FinBERT:

Sentiment Analysis of Financial Text with a Pre – Trained Natural Language Processing Model that is a Variation of Bidirectional Encoder Representations from Transformers

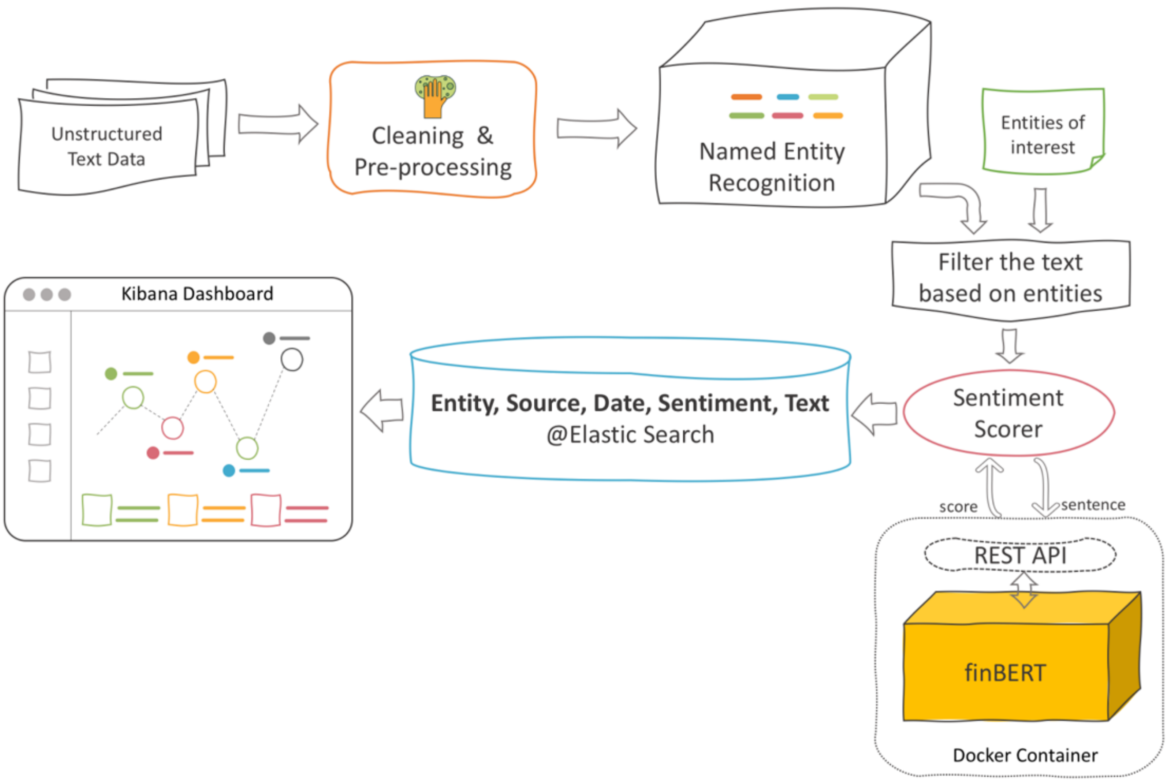
John Thomas Foxworthy

foxworthy@gmail.com

linkedin.com/in/john-t-foxworthy-1718073

https://github.com/sarilacivert/three/blob/main/finBERT_04b.ipynb

Los Angeles, California



Abstract

There is an overwhelming amount of financial news content for a retail investor to process, and there is a demand for automation so individuals can judge the direction of future asset returns. FinBERT is a Natural Language Processing algorithm for sentiment analysis through the Hugging Face library set that adds value by classifying text as positive, neutral, or negative. An implementation is processed on a large news corpus from the Financial PhraseBank with some overfitting concerns but with impressive results leading to future work in aggregating single company news into portfolio allocation.

