Fine-tuned Transformers for Financial Sentiment Analysis

Dylan Newman
University of California San Diego
La Jolla, CA 92093
dmnewman@ucsd.edu

Carlos van der Ley
University of California San
Diego La Jolla, CA 92093
lwanderley@ucsd.edu

Xing Hong
University of California San Diego
La Jolla, CA 92093
xihong@ucsd.edu

Cain Chen
University of California San Diego
La Jolla, CA 92093
xic023@ucsd.edu

Abstract

Textual data in the financial domain is becoming increasingly important as the number of financial documents rapidly grows. With the progress in natural language processing (NLP), extracting valuable information has gained popularity among researchers, deep learning has boosted the development of effective financial text mining models and made significant breakthroughs in various Natural Language Processing tasks. State-of-the-art models such as BERT (Devlin et al., 2019) model developed by Google, GPT2 (Radford et al., 2018) by OpenAI, XL-Net (Yang et al.,2019), pre-trained on a large scale of unlabeled texts from various corpuses, have shown their effectiveness by achieving good results on general domain data. However, these models are not effective enough on finance specific language and semantics, limiting the accuracy that financial users can expect from their NLP models. In this project, we will finetune the popular transformers based on the pretrained models provided by Hugging Face using our collected manually labeled financial data. Furthermore, we provide web scraping utilities along with the specific financial domain language models, in order to better assist users in financial domain capture market trends and be alerted by the risk factors.

2 Introduction

In recent years, Deep Learning has revolutionized the development of intelligent systems in many fields especially in Natural Language Processing (NLP) using state-of-the-art architectures that significantly improved many NLP tasks. With the recent progress made in NLP, researchers are starting to pay more attention to tackling numerous tasks in finance. As the amount of textual content generated in the financial domain is growing at an exponential rate, natural language processing is becoming a strategic tool for financial analysis. For example, financial practitioners are often required to use a set of NLP techniques, such as financial text classification and sentimental analysis for risk assessment, stock investment, and market trend detection. Some of the researchers construct an end-to-end model for making the prediction. Sentiment analysis approaches are more common in this field, thus providing insights on financial decision-making. Such models include VAE, Capsule network, hybrid attention network, BERT, XLNet, etc. While Transformer based language models are widely used for product understanding, as well as market trend prediction. The most used NLP sentiment analysis method is polarity, which classifies the input text as positive, neutral, or negative. BERT (Bidirectional Encoder Representations from Transformers) is an open-source Machine Learning (ML) model for NLP well-used for classification. However, BERT is pre-trained on general English corpora, and the financial domain has its technical jargon, which can lead to an output misinterpretation. Therefore, it's not suitable for understanding domain-specific vocabulary. Nonetheless, FinBERT, based on the BERT model, fills the gap with a domain-specific terminology by overlapping BERT vocabulary (BaseVocab) for another (FinVocab). As the industry developed and transformer architecture being widely used, in order to standardize all the steps involved in training and using a language model, Hugging Face was founded. They're democratizing NLP by constructing an API that allows easy access to pretrained models, datasets and tokenizing steps. Within HuggingFace, Transformers was build as an open-source library with the goal of opening up these advances to the wider machine learning community. There are other open source libraries such as Spacy that are very helpful in creating fine-tuned models specific for NLP tasks. Spacy uses a CNN and can perform a wide variety of tasks such as tokenization, part of speech tagging, named entity recognition, sentiment analysis, and so much more. In terms of financial applications, a Spacy model can be trained to pull keywords out of tweets and determine popularity of certain stocks and decide whether to invest or not using that information or a combination of stock history/data. Most Spacy models use word vectors, representations of different words that allow you to compare similarities between other words, in order to accurately take into account all parts of the sentence. In addition, this project will also include useful tools for financial users to utilize, in order to better capture the market trend and avoid risks. Particularly, scraping tools for tweets regarding companies will be prepared, while we will build tools for scraping federal financial reports as well. Therefore, users can identify risk factors as well as catch market sentiment in real time, thus better assisting financial decisions.

Our contributions can be summarized as follows:

- Collect financial related manually sentiment labeled data for training
- Construct handful tweets scraping tool
- Polarize part of real time tweets text using Spacy to expand training dataset
- Fine-tune 5 popular transformers based on pre-trained models in HuggingFace
  - Bert
  - FinBert
  - Financial Bert
  - XLNet
  - GPT2
- Build handy pipelines and APIs to retrieve financial information from Twitter to detect stock related sentiment.

