National College of Ireland

Can Textual Data from SEC Filings be used to predict the Directional Movement of Company Stock Price

MSc Research Project
MSCDA

Mairead O'Doherty
Student ID: 20172826

School of Computing
National College of Ireland

Supervisor: Mohammed Hasanuzzaman

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Mairead O'Doherty
Student ID:	20172826
Programme:	MSCDA
Year:	2022
Module:	MSc Research Project
Supervisor:	Mohammed Hasanuzzaman
Submission Due Date:	15/12/2022
Project Title:	Can Textual Data from SEC Filings be used to predict the Directional Movement of Company Stock Price
Word Count:	5986
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	1st February 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Can Textual Data from SEC Filings be used to predict the Directional Movement of Company Stock Price

Mairead O'Doherty
20172826

Abstract

Each year large companies are required to submit written information about their performance to investors. This text data provides an insider view of a companies trajectory and is often the first indication of future success or failure. These filings are large and complex documents, the goal of this work was to test if advancements in Natural Language Processing would enable a decision support tool that would predict the direction the stock price is expected to move in after the filing is published. The Longformer transformer model is tested against a basic XGBoost Classifier model to see if the advanced attention mechanism of the transformer model can capture the complexity of the text data.

1 Introduction

Publicly traded companies are required by the Securities and Exchange Commission (SEC) to disclose relevant information regarding their business operations via regulatory filings. Two such filings are the SEC 10Q and SEC 10K forms which must be filed on a quarterly and annual basis respectively. The purpose of these filings are to provide investors with information that they can compare to previous periods and use to evaluate outlook for the corresponding stock performance.[1] The filings provide analysts with a snapshot of the company's financial positions and includes information on financial statements, management discussions and analysis, disclosures and internal controls (Burke and Gunny; 2022).

Under the efficient market hypothesis it is asserted that asset prices adapt to new information entering the market (Ţiţan; 2015). Regulatory disclosures, such as those made in 10Q and 10K forms constitute new information and; therefore, may trigger the subsequent movements of stock prices. As a result of this, these company disclosures can aid in the process of financial decision making. These insights are used to inform speculative forecasting by financial investors and analysts before trading stocks (Li and Ramesh; 2009). This work usually involves humans correctly interpreting the content and taking clues from the text, a feat which has proved difficult for algorithms to mimic. A successful market decision support tool in this case will work to quantify if the financial disclosure conveys positive or negative content and as such can stock prices be expected to surge or decrease. This insight could then be used to aid the analyst in making trading decisions.

[1] https://www.sec.gov/

There have been links made between textual data and stock price movement in several fields. Media and internet postings have both proved to be useful in predicting stock prices. There tends to be high sentiment in these sources, for example a negative media coverage of a CEO's fraud accusation. There is also expected to be a quick effect in the market as the discussion is accessible to a large audience and a narrative has been established. The textual data available in SEC filings have a different linguistic style and tone as they are corporate, legal documents. Despite this, they contain important information as they are written by entities within the corporations and are often the first disclosure of important financial information external to the company.

There are several features of SEC filings which make them a challenging text source to draw predictive power from. They are long documents with the average length of a 10K Form being around 40,000 words in 2017 (Lesmy et al.; 2019). There is a financial benefit to the writers to highlight information that puts the company in a good light and downplay information which might cause worry in the markets, although this is moderated by legal considerations. This is a level of complexity not seen in news or internet sources. A possible method to deal with these complexities is deep learning especially transformer models, which accounts for many of the shortcomings of traditional Natural Language Processing (NLP) models.

A transformer model is a type of deep learning model that is defined by its self-attention mechanism. It processed information sequentially, allowing the model to take some consideration for the context words appear in and weighing differently the significance of each part of the text. This allows for more parallelization than other methods such as Recurrent Neural Networks (RNNs). However, the best results in NLP come from large language models such as BERT which are trained on millions of documents and require huge computational power. As this style of model creation is out of reach for all but the largest and best funded research teams in academia and industry, transfer learning is frequently employed to utilize pre-trained large language models and adapt their predictive power to individualized use cases.

Accordingly, this work sets out to address the following research questions: Can transfer learning on pre-trained transformer models improve predictions on directional stock movements using SEC filings data when compared to other traditional modelling techniques.

2 Related Work

Stock price prediction is a highly dynamic, non-linear and complex system based on a multitude of inter-correlated factors such as company profitability, media coverage, politics and wider economic factors. This set of factors influence how stock either rise or fall in valuation and make the prediction of such a noisy and volatile response variable, a difficult task.

