Financial Sentiment Analysis using Transfer Learning

Praveen Joseph, CFA
University of California, Berkeley
praveen.joseph@berkeley.edu

Lalit Vedula, PhD
University of California, Berkeley
lalitv1@berkeley.edu

Abstract

We propose a novel approach to financial sentiment analysis by adding non-domain specific labelled data and using transfer learning to achieve state of the art results. Sentiment analysis of financial text data is challenging because the language used in such documents is domain specific. Labeled datasets are not easily available in finance and even if available are usually proprietary and and expensive to acquire. We propose a novel transfer learning approach to improve model performance by adding a financial language model and IMDB sentiment engine on top of the general language model. Our results show that transfer learning improves the model performance considerably even for small data sets. We demonstrate that financial sentiment tasks using transfer learning on deep learning models is a highly performant approach to financial sentiment inference.

1 Introduction

The growth of financial texts in the wake of big data has escalated demand for analysis tools in financial services. Text streams are unstructured by nature, and more challenging to handle than numeric data streams. Unstructured text data provides fertile ground for deep learning which has produced state of the art results in NLP tasks using the transformer architecture (Devlin et al., 2018).

Sentiment Analysis is a natural language processing technique used to determine the polarity of unstructured data (text or speech). Since 2018, FinBERT models have burgeoned attempting to solve this problem and have produced impressive results. However, finance is a niche and secretive industry, with documents containing labelled data only accessible to highly-trained domain experts in select organizations. Previous studies have overcome this challenge of carefully annotated data by using unlabeled 10-K data for transfer learning (DeSola et al., 2019) or employing pre-processing techniques for using financial domain specific token embedding (Chan and Chong, 2017). One way to better understand fine tuning is by analogy to computer vision, where transfer learning had been widely applied prior to its crossover to NLP. In these deep learning models, the lower layers capture basic features such as edges and textures, while the higher layers depict more complete objects such as eyes, faces, legs, and dogs.

We propose a novel approach to financial sentiment analysis by adding non-domain labelled data (IMDB labelled data (Maas et al., 2011) to fine-tune the language model on sentiment tasks thus creating a sentiment engine on top of the transfer learned Financial language model. This will not only give our model Financial language understanding but also sentiment training to create "FinSen" - a BERT transformer model (Raffel et al., 2020) for sentiment analysis on financial data. We expect improvement in the model performance as more contextual information is added to the General Language model: General Language Model (BERT) → Financial Language Model (FinBERT) → IMDB Sentiment Engine → FinSen (Final Model).

We start with earnings calls data gathered from finnhub.io (to be used for academic research) and create a text file after parsing and cleaning the data for pretraining the financial language model. This will be followed by fine-tuning using 50,000 labelled data sets for IMDB movie reviews which will be used to train the sentiment engine (Maas et al., 2011). Our baseline model will be an out-of-the-box BERT model applied to sentiment analysis on financial data. If time permits, we will try and create a secondary benchmark using the Loughran-McDonald dictionary (Loughran and McDonald, 2011), widely used by the industry for financial text classification. We will use Cross-Entropy loss combined with F-1 and accuracy score as metrics to measure the performance of FinSen, which is an industry standard measure for financial sentiment classification tasks.

2 Background

Financial sentiment analysis can be used to determine investors' opinions of a company, stock, bond or any other financial asset. Sentiment may at times hint at future price action and nowhere is this imminently observable, than earnings calls where senior management and investment analysts discuss the health of the company. The signal-to-noise ratio in sentiment data is too low to meaningfully extract actionable price signal to trade upon. Therefore, stock sentiment alone cannot always predict changes in share prices, but when combined with tools such as technical analysis, a better understanding can be gained to determine possible scenarios. We can divide the recent efforts into two groups: 1) Machine learning methods with features extracted from text with ”word counting” 2) Deep learning methods, where text is represented by a sequence of embeddings. The former suffers from inability to represent the semantic information that results from a particular sequence of words, while the latter is often deemed as too ”data-hungry” as it learns a much higher number of parameters (Araci, 2019).

