Fine-tuning FinBERT to Perform Sentiment Analysis on 10-K Filings

Pranathi Alluri
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
alluri.p@northeastern.edu

Abstract—**This paper presents an in-depth investigation aimed at achieving enhancements in stock return prediction through sentiment analysis of 10-K financial filings. Amid the growing adoption of Natural Language Processing (NLP) in financial analysis, models like FinBERT have gained prominence. Acknowledging the inherent volatility of stock market dynamics, this study seeks to enhance the foundational FinBERT architecture by incorporating supplementary features and developing a multi-layered neural network.**

The research objective was to surpass the baseline FinBERT model's accuracy by a minimum of 10%. This goal was successfully achieved by the custom sentiment model. The analysis of misclassified filings revealed that while sentiment comprehension within these filings is achievable, translating it into precise market return predictions remains intricate and multifaceted. Moreover, this study establishes a robust groundwork for future investigations into refining sentiment-based predictive models for stock market returns.

Index Terms—**sentiment analysis, 10-K filings, FinBERT**

I. Introduction

In the domain of financial analysis, the evaluation of 10-K filings, which are comprehensive annual reports submitted by public companies to the U.S. Securities and Exchange Commission (SEC), holds significant importance. These filings serve as authoritative repositories of company information, disclosing critical financial performance metrics, risk factors, management discussions, and forward-looking statements. The intrinsic connection between the contents of 10-K filings and stock market behavior has spurred researchers and practitioners to leverage these insights for stock prediction.

Historically, understanding the content of these filings and extracting sentiment has been a common practice. Initial methods, such as Term Frequency-Inverse Document Frequency (TF-IDF) and bag-of-words, provided rudimentary sentiment analysis by quantifying word occurrences [1]. However, these techniques lacked contextual awareness and struggled to capture intricate linguistic nuances. While approaches utilizing specialized financial sentiment lexicons, such as the one proposed by Loughran and McDonald [2], seem appealing, their ”word counting” approach falls short when delving into the deeper semantic meanings of the text. Such limitations prompted the exploration of more sophisticated methodologies.

The advent of BERT (Bidirectional Encoder Representations from Transformers) marked a transformative milestone in Natural Language Processing (NLP). BERT's innovative bidirectional attention mechanism empowers it to comprehend the contextual significance of a word by considering its neighboring words on both sides. This unique approach equips BERT to capture the subtleties of language, rendering it highly effective across various NLP tasks, including sentiment analysis.

A. BERT

BERT's distinctive architecture comprises two core components: multiple transformer layers featuring self-attention heads, which adeptly model long-range dependencies, and masked language modeling, a process wherein tokens are masked, and the model predicts them. This architecture empowers BERT to discern the importance of each word in the context of its surroundings, enabling the modeling of contextual dependencies within textual data. The depth of this architecture, coupled with pre-training on extensive corpora, underpins BERT's impressive grasp of language semantics.

Leveraging BERT's capabilities, the financial sector introduced FinBERT — a customized adaptation tailored for financial sentiment analysis. FinBERT fine-tunes the pre-trained BERT model on financial text using Hugging Face's transformer libraries and refines it with labeled financial data. This strategic adaptation enhances FinBERT's adeptness at capturing domain-specific subtleties. Consequently, it elevates the sentiment analysis of financial documents, offering a nuanced perspective on market sentiment [3].

This paper aims to build upon the foundation provided by FinBERT to enhance sentiment analysis. The approach involves fine-tuning hyperparameters, integrating supplementary features, and comparing sentiment scores with historical trends to measure improvements. By undertaking these steps, the goal is to advance the accuracy and effectiveness of sentiment analysis, especially in the intricate landscape of financial text data.

II. Methodology

Three variations of the FinBERT model were employed to conduct a comparison of sentiment analysis techniques. The models included: the baseline FinBERT model applied directly to the data, FinBERT integrated with optimization and loss functions, and a custom sentiment neural model that leverages FinBERT as the initial layer.