Introduction

Financial text is accessible worldwide on various media channels for investors, but it takes too much time to process and interpret. One person that wants to invest in a single company's stock has to read dozens of financial news and opinions and then summarize content as positive to invest or negative to not invest. The manual process can always get stuck on a single headline, ignoring other headlines. Preconceived notions may miss the overall view that there is more positive than negative news because investors, like people, have biases.
One helpful task is to allocate phrases into quality categories of positive, negative, and neutral using Natural Language Processing (NLP). Aggregating sentiment would yield the big picture of a company's stock, and one type of NLP, FinBERT, would take advantage of this efficiency. As a Pre–Trained Natural Language Processing Model that is a Variation of Bidirectional Encoder Representations from Transformers (BERT), FinBERT, uses Deep Learning to classify financial news.

NLP with BERT is a layered machine learning model or deep learning model that replicates human cognitive attention by differentially weighting the significance of each part of the input data, and FinBERT focuses on the financial domain. Grammar and context learn the language as BERT encodes by producing embeddings that enumerate words with similar meanings. Decoders produce the following words from the embeddings as BERT transforms the language into a network. FinBERT builds on BERT by further training on a large financial corpus and fine-tuning for financial sentiment classification.

Literature Review

Writing about the FinBERT is not common, but there is a handful of literature resources in academia, books, and online data science content providers. The most commonly cited source for FinBERT, is the creator at the University of Amsterdam[þ]. In addition, there are an article online at Medium.com, a page in a practical NLP book from O'Reilly Media, and a graduate student's paper at Stanford University's Department of Computer Science.

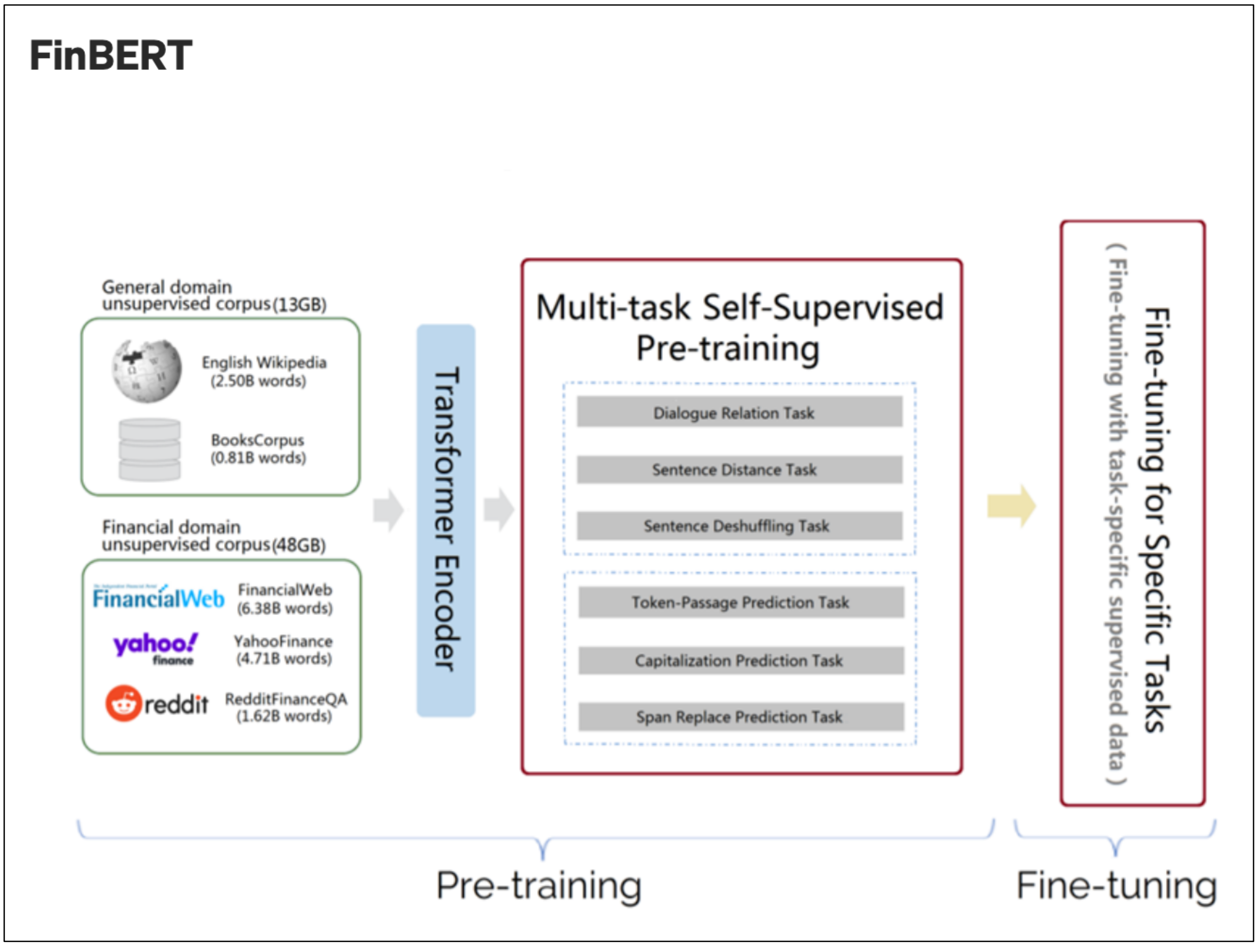
In 2018, Jacob Devline and his team at Google, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova created Bidirectional Encoder Representations from Transformers or BERT. Less than a year later, in 2019, Doğu Tan Aracı[§] implemented BERT for the financial domain by pre-training it on a financial corpus, configuring it for sentiment analysis, and calling it FinBERT. It was the first application of BERT for finance with pre-training on a domain-specific corpus. FinBERT was more accurate than other models, such as the Embeddings from Language Model (ELMo) and the Universal Language Model Fine-tuning (ULMfit). Moreover, FinBERT was able to surpass a large dataset of 3,000 examples and 500 examples, yielding a result of quantity and quality in tandem.