Financial sentiment analysis differs from general sentiment analysis not only in domain, but also the purpose. The purpose behind financial sentiment analysis is usually guessing how the markets will react with the information presented in the text. Sentiment indicators are typically used to determine whether a market is “bullish” or “bearish”. Loughran and McDonald (2016) presents a thorough survey of recent works on financial text analysis utilizing machine learning with ”bag-of words” approach or lexicon-based methods. For example, in Loughran and McDonald (2011), they create a dictionary of financial terms with assigned values such as ”positive” or ”uncertain” and measure the tone of a documents by counting words with a specific dictionary value. The Loughran and McDonald(LM) dictionary based model was considered baseline for financial sentiment tasks for over a decade but with the emergence of transformers and FinBERT models, we've seen new baselines which significantly outperform the standard bag-of-words approach based on LM financial lexicon.

The big breakthrough came in early 2018 when language modeling was combined with transfer learning. The idea behind transfer learning is to first “pre-train” a model on a large general-purpose dataset, then “fine tune” it on a smaller domain-specific dataset for a specialized task. (Vaswani et al., 2017) introduced self-attention and multi-headed attention concepts that laid the foundation for transformer models achieving high performance in large sequence data. Language models were able to develop significant levels of semantic awareness and become extremely useful for the pre-training stage of transfer learning.

In practice, fine tuning involves starting with a pre-trained language model (BERT) and swapping out the final layer, exchanging it for the specific building block that meets your needs. For example, if we want to do classification, we replace the final layer of the language model with a classifier head. We then retrain the model for the new task, adjusting the existing model's weights to incorporate learning from the fine-tuning dataset. The BERT language model can be further trained on unlabeled earnings call data which is abundantly available by unfreezing the 12-layers of BERT and allowing the model to learn financial vocabulary and domain specific semantic understanding. The classification head can be transfer learned using non-domain labelled data from a source such as IMDB which is also easily available. Now our financial language model understands both financial jargon and language polarity through the sentiment head - allowing the model to perform sentiment classification on unseen financial data. We can now fine-tune the model using a very small labelled financial data such a Financial Phrasebank (Malo et al., 2014). We are able to elegantly circumvent the lack of labelled data challenge through transfer learning and fine tuning.

One of the first papers that used deep learning methods for textual financial polarity analysis was (Kraus and Feuerriegel, 2017). They apply an LSTM neural network to ad-hoc company announcements to predict stock-market movements and show that method to be more accurate than traditional machine learning approaches. They find that pre-training the model on a large corpus improves the performance. However their pre-training is done on a labeled dataset, which is a more limiting approach than our proposed approach of pre-training on an unlabeled data set (which is more typical in the financial domain).

Due to lack of large labeled financial datasets, it is difficult to utilize neural networks to their full potential for sentiment analysis. Even when their first (word embedding) layers are initialized with pre-trained values, the challenge is in dealing with relatively small amount of labeled data. A more effective approach could be to initialize nearly the whole model with pre-trained values and fine-tuning those parameters with respect to the specific classification task.

3 Data

3.1 Training and Test data

1. Unlabeled earnings call data from finnhub.io: used for MLM on BERT
2. 50,000 labeled IMDB data: used for training the classification layer
3. Financial Phrasebank data with 4 levels of annotator agreement: used for fine-tuning and testing

3.2 Data Collection and Processing

To pre-train a FinBERT model, we scraped data from finnhub.io (licensed for academic research) to collect earnings call data and obtained 109,000 transcripts. After cleaning the data, we used BertTokenizer to pre-process the data and chunked it into lines containing no more than 510 tokens. This was done to ensure the maximum sequence length(MSL) does not exceed BERT's 512 input token limit including the special [CLS] and [SEP] tokens with additional [PAD] tokens included to ensure all input sentences have the same MSL = 512.

The classification layer uses 50,000 labelled reviews from IMDB movie reviews (Andrew and Gao, 2007) which will be used to train the sentiment engine. This transfer learning exercise is a key step in sentiment learning for our model.

For our labelled data, we use the Financial PhraseBank5 which is a public dataset for financial sentiment classification (Malo et al., 2014). The dataset contains 4846 sentences selected from financial news, on LexisNexis database, which are manually labeled by 16 annotators. Three of the annotators were researchers and 13 annotators were master's students at Aalto University School of Business with majors primarily in finance, accounting, and economics. The annotators were asked to give positive, neutral, or negative labels according to how they thought the information in the sentence might affect the stock price. The dataset also includes information regarding the agreement levels on sentences among annotators. Given the large number of overlapping annotations (5 to 8 annotations per sentence), 4 reference datasets, based on the strength of majority agreement, were provided:

1. sentences with 100% agreement [AllAgree]
2. sentences with $> 75\%$ agreement [75Agree]
3. sentences with $> 66\%$ agreement [66Agree]
4. sentences with $> 50\%$ agreement [50Agree]

3.3 Exploratory Data Analysis

The distribution sentiment labels for each agreement level can be seen in Fig. 1. We use an $80\%, 10\%, 10\%$ split to randomly divide the dataset into train, dev, and test sets.

