Project Proposal: Enhancing Multilingual Machine Translation with Context-Based Transfer Learning

Claire Jung, Arijit Nukala, Christine O'Connor

October 26, 2023

1 Introduction

Neural machine translation (NMT) has revolutionized machine translation, offering incredible performance compared to gold standard statistical translation methods. However, the resources and computational power required to train state-of-the-art models are vast and out of reach for many students and researchers. Transfer learning is a powerful paradigm for training machine learning models which applies pre-trained models to similar or new tasks. Leveraging pre-trained models requires significantly less computational power and training data to achieve performant models in tangential tasks. In the context of machine translation, transfer learning can be used to train models to translate tangential languages to the languages they were originally trained on, or even train models for complex translation tasks such as the translation of scientific or legal texts.

2 Project Summary

We would like to conduct transfer learning for the domain of economic/political texts. We will train off of the pretrained model KR-Bert because this model has existing knowledge that should allow it to perform well on general texts, but its ability to translate the domain-specific texts would likely benefit from extra training within the domain.

3 Project Outline

Our project is roughly based on the methodology and models used in the paper ”A pre-trained BERT for Korean medical natural language processing”. In the paper, the authors analyze the performance of KR-BERT, a BERT model trained on Korean language data. Although KR-BERT displays excellent performance on general Korean langauge translation, its performance in specialized domains such as medicine can be improved, as the paper aimed to do. We plan to extend KR-BERT to domains other than medicine using transfer learning and compare the performance of our resulting model to other NMT models for Korean to English translation. Transfer learning will enable us to train the BERT model, which is complex and resource intensive yet powerful, using minimal GPU resources.

To model the Korean language, we plan to use the KR-BERT model. We will use a dataset of economic/political related Korean-English sentence pairs to train the pre-trained model. We will split this dataset into train, dev, and test, with train used during training, dev used at the end of each iteration, and test used at the end to assess performance.

Our model will take Korean sentences as input ($\vec{k}$), and output English sentence translations ($\vec{e}$). We will assess the performance of the model using the BLEU score through the NLTK package and using the KLUE benchmark. We may also invite native speakers of the language to rate the translations and average their scores (as discussed during lecture). We also plan to compare our model's translations to those produced by existing Korean translation systems NaverNMT, so we will score translations from both of these.

Once we have a pre-trained model, we would like to assess its performance on data outside of our original dataset. To do this, we will implement real-world data scraping practices to parse published news articles on The Kyunghyang Shinmun, and their corresponding linked Korean news articles. These news article pairs tend to have more sentence-to-sentence translations and would ideally work with our evaluation metrics while still providing variation in data for us to test. We plan to use BeautifulSoup and/or Selenium packages to extract this web data.

If time permits, we are also interested in developing a small custom network trained on the in-domain text only to assess the ability of a smaller, less resource-consuming model to translate these texts. This may be similar to the network we implemented in Homework 4, with an encoder and a decoder with attention. We are also interested in analayzing how we can implement potential performance improvements in the model, such as by using Mojo, a langauge built as an extension of python with efficient JIT compilation and an extensive library of low level functions such as SIMD which can drastically improve the speed of matrix computation during inference.