2.1 Decision Support Tools in Stock Price Prediction

When an individual invests in the stock market either as an individual investor or as part of an institutional investment system, an in-depth knowledge of finance and stock market

dynamics helps to direct decision making. Decision support tools help an investor analyse large amounts of information about a company and the environment it is operating in, facilitating improved decision making. There has been a number of published attempts to predict stock prices, with the Machine Learning (ML) research falling into two broad categories - models which use time series data to predict stock price based on technical indicators and models which use other features to access the health of the company overall and predict possible stock price movements from this information. Using NLP features falls into the latter category of research (Patalay and Bandlamudi; 2021).

A number of these systems have been proposed in the past with varying degrees of success. Boonpeng and Jeatrakul (2016) used numerical data taken from the Stock Exchange of Thailand to predict an output class of buying, holding and selling stocks on a daily basis using three different neural network configurations. An accuracy of 72.5% was achieved using a optimized approximation algorithm neural network. This provides a strong indicator for which a potential investor can base their financial decision making. Similarly Xu et al. (2018) used Yahoo finance data and convolutional neural networks (CNNs), along with long-short term memory (LSTM) models to predict the directionality of stock price on a daily basis with an accuracy of up to 66%. Work by Malagrino et al. (2018) used a Bayesian network approach to trial stock price prediction across 12 indices worldwide using two timeframes, 24 and 48 hours with the accuracies ranging from 63% to 78%. These examples are all based on prior data from the stock markets that they are predicting within and seek to predict directional stock movement - if the stock rises, falls or stays the same during a given time period. This is useful as a decision support tool as the typical investor or analyst must make a decision to hold, sell or buy individual stocks to strengthen the portfolio.

2.2 The Link between Textual Data and Price

The results discussed thus far are based on time-series, numerical data taken from stock markets, however there has been extensive work done on how textual data can be used as input to stock market predictions. This text data comes primarily from three sources - news articles, corporate filings and online sources such as discussion forums and social media posts.

Social media and discussion forums are a readily available source of textual data which captures public discourse around a company, news events and the economy at large. This information can contain predictive signal around stock price movement. Social media data usually reflects personal opinions about a company or stock and tend to data sources suitable for sentiment analysis. In work by Schnaubelt et al. (2020) tweets are collected which mention stocks being traded in the SP500. The rapid pace of tweets is linked with minute to minute trade direction. The work depends on indicator features which are the presence of words that indicate a particular direction the stock price is moving in e.g., "increase" or "beats expectations" or words that indicate the relevance of a tweet such as "profit" or "sale". Meta features are also used which encode information about the Tweet other than its textual information e.g., account followers. The models are evaluated on risk-return metrics from a trading strategy employed from the model results are improvements are found to be statistically significant. Similar work by Coelho et al. (2019) is based on data from stock discussions forum StockTwits but relies

on the text data without feature extraction using the Term Frequency-Inverse Document Frequency, sentiment analysis and metadata as model inputs, with top accuracies being recorded at 67% in some model experiments. Work by Coyne et al. (2017) also focuses on data from StockTwits but looks at the degree of change between TFIDF vector groups and builds it's prediction on that analysis. This model did not perform much above their prior baseline and this is likely due to the use of a linear regression model with input which was likely complex and non-linear. The results were improved by 15 points when the dataset was limited to high value users using the user's meta-data information. The specific users chosen were evidently providing more signal than the general population using the service. These experiments show that textual data can be a source of signal for price movement in the stock market. However data discussed so far is highly emotive and reactive content which is likely steered by news sources. The public perception is what is contributing to a stock price changes, rather than tangible information about the status of the company which are coming from insiders or expert analysts.