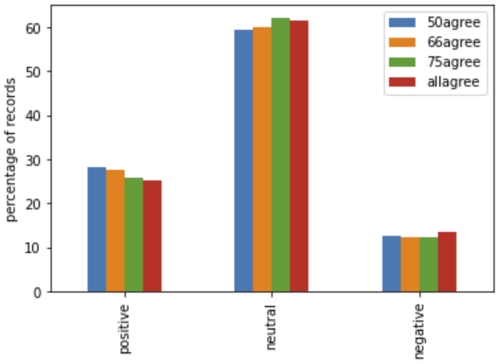


Figure 1: Distribution of classes in the datasets

From Fig. 1, we see that there is some imbalance in the dataset. We consider a couple of strategies to account for this imbalance:

1. Stratified sampling.

2. Custom loss function: the loss function for each label is scaled by the number of records for that label i.e.

$$CEloss = -\sum_{i=1}^{n_{labels}} \frac{CEloss_i}{n_i}, \quad i = 0, 1, \cdots n_{labels}$$

4 Methods

4.1 Modelling tasks

1. MLM to train BERT (110 Million parameters) using unlabeled earnings call data from finbub.io

2. Transfer learning classification layer using 50,000 labelled IMDB data

3. Fine-tuning the model using labelled Financial Phrasebank data

4. Predicting results using on the test data from Financial Phrasebank

We build on the BERTforSequenceClassification model using 'bert-base-uncased' for sentiment analysis. Our hypothesis is: since BERT is trained on general language, the performance of the sentiment classification will improve when we add financial language and a sentiment engine on top of the out-of-box base BERT model.

4.2 What is BERT?

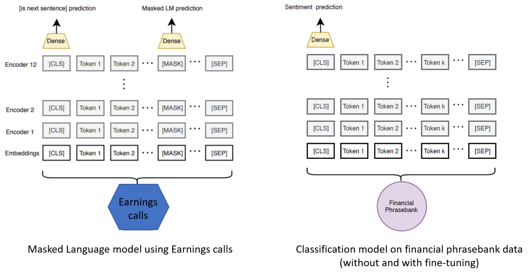
BERT (Bidirectional Encoder Representations from Transformers), released in late 2018, is a method of pre-training language representations that was used to create models that NLP practitioners can then use to either extract high quality language features from text data or fine-tune these models on a specific task (classification, entity recognition, question answering, etc.) with their data to produce improved model predictions. The model architecture is shown in Fig. 2

4.3 Modeling

We study the following models in this paper:

4.3.1 BERT, No fine-tuning

Our baseline model is BERT out-of-the-box with a classification task on the financial phrasebank.



Masked Language model using Earnings calls

Classification model on financial phrasebank data (without and with fine-tuning)

Figure 2: Model Architecture

4.3.2 BERT, With fine-tuning

Our baseline model is BERT out-of-the-box with a classification task using a portion of the financial phrasebank data for fine-tuning and the rest for testing.

4.3.3 BERT + IMDB Sentiment, No fine-tuning on financial phrasebank

We add a IMDB sentiment layer on top of BERT out-of-the-box, save the model, and perform a classification task using financial phrasebank as test.

4.3.4 BERT + IMDB Sentiment, With fine-tuning on financial phrasebank

We add a IMDB sentiment layer on top of BERT out-of-the-box, save the model, and fine-tune using some portion of financial phrasebank and use the rest as test.

4.3.5 FinBERT

We train a FinBERT (BERTforMaskedMLM from huggingface) from scratch using earnings calls data and then perform a classification task on financial phrasebank data set. This model trained on a financial corpus would be expected to understand financial language better and perform better on classification of financial sentiment (Fig. 3).

