Merian I Croos
CS410 Text Information System
Technology Review

## Introduction

Tensorflow is an end-to-end machine learning platform which can accelerate machine learning tasks and solve real-world problems with Machine Learning(ML) [1]. Tensforflow can be used for data preparation, build machine learning models , deploy models and implement MLOps to run models in production. Tensforflow models can be deployed in multiple platforms such as Android, iOS, and Web apps.

The main advantages of Tensorflow as promoted by the framework is the ability to build ML applications faster by utilizing pre-trained models from the Tensorflow HUB, research with state -of-the-art models using the model garden and build your own models using the Tensorflow core.

Tensorflow is also makes deploying anywhere, at any scale very easy with Tensorflow.JS library for the web, Tensforflow Lite for mobile and edge, and TFX library for servers. Tensorflow is backed by Google with a well maintained codebase and large community support.

Tensorflow support developing ML applications for vision, Natural Language Processing (NLP) and audio use cases. In this review I will focus on tools and library support for NLP tasks exclusively. Especially, the text generation, text translation, text classification and support for BERT are explored in this review.

## NLP Tools and Features

An NLP machine learning tasks involves several stages from preprocessing of texts, model training and model inference. Tensorflow provides several reference designs for text generation with RNN, text translation with seq2seq and transformer models. Text classification with BERT and RNN.

## Preprocessing

Tensorflow provides rich collections of libraries to support essential NLP preprocessing tasks such as text normalization, word tokenization and sub-word tokenization. Tensorflow makes available a tensorflow_text library also known as  TF.Text which supports text normalization and word tokenization. These features are provided via the Splitter, Tokenizers and Subword tokenizer API. Splitter API provides basic functionalities such as text splitting with offsets while tokenizer provides support WhitespaceTokenizer which splits strings on ICU defined whitespace characters while UnicodeScriptTokenizer splits string based on Unicode script boundaries.  Subword tokenizers include a data-driven tokenization schema called Wordpiece tokenizer, BertTokenizer based on the BERT paper, and a highly configurable SentencepieceTokenizer backed by Sentencepiece library. Besides these core tokenizers the TF.Text also provides splitters such a UnicodeCharTokenizer for splitting Unicode characters and RegexSplitter which can split string by defined regular expression.

**Text Generation**

Tensorflow provides three reference notebooks for implementing text generation applications using RNN, seq2seq  and transformer models. The RNN reference design demonstrates text generation using a character-based RNN using a dataset of Shakespear's writing. The objective is to predict the next sequence of characters given a sequence of characters from the Shakespear dataset.  The RNN model demonstrated has an embedding dimension of 256 and configured with 1025 Gated Recurrent Units (GRU).

Sequence-to-sequnce(seq2seq) model, a neural machine translation with attention model based on Text Translation Effective Approaches to Attention-based Neural Machine Translation (Luong et al., 2015) is provided as reference design. This architecture is demonstrated for translating Spanish-to-English using the Tensorflow core and TF.Text text preprocessing libraries.

The final reference design for text generation is a neural machine translation with a transformer and Keras. This design demonstrates Portuguese to English translation using a Transformer network which was originally proposed by Vaswani et al. (2017) in the paper "Attention is all you need".  The reference tutorial demonstrates full end-to-end design of the network from data preprocessing, implementation of components to generate positional embeddings, attention layers, the encoder and the decoder.

**Text Classification**

Tensorflow provides two reference applications to demonstrate NLP text classification tasks. The first design is based on Bidirectional Encoder Representations from Transformers(BERT) which uses the transformer encoder architecture to process each token before and after in the full context of all tokens. Tensorflow provides pretrained BERT models trained on large corpus of text which then can fine tuned to the client application specification. In this reference design IMDB dataset is extracted and trained on BERT to perform sentiment analysis. This model demonstrates incorporating a classifier with BERT model and techniques to tune the BERT model to suit our application needs.

## Conclusion

Tensorflow is a powerful framework not just for computer vision tasks but also has introduced mature and powerful libraries to handle NLP tasks from preprocessing, model building to deployment. The extensive tutorials and reference documents makes it easier for any machine learning enthusiasts to easily get up to speed on the NLP applications via the downloadable well documented notebooks. The Tensorflow core library along with TF.Text provides all the state-of-the-art core components to not just develop simple applications but also engineer NLP models for research and  new model algorithms development and experimentation for NLP.

Tensorflow is also readily supported and optimized for NVIDIA embedded systems such as the Jetson series via the TensorflowRT library which makes Tensorflow a powerful library all around.

[1] https://www.tensorflow.org/