3 Methods

3.1 Dataset

The dataset utilized during the sentiment classification procedure is important to the research since it has a substantial impact on classification performance. Given the purpose stated, the dataset must be text with short length relevant to the stock market, as the objective is sentiment analysis of stock market or company analysis. The main sentiment analysis dataset used in this project consists of three parts: Financial PhraseBank, IEEE DataPort and Kaggle tweets dataset. All three datasets are financial related and manually labeled with sentiment scores. In total, the dataset collected for this project has 11,936 financial related text, each with a custom labeled sentiment score in 0 (neutral), 1 (positive) or -1 (negative). Due to the lack of training text in the financial field and the scarcity of manually labeled ones, the collected dataset may be insufficient to train a model with better performance. The raw data should be mapped with sentiment scores across a larger size of tweets to perform sentiment analysis. For sentiment analysis, these sentiment scores are afterwards concatenated into the training dataset. There are several Python libraries that are available, to use to accomplish this type of preprocessing in Natural Language Processing, for example TextBlob and Vader Analyzer. In this project, an open source package called Spacy will be introduced for polarization of larger data sets and be used for training and validation. As a result, we will have 80% of data for training and validation while 20% for testing.

Financial Phrasebank consists of 4,845 English financial headlines that were categorized by sentiment class and were annotated by 16 researchers with a financial background. The sentiment label is either positive, neutral or negative. The dataset is available in four possible configurations, depending on the percentage of agreement of annotators (50%, 66%, 75%, and 100%). In this project, we choose to have the whole Data (at least 50% agreement).

IEEE DataPort (Taborda et al., 2021), dataset can be found here The IEEE DataPort stock market tweets dataset consist of tweets between April 9 and July 16, 2020, using the SP 500 tag (SPX500), the references to the top 25 companies in the SP 500 index, and the Bloomberg tag (stocks). 1,300 out of the 943,672 tweets were manually annotated in positive, neutral, or negative classes. In this project, we will only take the manually labeled data.

Kaggle tweets dataset (Yash Chaudhary, 2020), dataset can be found here Gathered Stock news from Multiple twitter Handles regarding Economic news and custom labeled, which was divided into two parts : Negative(-1) and positive(1). Negative count: 2,106 Positive count: 3,685. We will be programming a downloading mechanism to download the Kaggle datasets while running the pipeline to avoid memory wasting caused by pre-saving datasets.

3.2 Time Series

In this quarter, we were able to implement LSTM, CNN, and TreNet as three different deep learning models to predict financial time series data. From our progress in this quarter, we are confident that our goals for FinDL are achievable within the given timeframe. We will first focus on conducting research into how we can improve the current implementations of our models and search for other popular and state of the art models to include in our library. Based on the models and improvements we want to implement, we will design the implementation of the library, including the deep learning models, data pre-processing, and feature extraction methods. We will focus on ease of use and straightforward customization of models and fine tuning parameters that is catered towards users in the finance domain. With the design of the library and deep learning models, we will implement all of the models and methods in a finance deep learning library for other users to use in their finance domain machine learning applications.

3.3 Tokenizers and Models

Throughout Quarter 1 and Quarter 2, we have been investing time exploring and studying transformers tokenizers and models. Transformers were first introduced in 2017 and ever since have been revolutionary. Instead of using RNNs or convolution, transformers adopt a self-attention mechanism that has been proven to be more efficient. Transformers models are built upon seq2seq models that consist of encoders and decoders. Taking a machine translation model for example, the encoder

encodes input text and the decoder decodes the outputs from the encoder. NLP models from Transformers are based solely on attention mechanisms in order to draw global dependencies between inputs and outputs. There are apparent benefits of the self-attention mechanism, including learning long range dependencies, yielding more interpretable results, and maximizing the amount of parallelizable computations. We observe that BERT is the most popular language model on the platform, and many other models are built based on BERT. BERT is a language model pre-trained on Masked LM and Next Sentence Prediction. It is widely used for sentiment analysis, sentence classification and interactive QA. We decided to look further into BERT and understand the various variations of it. In addition to BERT, we fine-tuned several pre-trained models based on other architectures, for example XLNet and GPT2. Upon researching, we discovered that some of the models listed on Hugging Face are specifically designed for financial texts. Combining the models and the Transformers' architecture, we aim to try out this new approach towards sentiment analysis for our final Project.