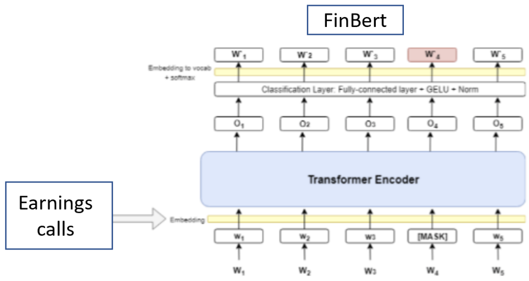


Figure 3: Pre-training FinBERT

4.4 FinBERT + IMDB Sentiment, No fine-tuning on financial phrasebank

We add a IMDB sentiment layer on top of FinBERT, save the model and then perform classification task using financial phrasebank as test (no fine-tuning)

4.4.1 FinBERT + IMDB Sentiment, With fine-tuning on financial phrasebank

We add a IMDB sentiment layer on top of FinBERT, save the model and then fine-tune using some portion of financial phrasebank and use the rest as test (training model to understand both financial jargon and sentiment analysis)

4.5 Evaluation Metrics

For evaluation of classification models, we used accuracy as our metric. We were using two frameworks (pytorch and TensorFlow) for our work and were getting erroneous results with custom functions written to extract metrics other than accuracy from TensorFlow). Since there is some imbalance in the dataset, for the 2 label dataset, it is $2:1$ and for the 3 label data, it is $4:1$, we will monitor the accuracy results for discrepancies.

4.6 Model Parameters

In our models, we use a dropout probability of p = 0.1, maximum sequence length of 128 tokens (based on Exploratory Data Analysis and model results), learning rate of $2e5$ and batch size of 32. We train the model for 4 epochs because it starts to over-fit around that number. We unfreeze all the layers during our experiments. The models were run using Tesla GPU on google colab pro. We were often able to obtain a Tesla V100-SXM2-16GB GPU which was highly performant and sufficient for the task at hand.

5 Results and discussion

The trend in training and validation loss as a function of epochs is shown in Fig. 4. Similar curves were observed for all the models, indicating minimal over-fitting.

All of our models use the 4 datasets described in section 3. Intuitively, we may expect that as the percentage of agreement among the 16 researchers who annotated the data increases, the model performance should improve since the data sets with higher agreement would have a higher polarity. This is seen in 1 for the 3 label classification task for the baseline model. However, there is another effect at play here that needs to be considered: the size of the dataset reduces as the percentage of annotator agreement increases. This results in non-monotonic behavior of the metrics in the 2 label classification task (note that the 2 label dataset has 40% of number of samples as the 3 label dataset).

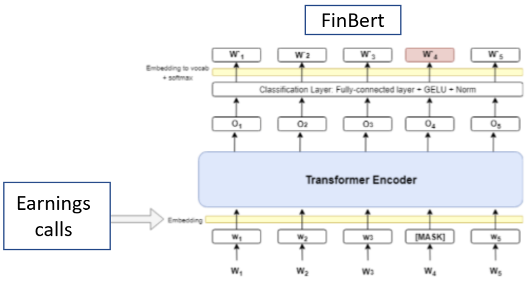


Figure 4: Training and Validation loss

The confusion matrix for the 3 label classification for the BERT base + fine tune model for the lowest agreement amongst the annotators, 50Agree (485 samples), and highest agreement amongst the annotators, AllAgree (227 samples), is shown in Figs. 5 and 6. [Convention: y-axis represents true label and x-axis represents predicted label]

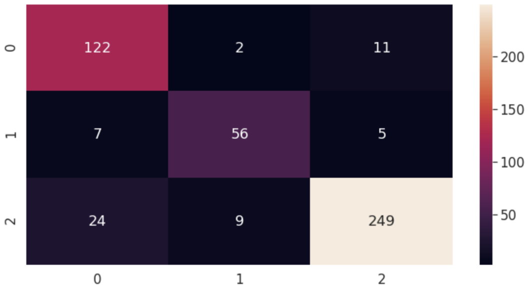


Figure 5: Confusion matrix for 3 label classification, BERT base + fine tune, 50Agree