The two Medium articles on FinBERT step through implementations of financial news sentiment analyzer to classify a headline as negative, neutral, or positive to add business value. Raviraj Shinde begins with BERT and extends to FinBERT in a short and summarized sequence, while Poulinakis Kon has a simple and short walkthrough that is not detailed about FinBERT.

þ The creator of the Python Programming language, Guido van Rossum, also went to the University of Amsterdam.

§ Pronounced Do who Tahn Ara je

Shinde shows a flow chart on the FinBert processing below, while Kon steps through his code in a practical setting.



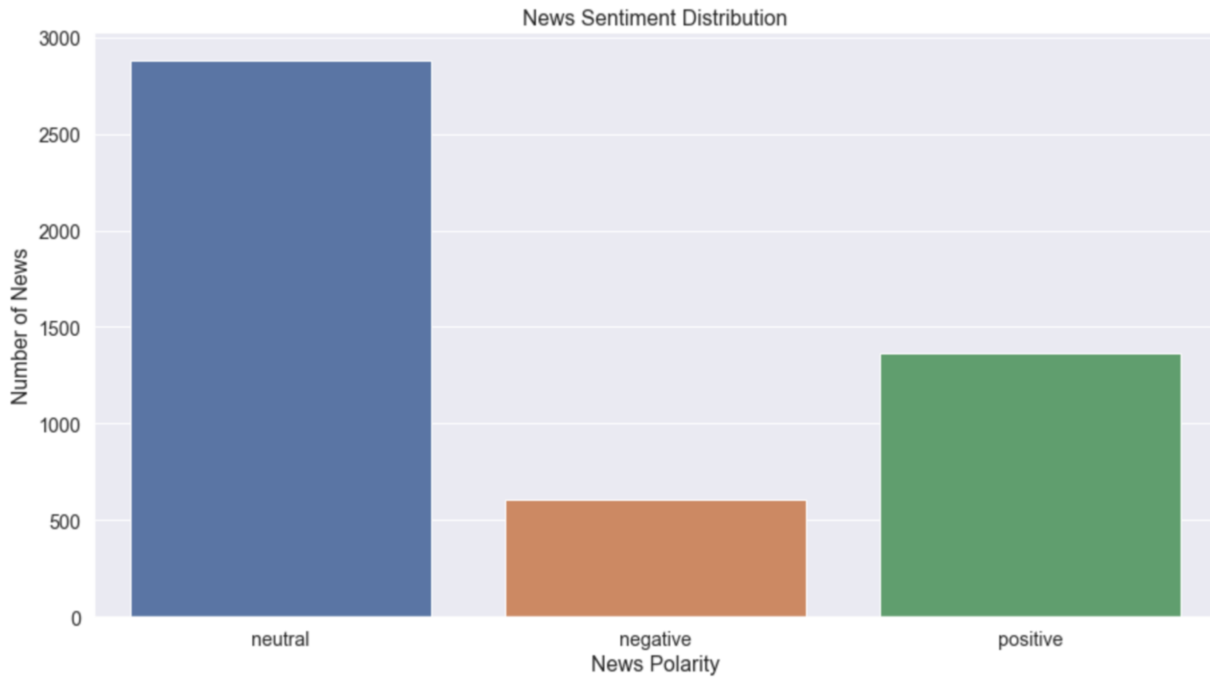
Oussama Fadil as a graduate student at the Department of Computer Science at Stanford University with a practical implementation of FinBERT on stock trading for Hedge Funds. He takes FinBERT further from sentiment analysis and uses it to predict the number of shares, not the returns of stocks, with a shortened version of BERT called DistilBERT. Fadil praises the performance and very high accuracy and hints to future work by scaling his work.

Lastly, in their Practical NLP book, Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana do a short deep dive on FinBERT. Positive sentiment in FinBERT refers to a positive increase in a company's stock, while negative sentiment is the opposite. Also, FinBERT architecture uses 4,000 sentences labeled by various people in business and finance from the Global Financial News Media company, Reuters.

Altogether, for the Literature Reviews, FinBERT uses Natural Language Processing to classify sentiment just like a human reading the article, as the definition of Artificial Intelligence is something non – biological that behaves biologically. So, potentially, FinBERT would replace dozens of human, financial news readers by automating and classifying sentiment.

Data

The dataset I will use is the Financial Phrase Bank from Kaggle, which has sentiments for financial news headlines for retail investors based on a research paper in the Journal of the American Society for Information Science and Technology. Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala are five professors at the Aalto University School of Business with mixed doctorate backgrounds in Computer Science, Economics, Finance, Linguistics, and Statistics. The team of researchers created the painstaking task of creating a dataset of 4,846 headlines and classifying each into sentiment categories of positive, neutral, and negative. There are 1,363 positive sentiments with 604 negative and 2,879 neutral. Alternatively, a single person would take many months, if not a year, to create a rich dataset for pre-training. The dataset is vital to capture the domain-specific language in the financial news media that influences the financial markets, such as the stock markets.



Next, the data preprocessing will extend to something noteworthy, as in the count plot above. The Sentiment Polarity Distribution as a form of Exploratory Data Analysis shows an imbalanced dataset, raising caution when preparing the training and testing datasets. The concern is that we have enough of each sentiment type for training and valuation data sets. At this moment, Exploratory Data Analysis is preliminary, judgmental, and incomplete.

Methods

The FinBERT algorithm, through the creator, Doğu Tan Aracı's GitHub, is the primary method followed by encoding, data creation, getting the tokenizer, and setting up. The overall goal is to prepare the dataset to run the FinBERT algorithm and optimize a scheduler. There are three preparatory steps before the setup outlined below.