3.4 Pre-train Model Approach

For this project, we chose to fine-tune on the pretrained models provided by Hugging Face, which consists of three major steps when designing.

- Choose a source model: From the available models, a pre-trained source model is picked. Over the years, many teams and researchers trained models based on vast and difficult datasets, while Hugging Face collected most of them with easy to use APIs. Thus, we can easily find ideal models from Hugging Face hub based on our task. For example, since our datasets deal with financial stock information, we aim to find models that are specifically tailored for financial datasets such as finbert.

- Model for Reuse: The pre-trained model can then be utilized to build a model for the second job of interest, in this case, we can use them for financial sentiment analysis. Depending on the modeling technique employed, this may entail using all or sections of the model. We create an easy mechanism for users to select and download our pre-trained models to best fit their use cases.

- Fine-tuned: On the input-output pair data available for the job of interest, the model may need to be altered or refined, in this project, we will continue the training using our collected data and alter some hyperparameters. In our final library, we will add APIs for users to self-tune the models in case they find better parameters.

3.5 Model Details

3.5.1 BERT

The use of a pre-trained model is prevalent. BERT and XLNet, and other models are examples of this type. There are numerous advantages of employing transfer learning. Higher start, high rate of skill increase, and superior converged skill are a few of them (Dussa, 2020). Bidirectional Transformer Encoder Representations (BERT) is a bidirectional encoder transformer paradigm. This model was created to help Google AI Language pre-train deep bidirectional representations to extract context-sensitive properties from input text (Devlin et al., 2018). BERT learns contextual relationships between words in a text via an attention mechanism in the transformer. Transformer is made up of two mechanisms: an encoder that reads the text input and a decoder that generates the task prediction.

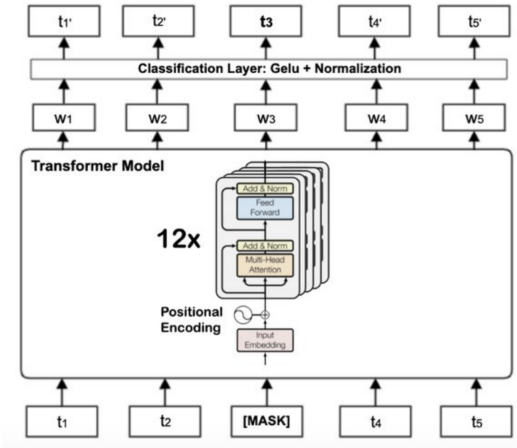

Figure 1. The Transformer based BERT base Architecture with twelve encoder blocks (Khalid et al., 2021)

One model that is trained based on the BERT model is the FinBERT model. This model is pre-trained to focus on sentiment analysis of financial text. (Araci, 2019) Like BERT, this model gives softmax outputs as labels for each piece of financial text. During training, we observe that performance of FinBERT appears to be better than the BERT model alone. Therefore we will use this model as our main model for training with additional arguments tuned for our training datasets.

3.5.2 XLNet

XLNet is a language model proposed by researchers from Google AI Brain Team and Carnegie Mellon University (Yang et al.,2019) that learns unsupervised representations of text sequences using a generalized autoregressive language model. BERT masks the words, presuming that the masked words have nothing in common. The interdependence of the disguised words is not considered. The XLNet system can overcome this disadvantage. XLNet employs the permutational language modeling technique. In order to cover both forward and backward directions, XLNet evaluates all potential permutations. Simply put, XLNet maintains the original sequence order, employs positional encodings, and employs a specific attention mask in Transformers to achieve the aforementioned factorization order permutation. To keep track of anticipated words and consider them in the next token prediction, XLNet employs a two-stream self-attention technique. (Yang et al., 2019; Dussa, 2020)