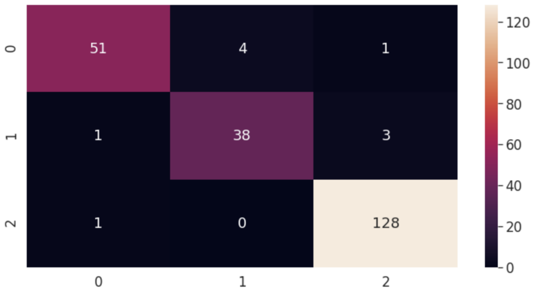


Figure 6: Confusion matrix for 3 label classification, BERT base + fine tune, AllAgree

We looked at some of the examples from the three label classification task which were classified

Table 1: accuracy, weighted precision, weighted recall, weighted f1 score, and Mathew Correlation Score for 3 label classification for BERT baseline model

Model	testdata size	accuracy	precision	recall	f1 score	MCC
AllAgree	227	96%	96%	96%	96%	92%
75Agree	346	95%	95%	95%	95%	91%
66Agree	422	91%	91%	91%	91%	83%
50Agree	485	89%	88%	88%	88%	79%

incorrectly to perform some error analysis. For e.g., one of the incorrect classifications was due to the inability of the model to recognize that a ”lower loss” would be interpreted by a human as positive news whereas the model assigns ”negative” sentiment after encountering the word loss in the sentence. In addition, we see that the positive statements being classified as negative and vice-versa require expertise and nuanced understanding of the context. Further, the dataset consists of single sentences and more information before and/or after the sentence in question would likely affect both the human labeling and model predictions.

1. Pre-tax loss totaled euro 0.3 million, compared to a loss of euro 2.2 million in the first quarter of 2005. **True Label**: Positive, **Predicted Label**: Negative.

2. IT services firm TietoEnator was bucking the general trend, holding flat at 22.70 eur, after slipping back from earlier gains. **True Label**: Negative, **Predicted Label**: Positive.

3. Nokia also noted the average selling price of handsets declined during the period, though its mobile phone profit margin rose to more than 22 percent from 13 percent in the year-ago quarter. **True Label**: Positive, **Predicted Label**: Negative.

For the three label classification task, a lot of the statements that are considered neutral are classified correctly. The most confusion in this use case is between positive and neutral labels. This makes intuitive sense since the examples where the different annotators had disagreement likely have lower polarity making it harder to distinguish between classes. Also, it is easier to differentiate between positive and negative but it can be challenging to determine whether a statement is neutral or positive (or) neutral or negative.

1. In the end, Sanoma News wants to secure its foundation with the savings. **True Label**: Neutral, **Predicted Label**: Positive.

2. According to Deputy MD Pekka Silvennoinen the aim is double turnover over the next three years **True Label**: Positive **Predicted Label**: Neutral.

Since the labelled IMDB dataset with three labels had only about 2000 examples, we chose to work with 2 label classification task for the rest of the paper since the labelled IMDB dataset for this case had 50,000 examples. The confusion matrix for the 3 label classification for the BERT base + fine tune model for the lowest agreement amongst the annotators, 50Agree (200 samples), and highest agreement amongst the annotators, AllAgree (72 samples), is shown in Figs. 7 and 8.

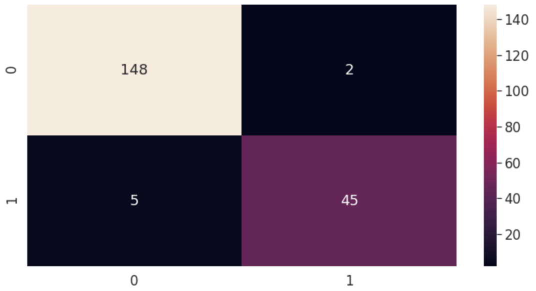


Figure 7: Confusion matrix for 2 label classification, BERT base + fine tune, 50Agree

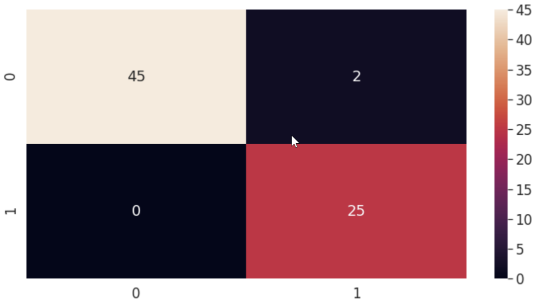


Figure 8: Confusion matrix for 2 label classification, BERT base + fine tune, AllAgree

Table 2: accuracy, weighted precision, weighted recall, weighted f1 score, and Mathew Correlation Score for 2 label classification for BERT baseline model (with fine-tuning)