First, we transform words into numbers to capture by using the encode sentiment values function. Specifically, possible sentiments are filtered as unique in a dictionary data type in the Python Programming language, then labeled. Below are the first five rows of the dataset ready to be encoded.

	sentiment	headline
0	neutral	According to Gran , the company has no plans t...
1	neutral	Technopolis plans to develop in stages an area...
2	negative	The international electronic industry company ...
3	positive	With the new production plant the company woul...
4	positive	According to the company 's updated strategy f...

The column to the right in blue is the encoded sentiments with neutral, negative, and positive labeled as 0, 1, and 2, respectively.

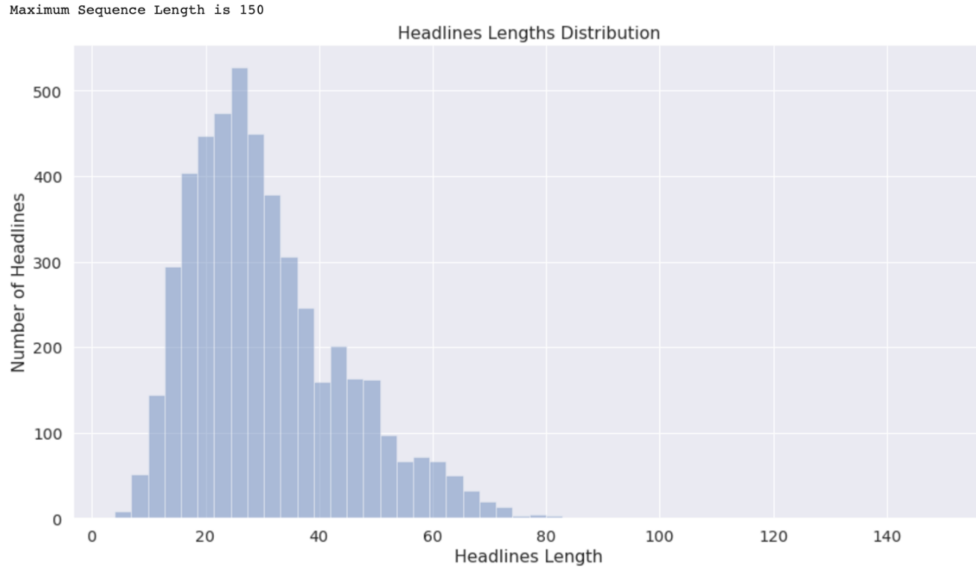
	sentiment	headline	label
0	neutral	According to Gran , the company has no plans t...	0
1	neutral	Technopolis plans to develop in stages an area...	0
2	negative	The international electronic industry company ...	1
3	positive	With the new production plant the company woul...	2
4	positive	According to the company 's updated strategy f...	2

Second, the data creation of the training and validation datasets to evaluate the model's performance by splitting the original data but ensuring each sentiment type is present. The train test split function uses the stratify argument, and the test size is 15%. The screenshot below has many occurrences for each type of negative, neutral, and positive sentiment value.

			headline
sentiment	**label**	**data_type**	
negative	**1**	**train**	513
		val	91
neutral	**0**	**train**	2447
		val	432
positive	**2**	**train**	1159
		val	204

