Jie Zheng - jzheng5
Yuanming Mao - ym18
Albert Yeh - ayeh2

CS410 - Classification Competition
Final Report

Summary

Our team was formed over slack when we were looking for fellow students to work with in the same metropolitan area. We had wanted to have the option to meet up in person, as we are all from southern California. Unfortunately, due to COVID 19, all of our meetings ended up just being virtual meetups. When one of us stumbled across the suggested tech review suggestions, BERT/ALBERT/Transformers caught our attention and appeared to be a perfect solution for the classification competition. Because the competition had suggested "You will need to research by yourselves some cutting-edge models that are more recent than those introduced in the lectures" and "achieve the state-of-the-art performance", we quickly realized that BERT and its many variants are going to be the perfect solution for the classification task at hand.

Initial Research

Each of our members spent many hours reading and researching much of the prerequisite knowledge required before understanding how to use BERT or how BERT worked. Starting from text preprocessing, tokenization, to word2vec/Glove/word embeddings, then understanding the use of artificial neural networks to recurrent neural networks, and LSTMs. After the initial research, we were finally able to understand the landmark paper, "Attention is all you need."[1] Transformers, a key part of all the variations of BERT and transformer-like models, required many hours of research. Additionally, we read "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"[2]. More importantly, our group was able to find multiple

[1] https://arxiv.org/abs/1706.03762
[2] https://arxiv.org/abs/1810.04805

papers on similar twitter classification problems which provided a source of inspiration on which models to use and how to preprocess the data.[3]

Trial and Error

Our team attempted a conquer and divide method for solving the classification competition. However, we were in constant communication with each other and ultimately one of the team members was able to achieve a score beating the baseline. When the team member who came up with the solution that beat the baseline, we decided to regroup and focus on the best solution. We attempted to both learn from and improve on the best solution to increase the F1 score. Ultimately, the best score was about 2 points higher than the baseline of .723.

Much of the learning happened throughout the trial and errors of each student. One of the approaches ended up with a F1 score of ~.68 and while ultimately scrapped, proved a useful learning tool. This method started using the pre-processing library "ekphrasis" found at https://github.com/cbaziotis/ekphrasis. This library offered the following:

- Social Tokenizer A text tokenizer geared towards social networks (Facebook, Twitter...), which understands complex emoticons, emojis and other unstructured expressions like dates, times and more.
- Word Segmentation. You can split a long string to its constituent words. Suitable for hashtag segmentation.
- Spell Correction. You can replace a misspelled word, with the most probable candidate word.

The pre-processing tools were very useful and was the biggest increase in F1 score from an initial score of around F1 50, to the mid 60's. In this approach, we learned about overfitting as using higher epochs, was not generally successful, and while it increased model training times, it did not do much for the score, or often ended up lowering it. We found this to be likely an overfitting of the model. We also considered the sample size of 5000 training inputs a potential factor as well. This approach also tried BERT's cased and uncased models but we did not find a significant difference in

[3] https://www.aclweb.org/anthology/2020.figlang-1.13.pdf and https://arxiv.org/pdf/2005.05814.pdf

the F1 score. Additionally, this approach found that the stopwords library from nltk.corpus had a marginal effect on the F1 score.

A second approach that the team took was using the RoBERTa model for solving the classification model but ultimately it also came up short with an F1 score of ~69. This approach attempted to follow the guide at https://rsilveira79.github.io/fermenting_gradients/machine_learning/nlp/pytorch/text_classification_roberta/. Our attempts to remove stem words, numbers, URLs, did not have significant impacts to our score. This approach also attempted to change the test_train_split size of .7 compared to .8 but did not lead to much success. We noticed that after removing punctuation during the preprocessing phase, that the classification accuracy decreased. We hypothesize this was due to the fact that punctuation plays a critical role in understanding the nuance of the english language, such as the question mark or exclamation mark, especially when it comes to training a language model to understand something as complex as sarcasm.

Last but not least, the final approach which was fully inspired by one of the many tutorials that our teams attempted to replicate, was found at https://analyticsindiamag.com/step-by-step-guide-to-implement-multi-class-classification-with-bert-tensorflow/. This approach also used the bert library optimization, tokenization as well as the library “tweet-preprocessor” and “preprocessor” which appeared to be extremely helpful. We used preprocess to clean up Emoji’s, @mentions, hashtags and more. Additionally, we added a custom preprocessor, to reverse the context, concatenation, and cleaned up some spacing issues. With this set of preprocessing, ultimately our approach that solved the classification competition used mostly the default hyperparameters from the BERT model (max_epochs = 3, learning_rate of .00001). Despite the group's best effort of trying to increase our F1 score past 74, we were unable to increase it significantly past this point. We believed that there was a ceiling to training the model because of overfitting.

Conclusion

Overall, the classification competition proved to be a worthy assignment for the group. As suggested by the initial competition requirement to use “cutting-edge models that are more recent than those introduced in the lectures”, we realized quickly that the lectures from ~2012 were likely to be quite dated, so we were excited to learn something new. The field of NLP has advanced leaps and bounds in just the last few years, especially with the introduction of CNN’s, LSTMs, and Transformers. Despite being dated, the vast majority of the lectures were fundamental in providing the foundation for our understanding of the state-of-the art NLP models. From bag-of-words in lecture 1 to TF-IDF weighting to understanding sentiment analysis and classification, each student was well prepared to digest the papers on transformers and BERT. We understand that while there was certainly more room to improve both the parameters for this competition or potentially concatenate additional preprocessors/models to get a better score, the fundamental ideas were well learned by the group.