A. Dataset & Preprocessing

The study employed a dataset of 10-K SEC filings sourced from Hugging Face [4]. This dataset featured columns encompassing a sentence from a 10-K filing, company particulars, and filing specifics. The focus is on sections 1A, 7, and 7A of filings, which have been demonstrated to heavily influence stock predictions, as well as post-2016 filings. Subsequently, sentences underwent concatenation, conversion to lowercase, and punctuation removal, rendering the text amenable for model training.

For sentiment analysis, labels were derived by tapping into historical stock prices and returns within a day from the filing date. Notably, a 2% surge in stock prices denoted positive sentiment, while a corresponding decrease signified negative sentiment; all other scenarios were categorized as neutral sentiment. The data was then split randomly into 80% training and 20% testing. This approach harnessed actual stock behavior to label sentiment, thereby augmenting the relevance and applicability of the sentiment analysis outcomes.

B. Model Architecture for Custom Sentiment Model

A neural network classifier was constructed using a pre-trained model, BERT, with the following structure:

- Bert Integration: The FinBERT pre-trained model was integrated, processing input text sequences to obtain a pooled output that captures key features from the text data.
- Feature Concatenation: The pooled output was combined with additional contextual features, such as sections, states, and timestamps. This combined feature vector enriched the model's understanding by incorporating relevant context.
- Batch Normalization: After feature concatenation, Batch Normalization layers were applied to counter internal covariate shifts, enhancing convergence speed during training. This process normalized layer inputs to zero mean and unit variance, promoting stability.
- Fully Connected Layers: The normalized feature vector passed through fully connected layers. The first dense layer (fc1) operated on concatenated features, transforming them into a higher-dimensional representation. Subsequently, a rectified linear unit (ReLU) activation function introduced non-linearity.
- Dropout: Following ReLU activation, a dropout layer was included. Dropout randomly deactivated neurons during each pass, mitigating overfitting by encouraging robust, generalized feature learning.
- Output Layer: The final fully connected layer (fc2) condensed the hidden representation to sentiment class count, producing logits. These logits underwent softmax activation to generate class probabilities."

C. Training Models

The FinBERT tokenizer was used to encode and build the data loaders for each model, in conjunction with KFold cross-validation using 5 splits. Furthermore, the Cross-Entropy Loss function and Adam Optimizer guided the training process, fostering accurate predictions and convergence.

1) Hyperparameter Tuning: In an effort to find the best parameters, a grid search was also run on the Custom Sentiment model. Four key hyperparameters were investigated:

- Number of Epochs (num epochs): The study considers two values for the number of training epochs, namely 5 and 10. This parameter represents the number of times the model iterates through the entire training dataset during training. A higher number of epochs might lead to better convergence, but excessive epochs can result in overfitting.
- Batch Size (batch size): The batch size determines the number of data samples processed together in each iteration. Two batch sizes, 16 and 32, are explored. Smaller batch sizes offer better generalization, but larger batch sizes might accelerate training due to parallelism.
- Hidden Size (hidden size): The hidden size defines the dimensionality of the hidden state in the model. It's examined with values 256 and 512. A larger hidden size might enhance the model's capacity to capture intricate patterns but can also increase computational complexity.
- Dropout Probability (dropout prob): Dropout is a regularization technique that randomly sets a fraction of the input units to zero during each forward pass. The study investigates two dropout probabilities: 0.1 and 0.2. Higher dropout probabilities can prevent overfitting by encouraging the network to be more robust.

Beyond fine-tuning these hyperparameters, the utilization of gradient clipping during training of the custom model addressed potential gradient explosion, contributing to model stability. In addition, the incorporation of a step-based learning rate scheduler facilitated smoother convergence during optimization