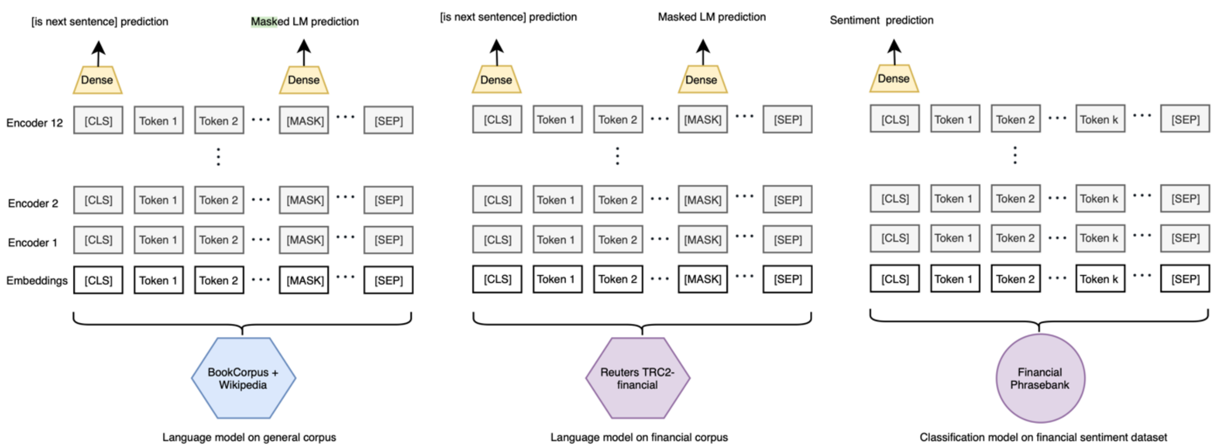
Third, FinBERT has a fixed vocabulary and a certain way to handling words, so we need tokenization. The maximum sequence length is 150, but the common tendency is between 20 and 40. Moreover, to break down text for Natural Language Processing, tokens as words, characters, or subwords* are classified.

```
Maximum Sequence Length is 150
```

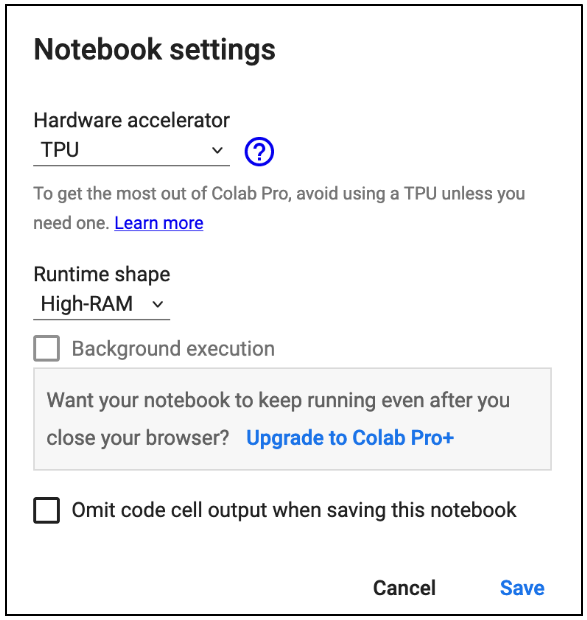


* A subword for example is a string part of a longer string like the word replace in replacer

Tokens are added to the start and end of sentences, text inputs are padded with similar shapes and sizes plus truncation when necessary for efficient processing. Below is an infographic from the original FinBERT paper as Doğu Tan Aracı expresses the pre – training steps on the labeled sentiment dataset, including fine – tuning. The overall objective is the attention mask that allows sending a batch into the transformer even when the examples in the batch have varying lengths and made possible. In particular, a Tensor in the PyTorch library is a multi – dimensional matrix containing elements of a single data type, similar to a Python List data type that turns tokenized inputs into a sequence of integers that correspond to items in the transformer's library.



Fourth, is the last preparation before obtaining the model results by loading the prepared dataset and configuration. The Google Colab platform with notebook settings of Cloud Tensor Processing Units (TPUs) for the Hardware Accelerator and High – RAM in the Runtime Shape for the fastest and most efficient processing. The settings are in the Runtime menu under Change Runtime Type.



The HuggingFace transformers library has a set of Auto Classes to choose the architecture of the FinBERT algorithm. The AutoModelForSequenceClassification is a generic model class that selects the FinBERT pretrained model, the number of labels, and other variables. We do not need output attentions or output hidden states, so we set them to False as the information is not useful for our results analysis and interpretation.