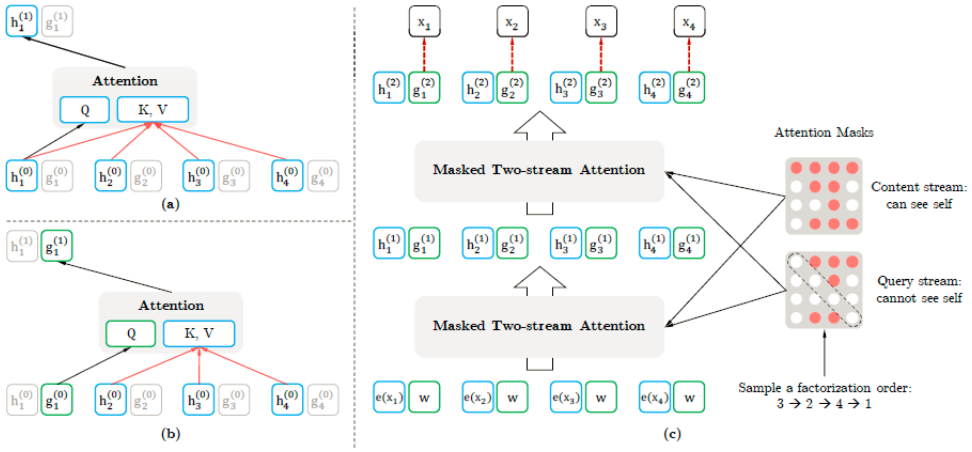

Figure 2. Diagram of XLNet's architecture (Yang et al., 2019; Dussa, 2020)

3.6 Implementation Details

The dataset was collected through three sources: Financial PhraseBank, IEEE DataPort and Kaggle tweets dataset. The features of the training data consists of financial text and the labels are either 0(neutral), 1(positive) or -1(negative). 11,936 samples were collected, and the training process is efficient with less time cost. Additionally, a pipeline for training multiple transformers was deployed on 1 NVIDIA 2080ti GPU. We tokenize our inputs using AutoTokenizer provided by Transformers on the pre-trained FinBERT model. While tokenizing, we pad the inputs according to the maximum length in the dataset to ensure that the vectors are of the same dimensions. FinBERT is a project brought out by Prosus that was fine-tuned on BERT specifically for financial text in 2019. Compared with the traditional BERT language model, FinBERT performs much better on financial text. Our training set consists of the FinancialPhraseBank, which was originally used to develop FinBERT, and the dataset of similar format. We use the Trainer provided by Transformers to train our model. Upon training, we have discovered that there are many ways we can fine-tune the already fine-tuned FinBERT model. We believe that we can achieve a better performance if we adjust the parameters specifically for stock information text.

3.7 Polarity Assignment and Target Dataset - Spacy

In order to run the package in real-time, a data generation pipeline had to be set up so current data could easily be pulled into the model for evaluation as well as building a target dataset for our final objective, capture market trend. The easiest way to do this was to use the internet's town square, Twitter. When a user inputs a certain stock into the model, e.g. AAPL, we used the Twitter API to go and grab the latest tweets containing the keyword "AAPL", and send those to the model for classification. This was great until we found out that Twitter was going to start charging for access to their API, something that had been free at a basic level for many years.

The only issue with this is that the data looks slightly different from training data, and to create a bit more training data as well as a new validation dataset on these tweets can be beneficial for a better performance. That is where the Spacy model comes in.

A Spacy model was fine-tuned in order to speed up the process of creating more training and validation data where sentiment could be generated automatically and then manually checked by a human. This speeds up creating new training data as the majority are already correct when generated and only few here and there need to be changed. For this project, the last 10000 tweets related with the five technology giants that dominate the SP 500 index was scraped and labeled with Spacy for training and testing purposes, they are Apple (ticker AAPL), Microsoft (MSFT), Alphabet (GOOG), Amazon.com (AMZN), and Facebook (FB).

3.8 Future Focus

During Quarter 2, our first objective was to try out more models on HuggingFace to get a better understanding of how different language models work in general. We would like to fine-tune them and eventually figure out what the best model is for our dataset. Then we would like to obtain more training data, preferably from real-time sources. By obtaining more data, we hope to further improve our accuracy on the sentiment analysis task. Finally, we would like to try to build our own language model for stock text, likely on top of BERT. Even though FinBERT has performed relatively well on financial textual data, we believe that a language model that is specifically built for stock textual data would better fit our goal. We aim to integrate our sentiment analysis method into the library that we will deliver at the end of Quarter 2. We would like to provide a model that is easy to understand and also user-friendly.

4 Results/Experiment evaluation

The following table presents the sentiment analysis results in a classification report on the test set. Although our fine-tuned models did not significantly outperform common baselines, the BERT-base (Devlin et al., 2019), they still achieved decent results. Our models achieved better performance than the state-of-the-art model for general tasks on testing dataset, which demonstrates its effectiveness in financial sentiment analysis.

