Victor Morgan
11/02
CSC-570
Dr. Osunmakinde

Conversion to my AI field of choice

For my Natural Language Processing Model Comparison program I wanted to get a full scope of what each model is useful for. I implemented 3 models, the first being FinBert, a model tailored towards financial sentiment analysis. I also used distilbert, a deep learning model, and TextBlob as a baseline for the comparisons. I tested each model against 3 types of data. Financial news, Healthcare reviews, and Amazon Retail reviews.

The PEAS are as follows:

Performance : The programs' ability to test the program using different datasets, .csv and .txt.
Environment : The Finance and Healthcare csv files, the amazon review txt file, and any additional csv or txt files the user identifies for testing.
Actuators : Utilizing each dataset to test model accuracy. Outputting model accuracy and confusion matrix for comparison.
Sensors : The raw inputs pulled from each dataset, Tokenizers from NLTK, TextBlob, and Hugging Face Transformers, FinBERT, DistilBERT, and TextBlob.

I designed my program to be semi-autonomous. It's autonomous for the first run, automatically checking 3 files I've chosen, and then it opens up to user file requests, where they can enter the path name or q to quit.

Feature Engineering

My first and largest feature was the addition of two additional NLPs. I added both FinBERT and DistilBERT, to be compared against textblob. In addition, I use one file at a time to test all 3 models, and then output their confusion matrices and accuracies.

A more subtle feature is my output. I observed the time it takes these models to run, and so I decided to also let the user know the runtime for each model when they receive the accuracy. I've formatted the output so the user can easily discern which model results they're looking at, and for which domain(in the autonomous phase) the models were tested against.

Next is a feature I added for FinBERT. Since this is matching my area of research, finance, I decided to focus on it more than the others. FinBERT predicts neutral cases, so I decided I'd showcase the model including and not including those neutral cases. Each body of text receives a positive, neutral, and negative score, with the highest being the determined sentiment. For neutral cases I sided it either to positive or negative depending on which one was higher. I also displayed the accuracy of the model’s determined positives and negatives, by removing the neutral cases altogether.

My next feature involves the types of files I accept. I wanted to be able to make use of both the csv datasets I found, and the txt file provided in class, so I modified my program to check for the file extension to see if it matches csv or txt, and handle both cases accordingly. CSV sticks to the comma delimiters, while with txt the delimiter is set to tabs.

Since my model runs continuously, I also made sure to clear my dataframes so that no data gets mixed in with others. I also make sure to close any plots, and I convert any non-integer sentiments such as “positive” or “negative” to 0 or 1, so the user has more flexibility with the datasets they choose. This is handled dynamically, so sentiments that are already 1 and 0 don’t get overprocessed.

Conclusion

In conclusion, I’ve managed to design and produce a program that has increased flexibility when handling different file types and different types of columns: categorical and non-categorical sentiment. I’ve implemented two new Natural Language Processors, FinBERT and DistilBERT and compared them to each other and TextBlob. I made use of different areas of research as it pertains to my datasets: Finance, Medical, and Retail, and formatted my output to allow the user to know what’s happening at all times. I’ve also returned run-times for further comparison of each program. One program may be slightly better suited for a text theme, but have drastically larger runtimes. This is important when deciding which one is best to use.

This program has allowed me to confirm the usefulness of FinBERT for finances over other models. It has higher accuracy, but does require more time and processing power to run.