Next, we generate the data loaders setting the batch size to 5 as a start by creating an iterator with the DataLoader class and then configure the optimizer and scheduler. The training hyperparameters are AdamW optimizers, a stochastic optimization method that modifies the typical implementation of weight decay with a learning rate and numerical stability settings. The default learning rate is 0.001, but given the size and shape of the dataset, the learning rate has been divided by 100 to 0.00001 or, in scientific notation, 1e – 5. Also, the numerical stability parameter or machine epsilon is set to the default of 0.00000001 or, in scientific notation, 1e – 8. Lastly, since higher values take too much time for training, the number of epochs has been set to

3, and the scheduler is defaulted with linear scheduling, so the learning rate schedule increases linearly from a low rate.

Results

Accuracy and the F1 Score evaluate the FinBERT model's performance, and it took a little over 50 minutes to process the three epoch passes for the 4,846 headlines in the dataset. By looking at what comes out of the algorithms, we have the training loss, the validation loss, and the F1 weighted score outputs of the three epochs. The training loss begins with 0.46 in the first epoch, then 0.25 and 0.15 in the other epochs. After that, the validation loss went from 0.43 to 0.48 to 0.53, and the F1 score was steady at 0.86, 0.87, and 0.87. Each of the three epochs took about 17 minutes to process, with the first epoch at 16 minutes and 56 seconds, the second epoch, cumulatively, at 34 minutes and 27 seconds, and altogether at 50 minutes and 29 seconds.

Analysis and Interpretation

The first overall interpretation is the overfitting that begins in the second epoch as the training loss keeps decreasing while the valuation loss keeps increasing. To stop the conceptual overgeneralization of the text-based dataset, the model should stop in the first epoch. Therefore, the second and third epochs were unnecessary as the first epoch is the best model. The F1 score that measures the accuracy on the dataset is good in the high eighties, but not very good if it was 0.90 or more.

Furthermore, the last block of code loads the first epoch model and predicts neutral, negative, and positive sentiment in the AutoModelForSequenceClassification function. From our Exploratory Data Analysis of the Sentiment Distribution, the neutral sentiment had 2,879 headlines of the overall 4,846, or almost 60%, and had a prediction accuracy of 395 over 432 or a little over 91%, which is impressive. Negative sentiment labeled news has the smallest share in the dataset with just more than 12% and prediction accuracy of 73 over 91 or just a little more than 80%. Positive sentiment linked to favorable rates of return was about 28% of the dataset and prediction accuracy of 160 over 204 or about 78%. Overall, there are high accuracy and evaluation scores, regardless of the imbalanced dataset in the Sentiment Distribution.

Conclusions

The significant advantage of FinBERT is that you do not need a large dataset for fine-tuning because the model learns about natural language in the model training. Historically, the abundance of textual data was a requirement for prior models like ELMo and ULMfit, leading to storage, capacity, processing power, and more data engineering issues. It is BERT that introduces the transformer, the T of BERT, and bi-directionality, the B of BERT, to progress the practice of Natural Language Processing. To elaborate, the pre-training and fine-tuning of BERT architecture would popularize the machine translation process and the Masked Language Modeling (MLM) task with 15% random masked tokens for prediction to enable bi-directionality or opposite direction data flows. The data preparation logic and model processing efficiency of BERT coupled with the application of the Financial Phrasebook dataset for the FinBERT algorithm provides interesting implications for retail investors to buy, sell or hold their trading positions.

However, there are disadvantages because we assume the sentiment of the headlines is correct and representative of the content of the news articles. Does the headline sentiment match the article sentiment, or is it clickbait? Some online content can be deceptive, sensationalized, or even misleading, but such content is minimal otherwise, retail investors would stop reading a specific online posting source because they would lose money on their trading positions. Also, and more importantly, why did FinBERT fail? The accuracy failure is much smaller regardless of the sample size and sentiment type distribution, but we have no explanation or path of analysis to investigate. Constant use of FinBERT does require caution as the absence of understanding of its inaccuracies leads to confirmation bias. In other words, prior beliefs or values look for validation, especially after several benefits of FinBERT for sentiment analysis of your retail stock positions. If you ignore your failures in financial news sentiment analysis and only validate your successes, you may risk a more significant failure without any preparation or understanding.