News articles are also used to predict stock price movements, work by Shynkevich et al. (2015) tried to do just this using data from the Reuters US news archive during the year 2011. This paper outlined several interesting benchmarks used to compare against their model. The single direction which predicts stock price moving up or down depending on prior behavior of the stock. The random walk benchmark predicts whatever happened in one period will happen in the next, which over time converges to 50% and a technical analysis model based solely on the time series data. The overall goal was to classify the directional change of the stocks either upwards or downwards.Against the above benchmarks the Naive Bayes model using Latent Dirichlet Allocation (LDA) topic modeling as features. It was found that this approach did not exceed the benchmarks performance in closing price but did in the volatility of stocks being sold. This suggests that a company being in the news causes stock to be sold and bought in larger quantities but does not influence pricing in a particular direction. In work by Shynkevich et al. (2015) directional accuracy and return percentage was calculated using several varieties of support vector machines (SVMs) and a k-Nearest Neighbours (kNN) model. Accuracy as high as 76% were reported with best results achieved using a polynomial SVM model. Experiments in this study also siloed stocks into industry and sector specific buckets and predicted directional stock movements for these groups. These results were some of the higher in accuracy, pointing towards news stories being particularly influential at a macro-scale. A point of note in this work was that only stocks which had appeared heavily in the news were included in the study. This means the effect of just being in the news vs. not appearing was not considered. Work by Gite et al. (2021) used only the headlines of news articles as input data, applying sentiment analysis and LSTM models for prediction of directional stock price. This work reported a very high accuracy of 89% which has been met by scepticism by other authors but gives an indication of how successful such strategies can be.

2.3 The Connection between Corporate Filings and Stock Price

10K and 10Q filings are similar to social media posts and news articles in that they can inform perspective investors about the status of the company in question. However, there are significant stylistic differences between the data sources - SEC filings are legalistic and corporate documents, very long and exact in their wording. Despite this, they are

regarded as an essential tool for investing (Chang et al.; 1983). Natural language processing techniques became popular to try and speed up the processing of these documents and avoid bias when making decisions based on their contents. Initial work in this space relied heavily on features such as the length and readability of the documents as it was hypothesised that document writers would try to obfuscate the unflattering information in complex language (Li and Ramesh; 2009). Work in the field then went on to try and account for changes in the linguistical tone of the filings and find a correlation with stock prices. Another technique which is well known in the space is the use of word lists pioneered by Loughran and McDonald (2011) who used specifically compiled lists of technical words to pre-process filings for stock prediction. This is a relatively simplistic method when compared with modern natural language techniques.

There is a lack of work where machine learning models are used to predict stock prices from SEC filings in the literature. Some notable instances include work by (Doucette and Cohen; 2015) who used Dynamic Markov Compression to predict how stock price would change year-to-year, using 10K filings as input data. This work provided a dataset as part of their contributions, which spanned the years 1995 to 2010 but by the authors admission was flawed due to poor data collection in years before the stock market was digitized fully. The model was accessed based on returns from the trading strategy and showed some promising performance around the 2008 financial crises with authors noting they would expect an improvement as data quality and size increased.

Deep Learning often provides the best results in terms of NLP models, with their extended complexity allowing increasingly elaborate signal to be captured. Despite this they are rarely used to predict stock price from textual sources (Fataliyev et al.; 2021). There are examples with textual sources other than SEC Filings such as the aforementioned work by (Xu et al.; 2018) and similar work by (Kraus and Feuerriegel; 2017) which uses news filings and transfer learning to provide promising results. Deep learning seems to be underutilized in the areas of financial support tools as found in recent reviews Khadjeh Nassirtoussi et al. (2014) Ravi and Ravi (2015). This suggests that there is an area of research is underdeveloped, especially when using corporate filings as this review could not find any work that combined corporate filings and deep learning techniques.

2.4 Long Document Processing

SEC Filings are classified in NLP as 'long documents' meaning that single documents are on average longer than the input sizes of most of the benchmark pretrained models. RNN, LSTM and Transformer models are all trained with a set input size, with a notable example being BERT with input size of 512 words. Despite this there are several techniques which can be used to process long documents. Recurrence over BERT (RoBERT) and Transformer over BERT (ToBERT) models proposed by Pappagari et al. (2019) use the BERT model to create textual representations of the text in 512 word segments and then use reecurrent LSTM or transformer models to perfrom the classification. This technique is most useful on small, domain specific classification problems.

Another approach is to use extractive text summarization, this involves using a model to extract the most relevant piece of information in the long document and using this summarized document as the basis for model building. Extractive text summarization

is a field which is focused on condensing the large amount of textual information which has become available on the internet such as scientific articles, news stories and stock market analysis. There are two forms, abstractive and extractive summarization, extractive focuses on prioritizing the text based on the most important segment, paragraph or sentence, while abstractive generates new text to provide a summary. Extractive summarization can take place via unsupervised techniques such as graph based approaches (Mihalcea and Tarau; 2004), fuzzy-logic approaches such seen in work by Mihalcea and Tarau (2008) or by measuring difference between 'concept' document repositories and the text of interest (Gudakahriz and Mahmoudi; 2022).

