

Samir Nafez

2 June 2023

CSE 144

Assignment 3 Report & Discussion

For this assignment, I had to try many different things in order to get my preprocessing working properly. Initially to get my sentiment analysis I had tried tokenization from the Hugging Face library which helped tokenize the input text into separate tokens, this first step was necessary however I began to realize that this is a very big data set. Upon realizing this, I attempted to use truncation to limit maximum length of each token which did help a bit with my overall performance. Regarding choosing my model architecture, I based some of my code off of the provided transformer.py which used the BERT model as the foundation for the sentiment analysis. I attempted to use other transformer based models but the problem was I could not get them to work properly so I ended up going back to the TA's example code to begin.

I chose to use an optimizer I found online from <https://keras.io/api/optimizers/> which led me to choose the AdamW optimizer with a learning rate of 2e-5. As far as the number of epochs to run through, I chose 5 because 10 was too much and took far too long and less than 5 was not enough. It is important to note that my training loop did end up working for me but took incredibly long. I was unable to find a way to better optimize it and after much trial and error this was the only loop that ended up working and producing output for me. My model provided the following measurements: Accuracy: 91.2%, Precision: 87.6%, Recall: 91.7%, F1-Score: 90.3%. The traditional machine learning model provided the following: Accuracy: 82.5%, Precision: 80.1%, Recall: 84%, F1-Score: 82.9%. These results were not very surprising as the BERT

model had higher results in all aspects. This is because models like BERT are able to learn complex features from raw text data and are able to retrieve far more specific information. These deep learning models are able to learn context and semantics which allows them to learn natural language. Traditional learning models like Naive Bayes typically work in a linear fashion and have one set way of doing it whereas a deep learning model evolves and is non-linear in all ways of learning.

Using deep learning models for sentiment analysis is an amazing feature humans have discovered and holds much power in todays world, however there are certain limitations as well. Deep learning models have become really good at discovering complex relationships in data and has been able to understand context which allows it to understand human language better. In using deep learning models in areas such as Twitter for sentiment analysis of tweets, we have the ability to categorize mass amounts of data and determine the sentiment. This is saving companies a lot of time and is also allowing a lot of data to be processed at a much higher rate. These models are also saving the manpower of needing to sift through huge amounts of data to determine sentiment analysis and avoid traditional learning models which work but are nowhere near the complexity of deep learning. Models such as BERT and GPT have been pretrained on knowledge learned from large scale datasets which help in understanding human language and context. However, there are some limitations to these deep learning models. For example, GPT was trained on data from 2021 and prior meaning that it is not up to date with current events and is slightly behind on the worlds data. Another issue is that these deep learning models more often than not need to be trained on huge amounts of labeled data to get maximum performance which can take long or be hard to gather. The reason for this is that as we have learned, if you attempt

to train a model on a dataset too small the model may have overfitting or may have limited knowledge and output biased results.

For future research it may be beneficial to compress these models in order to help reduce the computational requirements needed to run these models. Even in my model if there was someway to do this it would increase my performance and efficiency a lot. Regarding sentiment analysis, it can be difficult since it may not always be trustworthy or may be biased. Working to continue to make this close to perfect would benefit deep learning models a lot as they can produce misinformation or inappropriate responses. This will be hard to do as language context and things like tweets do not always have obvious context or can be hard to determine if something is serious or a joke. Boosting the understanding of linguistics in these deep learning models would increase their trustworthiness by a lot and maybe even help the general public become more accepting of deep learning. This might be able to be done by training a model on data that has correlations to human language such as irony, sarcasm, and logical fallacies. In conclusion, deep learning models so far are making massive improvements in sentiment analysis but still have a lot that can be improved.

Works Cited

Team, Keras. “Keras Documentation: Optimizers.” *Keras*, keras.io/api/optimizers/. Accessed 2 June 2023.