Directions for Future Work

Sentiment analysis can only identify direction of possible stock price direction with a high degree of accuracy, but it does nothing to tell you about how much you should allocate your trading positions in your overall portfolio. Aside from tinkering your batch size to a higher number than 5 for the FinBERT model for greater processing efficiency, FinBERT should be accompanied with portfolio analysis algorithm to generate a percentage of allocation based on a sentiment analysis. Each company that issues a publicly listed stock in a stock market would collect news stories in a data container, like the popularly available, Docker.

Since FinBERT processes polarity, a contemporary regression model like Support Vector Machine (SVM) would enumerate the polarity in one of the most cited financial econometrics that won the Nobel Prize in Economics. Eugene Fama, that received the Nobel Prize in 2013 teamed up with Kenneth French to create the Fama – French Three-Factor Model in the early 1990s with a classical linear regression model that could be updated with a Support Vector Regression with the same input features. The overall stock market return and then internal features of a specific company like its market size relative to other companies in an index and its growth style would explain a company's expected return and market risk. Essentially, an individual stock's historical returns would be assessed in terms of these expected returns with the polarity modeling of an SVM, an expected improvement of the linear regression so a retail investor can decide how much to allocate their trade positions. SVM processes data outliers better than linear regression because of the linear separability of a dataset by drawing a line to separate a cluster of data points.

Joining FinBERT would enhance SVM portfolio allocation. In other words, lots of positive sentiment in the financial news Docker container would accompany a historically low risk and expected high return company with a prominent position allocation. Conversely, lots of negative sentiment with high risk and low returns would have no allocation. Neutral sentiment would be something in between so a retail investor overall can process numerical information of a company's financial details with the rumors and facts in the financial news media industry.

Sources

Aracı, Doğu Tan. "Finbert: Financial sentiment analysis with pre-trained language models." *https://arxiv.org/pdf/1908.10063.pdf* (2019).

Aracı, Doğu Tan. "ProsusAI/Finbert: Financial Sentiment Analysis with BERT." GitHub. ProsusAI, 2021. *https://github.com/ProsusAI/finBERT*

Burnwal, Shivam. "Text Classification Financial News." Kaggle. 2020. https://www.kaggle.com/code/shivamburnwal/text-classification-financial-news/data

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *https://arxiv.org/abs/1810.04805* (2018).

Fadil, Oussama. "CS224N: Predicting Hedge Fund Holdings from 10K/Q text Analysis." Stanford CS 224N | Natural Language Processing with Deep Learning, 2021. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report074.pdf

Fama, Eugene F., and Kenneth R. French. "The cross-section of expected stock returns." *the Journal of Finance* 47, no. 2 (1992): 427-465.

Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." *https://arxiv.org/abs/1801.06146* (2018).

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *https://scholar.google.com/scholar_lookup?arxiv_id=1412.6980* (2014).

Kon, Poulinakis. 2022. "Stock News Sentiment Analysis with FinBERT Transformer." *Medium.* https://medium.com/codex/stocks-news-sentiment-analysis-with-deep-learning-transformers-and-machine-learning-cdcdb827fc06

Sarzynska-Wawer, Justyna, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. "Detecting formal thought disorder by deep contextualized word representations." *Psychiatry Research* 304 (2021): 114135.

Shinde, Raviraj. 2021. "*Financial News Sentiment Analysis using FinBERT.*" *Medium.* https://medium.com/@ravirajshinde2000/financial-news-sentiment-analysis-using-finbert-25afcc95e65f

Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. 2020. *Practical Natural Language Processing*. Sebastopol, CA. O'Reilly Media. Page 363

Malo, Pekka, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. "Good debt or bad debt: Detecting semantic orientations in economic texts." *Journal of the Association for Information Science and Technology* 65, no. 4 (2014): 782-796.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "*DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*". https://arxiv.org/abs/1910.01108 (2019)