NLP Frameworks for Grammatical Error Correction:
A Comparison of OpenNMT and Fairseq

The goal of grammatical error correction (GEC) is to provide an automatic way to automatically detect and fix mistakes in text. It can be framed as a translation problem, but instead of translating between human languages like English and French, GEC translates between informal or incorrect text to a more formal representation. When viewed as a translation problem, the tools developed for neural machine translation (NMT) become immediately applicable.

Two popular open source NMT frameworks were considered for this project: OpenNMT and Fairseq. Both are very capable and mature systems that would provide significant value to help develop a GEC solution. After carefully evaluation, however, we decided to proceed with Fairseq.

OpenNMT

OpenNMT was developed at Harvard's natural language processing group by PhD candidate Yoon Kim and his advisor Alexander Rush. The open source implementation was announced in December 2016. While OpenNMT focuses on translation, it also provides support for related sequence-to-sequence problems such as text summarization, image-to-text, and speech recognition. While the project was originally developed for the Lua-based Torch machine learning framework, there are also supported versions for PyTorch and TensorFlow.

Fairseq

Fairseq was developed by the Facebook Artificial Intelligence Research (FAIR) team as part of their work to develop a new method of NMT based on convolutional neural networks (CNN). Unlike traditional recurrent neural networks (RNN), the CNN approach tries to exploit hierarchical information in text. It can also better exploit parallelism in GPUs for more efficient training and inferencing. The source code was release in May of 2017. Like OpenNMT, Fairseq was originally based Torch, but new development has shifted toward a reimplementation in PyTorch. In addition to the NMT models using both the traditional RNN as well as the new CNN architectures, Fairseq also has support for language modeling and text summarization.

Using the Frameworks

Both frameworks have a similar structure. They each contain a Python API to build and train a model, as well as a set of configurable top-level scripts to preprocess data, train the model, and generate test translations. These scripts provide an easy way to get started with each of the frameworks.

OpenNMT goes a step beyond by providing a Dockerfile to generate a fully configured build environment. It also includes a set of useful tools from third party projects. One of these is a script to generate BPE tokens. BPE tokenization is a process to further divide text into sub-word units to make better use of a smaller number of tokens in the vocabulary. While the Fairseq framework understands BPE tokens, the user is left to use other toolkits to process the training data.

Another strong point for OpenNMT is automatic support for unknown token substation. With appropriate options set in the preprocessing step, OpenNMT will store the raw training data and combine it with per word attention scores to deduce which source word was not found in the target dictionary. Fairseq is also able to perform unknown token substitution, but requires the user to use at least two other toolkits to generate an alignment dictionary mapping between source and target language vocabularies.

Fairseq, however, shines in its support for advancing research NMT models. It natively supports several variant CNN models that have achieved state of the art results in machine translation. OpenNMT, in contrast, only offers experimental support for CNN models and in our initial experiments we were not able to generate a model that was able to converge.

In addition, Fairseq is highly extensible and provides a registry system that allows users to develop new language modeling tasks and model architectures that can be integrated seamlessly and without code changes into its top-level training scripts. By using Fairseq's support for customizable tasks, we were able to use a pre-trained language model to score our translations without using yet another third party NLP framework.

CONCLUSION

OpenNMT and Fairseq are both robust and capable toolkits based on the popular PyTorch machine learning framework that can be leveraged for GEC research. Of the two, OpenNMT provides a gentler learning curve by providing more support for traditional NMT tasks and models. In fact, we were able to use OpenNMT in our early experimental prototypes to build simple models to prove that GEC functionality could be built from deep neural networks. Ultimately, however, we chose to focus our work on Fairseq. Although its initial steep learning curve was substantial, its support for state of the art models and high degree of extensibility made it possible to extend the framework to work in new ways for this project.

RESOURCES

Fairseq:
https://github.com/pytorch/fairseq/
https://fairseq.readthedocs.io/en/latest/

OpenNMT:
https://github.com/OpenNMT/OpenNMT-py/
http://opennmt.net/OpenNMT-py/

Harvard Launches Open-source Neural Machine Translation System
https://slator.com/academia/harvard-launches-open-source-neural-machine-translation-system/

A novel approach to neural machine translation
https://code.fb.com/ml-applications/a-novel-approach-to-neural-machine-translation/