Model	Accuracy	F1
BERT-base (Devlin et al., 2019)	0.76	0.83
BERT-base fine-tuned	0.81	0.86
FinBERT fine-tuned	0.83	0.86
FinancialBERT fine-tuned	0.84	0.88
XLNet fine-tuned	0.81	0.85
GPT2 fine-tuned	0.82	0.84

5 Discussion

Many works attempt to capture the price movement of the financial instruments. This project and the following package can be used for finance companies deploying models for automatic document classification. In this case, they will need approaches that can scale well and are simple to deploy, but have no need to train or fine-tune the model. By using the package we provide, they are able to search the model and use a pretrained and fine-tuned classification model designed for accurate, fast inference. If needed, continuing training on user customized data is possible. They were able to run and deploy the model directly from our package with no required research or ML expertise.

Risk is also an important attribute of financial instruments. For the purpose of detecting risk, we will build handy scraping tools to extract annual financial reports from the Security and Exchange Commission (SEC), which consists of Risk Factors and other crucial information in terms of investment. In other words, this package enables consultants to build a dynamic portfolio and monitor assets through market sentiment and evaluate the risk of a company from the financial reports.

6 Conclusion

Using a combination of different models we were able to achieve an accuracy above baseline model, although they are not significantly outperforming the baseline model, but it's as we expected given that our future plan consists of further training and fine tuning. Along with that, our F1 score was above baseline model as well. Looking at the F1 score is important in addition to the accuracy as it allows us to get a better sense of the data even if the distribution of the data is uneven. When feeding in data in real time, there may be a large influx of either positive or negative tweets about a certain stock that can skew the data. Overall, we were very pleased with the performance of our models and are excited to further increase the performance and utilities of our package next quarter.

7 References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information pro- cessing systems, pages 5998–6008.

Bruno Taborda, Ana de Almeida, José Carlos Dias, Fernando Batista, Ricardo Ribeiro. (2021). "Stock Market Tweets Data." Web.

Deli Chen, Shuming Ma, Keiko Harimoto, Ruihan Bao, Qi Su, and Xu Sun. 2019d. Group, extract and aggregate: Summarizing a large amount of finance news for forex movement prediction. In Proceedings of the Second Workshop on Economics and Natural Language Processing, pages 41–50, Hong Kong. Association for Computational Linguistics.

Dussa, A. (2020). Finetuning Pre-Trained Language Models for Sentiment Classification of COVID19 Tweets. ARROW@TU Dublin. https://arrow.tudublin.ie/scschcomdis/224/

Huang, Allen and Wang, Hui and Yang, Yi, FinBERT - A Large Language Model for Extracting Information from Financial Text (July 28, 2020). Contemporary Accounting Research, Forthcoming, Available at SSRN: https://ssrn.com/abstract=3910214 or http://dx.doi.org/10.2139/ssrn.3910214

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics

Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao, and Liang Yang. 2019. Transformer- based capsule network for stock movement prediction. In Proceedings of the First Workshop on Financial Technology and Natural Language Processing, pages 66–73, Macao, China

JOSEPH Engelberg and Pengjie Gao. 2011. In search of attention. The Journal of Finance, 66(5):1461– 1499.

Khalid, U., Beg, M., Arshad, M. (2021). RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning. Retrieved 5 December 2022, from https://arxiv.org/abs/2102.11278

Linyi Yang, Ruihai Dong, Tin Lok James Ng, and Yang Xu. 2019. Leveraging BERT to improve the FEARS index for stock forecasting. In Proceedings of the First Workshop on Financial Technology and Natural Language Processing, pages 54–60, Macao, China.

Malo, Pekka Sinha, Ankur Takala, Pyry Korhonen, Pekka Wallenius, Jyrki. (2013). FinancialPhraseBank-v1.0.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

Sara Sabour, Nicholas Frosst, and Geoffrey E Hin- ton. 2017. Dynamic routing between capsules. In Advances in neural information processing systems, pages 3856–3866.

Shubham Jangir. 2021. Finetuning BERT and XLNet for Sentiment Analysis of Stock Market Tweets using Mixout and Dropout Regularization, Dublin, Technological University of Dublin

"Spacy 101: Everything You Need to Know · Spacy Usage Documentation." SpaCy 101: Everything You Need to Know, https://spacy.io/usage/spacy-101.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 627–637, Copenhagen, Denmark. Association for Computational Linguistics.

Yash Chaudhary (2020). Stock-Market Sentiment Dataset: Positive-Negative sentiment at stock tweets

Yumo Xu and Shay B. Cohen. 2018. Stock movement prediction from tweets and historical prices. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1970–1979, Melbourne, Australia. Association for Computational Linguistics.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pages 261–269. ACM