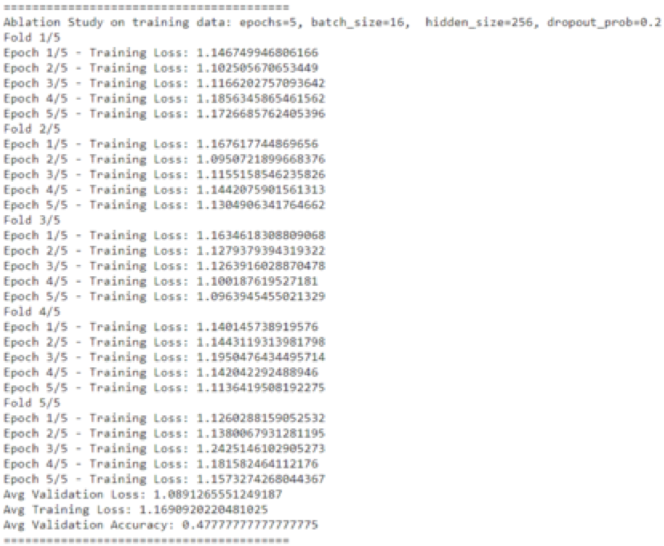
2) Model Evaluation: Validation loss provided insights into the model's generalization performance on unseen data, while training loss tracked the convergence of the training process. Validation accuracy served as a key metric to gauge the model's predictive proficiency. Through this process, the configuration yielding the highest validation accuracy was identified, establishing the foundation for the final model.

This selected configuration was used to train the final model, which was then fit on a separate test set (20% of the data, previously set aside). The evaluation encompassed various metrics to comprehensively gauge the model's performance on this previously unseen data. This approach not only effectively calibrated the model but also thoroughly evaluated it, ensuring its robustness and applicability in real-world sentiment analysis scenarios.

III. Results

The best-performing configuration for our 'customSentimentModel' after running cross-validation and grid search emerged as follows: 5 training epochs, a batch size of 16, a hidden layer size of 256, and a dropout rate of 0.2, as illustrated in "Fig. 1".

Although a validation accuracy of 48% is not impressive on its own, when comparing it to the other baseline BERT models, it becomes evident that the custom neural network model outperforms the others. Even simple fine-tuning of the baseline model results in better performance. This is evident through the 7% increase in accuracy from baseline finBERT, whose accuracy wasn't much better than random guessing, to finBERT with training and optimization. Additionally, incorporating finBERT into a neural network architecture that takes in other features yielded an additional 6% increase in validation accuracy.



```
========================================
Ablation Study on training data: epochs=5, batch_size=16,  hidden_size=256, dropout_prob=0.2
Fold 1/5
Epoch 1/5 - Training Loss: 1.146749946806166
Epoch 2/5 - Training Loss: 1.102505670653449
Epoch 3/5 - Training Loss: 1.1166202757093642
Epoch 4/5 - Training Loss: 1.1856345865461562
Epoch 5/5 - Training Loss: 1.1726685762405396
Fold 2/5
Epoch 1/5 - Training Loss: 1.167617744869656
Epoch 2/5 - Training Loss: 1.0950721899668376
Epoch 3/5 - Training Loss: 1.1155158546235826
Epoch 4/5 - Training Loss: 1.1442075901561313
Epoch 5/5 - Training Loss: 1.1304906341764662
Fold 3/5
Epoch 1/5 - Training Loss: 1.1634618308809068
Epoch 2/5 - Training Loss: 1.1279379394319322
Epoch 3/5 - Training Loss: 1.1263916028870478
Epoch 4/5 - Training Loss: 1.100187619527181
Epoch 5/5 - Training Loss: 1.0963945455021329
Fold 4/5
Epoch 1/5 - Training Loss: 1.140145738919576
Epoch 2/5 - Training Loss: 1.1443119313981798
Epoch 3/5 - Training Loss: 1.1950476434495714
Epoch 4/5 - Training Loss: 1.142042292488946
Epoch 5/5 - Training Loss: 1.1136419508192275
Fold 5/5
Epoch 1/5 - Training Loss: 1.1260288159052532
Epoch 2/5 - Training Loss: 1.1380067931281195
Epoch 3/5 - Training Loss: 1.2425146102905273
Epoch 4/5 - Training Loss: 1.181582464112176
Epoch 5/5 - Training Loss: 1.1573274268044367
Avg Validation Loss: 1.0891265551249187
Avg Training Loss: 1.1690920220481025
Avg Validation Accuracy: 0.4777777777777775
========================================
```

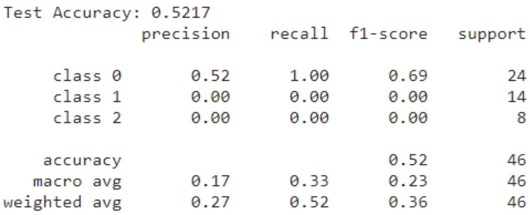
Fig. 1. Evaluation metrics for customSentimentModel using best hyperparameters on training data.