The most accessible method from the above is the Text Rank algorithm which uses graph based sentence ranking methodology to determine which are the most relevant sentences in a document (Yadav et al.; 2021). The algorithm constructs a graph from the document with graph sentences represented by the vertices and the edges representing the overlap between sentences. This overlap is calculated by determining how many words two sentences have in common. The Pagerank algorithm then ranks the sentences based on relevance, a cut-off can be manually set to extract the most important sentences of the text and create a summary (Mihalcea and Tarau; 2004).

Work by Park et al. (2022) compared transformer models to determine which modeling techniques performed well on long documents. It compared a number of techniques, including text summarization techniques coupled with the BERT model. The Longformer model performed best, closely followed by TextRank/BERT model combinations. The Longformer model can ingest much larger strings than most pre-trained transformer models, it relies on a modified attention mechanism with a drop-in replacement rather than self-attention mechanism that combines a local attention window with a global attention mechanism (Beltagy et al.; 2020). These methods outperformed hierarchical systems like ToBERT and the newly published CogLTX method (Ding et al.; 2020) on a number of common classification datasets. Results of the work by Park et al. (2022) showed that much of there is a lack of datasets that fall into the 'long document' category and so the authors were not entirely confident in the ranking of transformer methods for classification.

3 Methodology

3.1 Dataset Preparation

This work requires that a new dataset be created as there was no suitable publicly available dataset although the data used is open source. The first step of this process is to select companies to examine in this work. The companies in the SP500 for the ten years from 2010 to 2020 were manually reviewed and any outlying companies were removed. This could happen because a company was only temporarily in the SP500 or were acquired by another company during this time. This resulted in a list of 493 companies. Once the list was compiled, company's corresponding 'tickers' were found. Tickers are an abbreviation of a company name which are used to identify publicly traded shares of a stock on the market.

Once the list of tickers was compiled, the corresponding filings that each company

provided to the SEC could be found. This involved querying the SEC filings API called EDGAR to initially get a list of all URLs associated with the filings each company provided during the stated timeframe. The filings chosen for this work are the 10-K and 10-Q annual and quarterly filings. The corresponding text could be queried using the URL link and the text could be downloaded from specific sections of the data. The sections chosen were "Financial Statements", "Risk Factors", "Market Risk" and "Management Discussions". The reason to select specific sections of the filings were based on their content being thematically most linked to stock price movement and to select only textual data. It also reduced the amount of text that was necessary for the model to process.

The response variable was then created. Previous work has used a number of different methods to a categorize directional stock price movement however there does not seem to be a standardised method used across the work summarized. Some work uses only Up and Down directional metrics (Shynkevich et al.; 2015) and some work splits data based on a buy vs ignore metric (Doucette and Cohen; 2015) with a cut off between over performing and normal/under-preforming stocks. This work does not frequently go into how these cut-offs are established so it is likely that it is based on subject matter expertise. Based on discussion with stock analysts, an up, down and stay metric is used in this work. With the stay metric being defined as any price which moves less than 1% during the time frame under consideration. The time frame chosen for this work is a 30 day period, most time frames for social media data are in daily blocks Coelho et al. (2019), while news media tends to require longer periods of days or weeks to require show an impact in the stock market. The period of time it takes new information to permeate the market and cause a price change will depend how widely the information is disseminated and how accessible it is. 10-k and 10-Q filings are lengthy documents which would take several hours to read in their entirety and require specialist knowledge to successfully part meaning from. Previous work by Doucette and Cohen (2015) used 10-K filing's data to predict stock price movement on a yearly basis but did not rely on solely textual data. The years taken into consideration were 2010 to 2020, this timeframe was chosen to ensure a large enough dataset while avoiding the data quality issues reported by Doucette and Cohen (2015) in their 2000 - 2008 dataset prior to all stock prices being automatically digitized.

$$TypicalPrice = \frac{HighPrice + LowPrice + ClosingPrice}{3}$$

$$VolumeWeightedAverage = \frac{TypicalPrice * Volume}{CumulativeVolume}$$

Stock price data was downloaded using the yfinance python package which provides access to stock data from yahoo finance data sets. All available daily stock price data for the companies during the timeframe was collected, including opening/closing price, volume of stocks traded and daily highs and lows. The volume weighted average price (VWAP) was used to get a reliable average value for each days price between all of these figures as is common practice when comparing stock prices over time. This figure was then compared to the same figure 30 days in advance and the difference between the two values was compared. Creating the categorical Up, Down, Stay response variable. This

response variable was then compared via publishing dates to the 10-k and 10-Q textual data as can be seen in Figure 1. The final dataset contained 9614 individual rows.

