive Exploration of Neural Machine Translation Architectures" Arvix, 2017. (https://arxiv.org/pdf/1703.03906.pdf)

Johnson, Melvin, et al. "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation." Arxiv. 2016. (https://arxiv.org/pdf/1611.04558v1.pdf)

Sutskever, Ilya, et al. "Sequence to Sequence Learning with Neural Networks." NIPS. 2015. (https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf)

Cho, Kyunghyun, et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." ACLweb. 2014. (http://aclweb.org/anthology/D/D14/D14-1179.pdf)

Bahdanau, Dzmitry, et al. "Neural Machine Translation By Jointly Learning To Align and Translate." Arxiv. 2016. (https://arxiv.org/pdf/1409.0473.pdf)

Luong, Minh-Thang, et al. "Effective Approaches to Attention-based Neural Machine Translation." Arxiv. 2015. (https://arxiv.org/pdf/1508.04025.pdf)

Wu, Yonghui, et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." Arvix. 2016. (https://arxiv.org/abs/1609.08144)

Neubig, Graham, et al. "Neural Machine Translation and Sequence-to-sequence Models: A Tutorial." Arvix. 2017. (https://arxiv.org/abs/1703.01619)