Model	testdata size	accuracy	precision	recall	f1 score	MCC
AllAgree	72	97%	97%	97%	97%	94%
75Agree	134	95%	95%	95%	95%	87%
66Agree	173	98%	98%	98%	98%	95%
50Agree	200	96%	96%	96%	96%	91%

For the AllAgree dataset, the accuracy of the model using out-of-the-box BERT is 87.3%. Further fine-tuning the out-of-the-box BERT model with IMDB sentiment data without any fine-tuning using financial phrasebank data (i.e., we make predictions directly on financial phrasebank data) the accuracy increases to 90%. Finally, by fine-tuning the out-of-the-box BERT model with IMDB sentiment data and further fine-tuning using a portion of the financial phrasebank data, we get an accuracy of 94%.

Some observations on the results in table 2:

1. For the AllAgree dataset, BERT base, with no fine-tuning, shows a low accuracy of 64%. Adding a sentiment engine on top of BERT base, with no fine-tuning, improves the accuracy to 76.2%. Further fine-tuning this model using a portion of the financial phrasebank data leads to a significant increase in accuracy of 98.5%. This shows the ability of transfer learning to improve model performance even with relatively small datasets.

2. Similar trends in accuracy behavior hold for other datasets.

3. As the agreement between annotators increases, we would expect a higher polarity in the dataset. At the same time, the size of the dataset decreases. Because of the competing effects, this results in a lower accuracy for the AllAgree dataset compared to the 50Agree dataset when using BERT base, no fine-tuning.

4. Using the FinBERT model, no fine-tuning, gives about 4% higher accuracy than FinSen, no fine-tuning. Thus, a model trained on domain specific language on a smaller corpus performs better than a general language model trained on a much larger corpus which is further enhanced by adding a sentiment engine on top. The next step would be to look at the results from FinBERT, with fine-tuning. We could not complete this step at the time of writing this report but will continue to pursue this approach.

6 Conclusions and Future Work

In this paper, we demonstrated state of the art results achieved on financial sentiment analysis. Our novel approach of adding non-domain labelled data and using transfer learning allowed the language model to learn both financial jargon and sentiment classification tasks and apply them together to achieve greater than 95% accuracy in our financial domain test dataset.

We are extremely grateful to the NLP community for the prolific amount of research and the open source philosophy that has been widely adopted. The publicly available datasets, pre-trained models and shared code base allowed us to avoid hundreds of hours of pre-training and focus on the research problem we were most interested in. We will aim to continue the tradition and make our model, data and research results publicly available for others to build upon.

We believe there is still scope for future work in this project, one particular area is with pre-training of FinBERT. We trained the financial language model finBERT using 20.7 million token whereas BERT was trained on 3.2 Billion tokens, this allowed 'BERT with transfer learning' to outperform 'FinBERT with transfer learning' on the financial phrasebank test dataset. We intentionally limited the amount of earnings call data to keep training time and compute cost limited for the scope of this paper, since we wanted to specifically leverage the transfer learning component to achieve outstanding results. In future work, we would look to add additional 10-K and larger earnings call corpus to enable further pre-training improvements in FinBERT.

Table 3: Accuracy score for 2 label classification (FinSen = BERT base + Sentiment, FT = fine-tuning)

Model	testdata size	BERT base, no FT	FinBERT, no FT	FinSen, no FT	FinSen, with FT
AllAgree	873	64.0%	81.7%	76.2%	98.5%
75Agree	1307	66.3%	82.2%	77.6%	98.2%
66Agree	1682	67.2%	83.4%	78.7%	97.8%
50Agree	1967	67.1%	85.6%	78.7%	96.5%

References

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models.

W.K. Chan and Mickey W.C. Chong. 2017. Sentiment analysis in financial texts. *Journal of Decision Support Systems*, 94:53–64.

Vinicio DeSola, Kevin Hanna, and Pri Nonis. 2019. Finbert: pre-trained model on sec filings for financial natural language tasks.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Journal of the Association for Computing Machinery*, (1).

Mathias Kraus and Stefan Feuerriegel. 2017. Decision support from financial disclosures with deep neural networks and transfer learning. *CoRR*, abs/1710.03954.

Tim Loughran and Bill McDonald. 2011. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *Journal of Finance*, (1):35–65.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.