TABLE I
Validation Loss for finBERT Models

Model	Validation Accuracy
Baseline finBERT	0.361111
finBERT w/ training	0.427777
custom sentiment model	0.477777

The custom sentiment model tuned using the best parameters also performed decently on the testing dataset, indicating its ability to generalize. Once again compared to baseline BERT it's clear the significance of utilizing a carefully tailored model architecture.

This is not to say that the Custom Sentiment Model is perfect. The correlation matrix indicates that the model is currently labelling all the text data as neutral. This can be explained by the imbalance present in the dataset for neutral classes. With so many of the labels being neutral, when training the data on a small dataset such as the one used, these patterns may have gotten overlooked. To prevent this in the future a larger dataset should be used and the classes should be balanced properly, through undersampling of the majority class or oversampling of the minority classes.

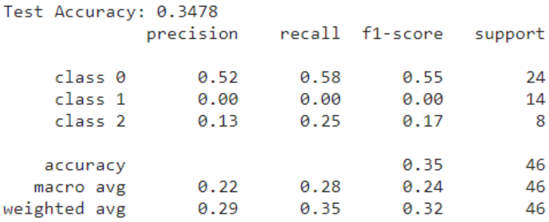


```
Test Accuracy: 0.5217
              precision    recall  f1-score   support

     class 0       0.52      1.00      0.69        24
     class 1       0.00      0.00      0.00        14
     class 2       0.00      0.00      0.00         8

    accuracy                           0.52        46
   macro avg       0.17      0.33      0.23        46
weighted avg       0.27      0.52      0.36        46
```

Fig. 2. Correlation Matrix for test data fit on customSentimentModel with the best hyperparameters.



```
Test Accuracy: 0.3478
              precision    recall  f1-score   support

     class 0       0.52      0.58      0.55        24
     class 1       0.00      0.00      0.00        14
     class 2       0.13      0.25      0.17         8

    accuracy                           0.35        46
   macro avg       0.22      0.28      0.24        46
weighted avg       0.29      0.35      0.32        46
```

Fig. 3. Correlation Matrix for test data fit on baseline FinBERT model.

Furthermore, this may indicate that using stock prices from the day of fillings are not good labels. For example, even looking at the word cloud of a misclassified text that was negative, words like revenue and increase are in bold, while words like loss and decrease are really small, or for a misclassified positive test most of the words used were neutral.

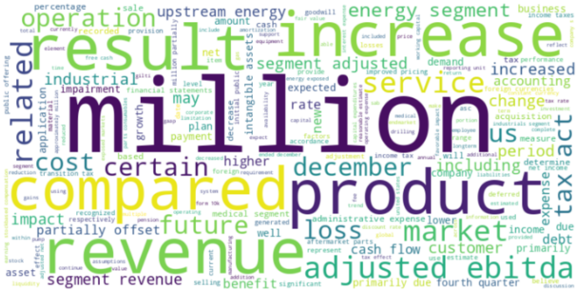
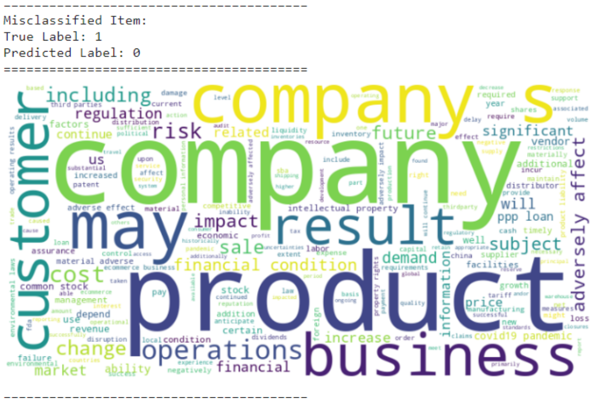


Fig. 4. Word Cloud of a negative filling mislabeled as neutral.

IV. Discussion

As seen in the results, it's evident that fine-tuning hyperparameters and building neural networks using pre-trained models, such as FinBERT, can greatly improve sentiment analysis. However, in order to capture the patterns of the text it's important that a balanced and large dataset is used. This comes with its own problems, as this may result in new GPU limitations and long run times.



```
----------------------------------------
Misclassified Item:
True Label: 1
Predicted Label: 0
========================================
```

```
----------------------------------------
```

Fig. 5. Word Cloud of a postive filling mislabeled as neutral.

Furthermore, one of the biggest challenges that come with running sentiment analysis on financial data is having labels to train and validate the model on. Since the ultimate goal of getting the sentiment of SEC fillings is to predict the stock value for said company, it seems reasonable to use historical stock prices. However, as seen through the results using immediate historical stock data doesn't present the impact SEC fillings have on the financial world correctly. It's known that stock values are extremely volatile, so there are many other factors that need to be taken into consideration.

Also, it's evident that reading one 10-k filling actually doesn't present much information on the company's financial outlook. Rather aggregating through multiple 10k fillings for a company and understanding a company's business model and weakness over time is the way to go when deciding what companies to invest in.

V. Conclusion

This study has demonstrated the significant impact of leveraging advanced techniques in sentiment analysis, specifically through the integration of neural networks built on top of pre-trained transformer models and the careful optimization of model parameters. By enhancing the FinBERT model with tailored architectures and fine-tuning, we have achieved notable improvements in sentiment analysis performance. The results reaffirm the potential of machine learning to uncover insights from complex financial text data that go beyond traditional methods.

However, our findings also underscore the importance of a holistic approach when analyzing the intricacies of 10-K filings. While sentiment analysis can offer valuable insights, it's clear that a comprehensive understanding of a company's business model and evolution over time, as encapsulated by multiple 10-K filings, is essential to make well-informed investment decisions. The individual snapshots provided by single filings might not capture the full spectrum of a company's financial outlook.

Looking ahead, the future of sentiment analysis within financial analysis holds exciting prospects. Developing summarization models tailored to financial text data, potentially by extending the capabilities of transformer pipelines, could yield concise yet informative representations of 10-K filings. This would enable analysts to grasp the core insights more efficiently, further enhancing decision-making processes.

Furthermore, an intriguing avenue for exploration involves the grouping of sentiment trends over time and comparing filings from different years using similarity scores. Such an approach could unveil evolving patterns and provide deeper insights into how companies' financial narratives change over periods, shedding light on trends, shifts, and potential indicators of financial health.

In essence, this study contributes to the growing body of work that highlights the synergy between cutting-edge natural language processing techniques and financial analysis. While we've made notable strides in sentiment analysis, there remains a wealth of untapped potential within the realm of 10-K filings and financial text data. By harnessing the power of NLP models, optimization strategies, and novel analytical approaches, we can unlock a more comprehensive and nuanced understanding of the complex world of financial reporting.

Acknowledgment

This project was completed for CS6120 under the instruction of Professor Uzair Ahmad.

References

[1] Loukas, L., Fergadiotis, M., Androutsopoulos, I., & Malakasiotis, P. (2021). Edgar-Corpus: Billions of tokens make the World Go Round. Proceedings of the Third Workshop on Economics and Natural Language Processing. https://doi.org/10.18653/v1/2021.econlp-1.2

[2] LOUGHRAN, T., & MCDONALD, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-KS. The Journal of Finance, 66(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

[3] Liu, Z., Huang, D., Huang, K., Li, Z., & Zhao, J. (2020). Finbert: A pre-trained financial language representation model for financial text mining. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2020/622

[4] Khan, A. (2023). Financial Reports SEC . JanosAudran/financial-reports-sec · Datasets at Hugging Face. https://huggingface.co/datasets/JanosAudran/financial-reports-sec