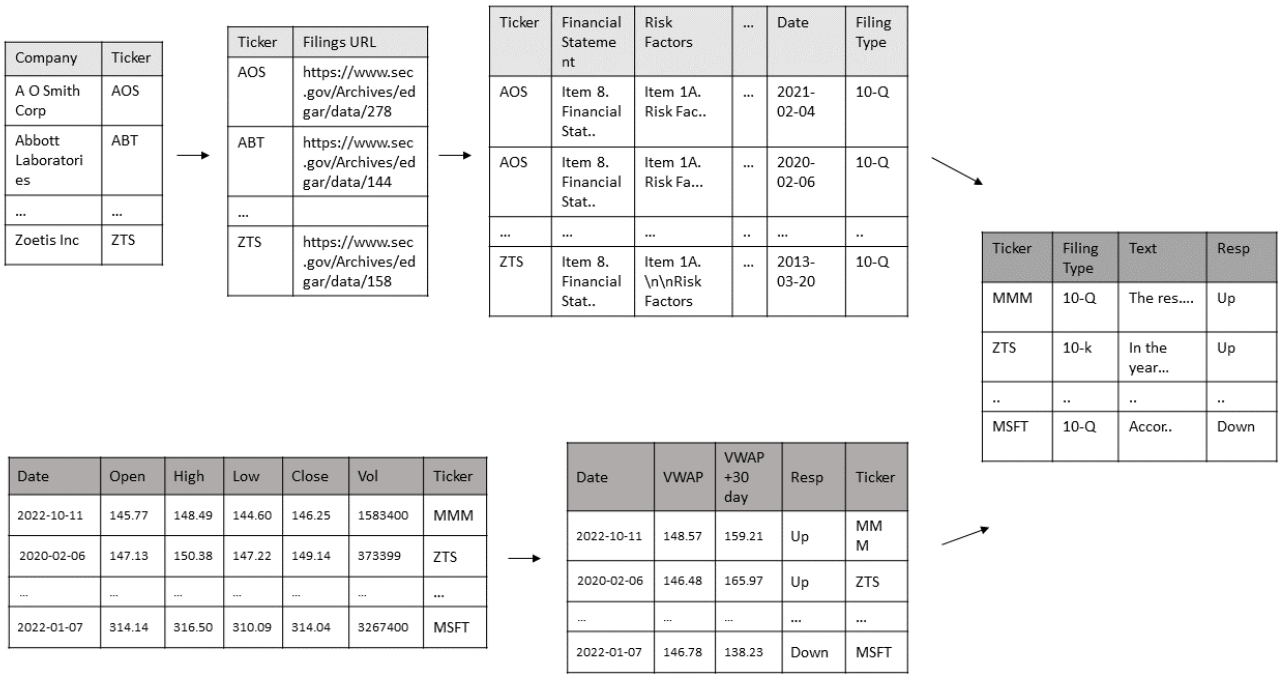

Figure 1: Formulation of the final dataset using SEC filings text and stock price data

3.2 Text Cleaning

Once the data set has been prepared, the text needed to be cleaned in order to make it suitable for modeling and to remove unnecessary noise from the data set. The four section of text that were extracted from the documents were concatenated together to create one text block from each filing. A number of steps to clean the text then took place, this included removing newline tabs, numbers and stopwords. Stopwords were also removed from the text.

3.3 Modeling

The average length of the text extracted from the filings was around XX in length. This is too long for most deep learning models extract meaning from, because of this, TextRank text summarization is employed to rank the sentences based on relevance and allows a cut-off to be established to reduce text length. The TextRank algorithm relies on the vectorization of sentence, in this case Glove embeddings were used and the output summed on a sentence level to create a sentence vector. The data was split on sentence basis and further cleaning of all other punctuation was applied. Each sentence is compared with all others in the document and the similarity is stored in a matrix representation. This similarity matrix is then converted into a graph representation with sentences as vertices

and similarity scores at the edges, then similar to Googles' PageRank algorithm the based on the probability of certain words occurring together. This was a time-intensive process with the application of the PageRank algorithm requiring several days to be applied to the entire dataset. Once the sentences in each document were ranked based on relevance, a cut off of 30 sentences was applied to bring the text length to a more manageable level. The new average text length after summarization and cut off was XX.

Once the TextRank algorithm was completed, the initial goal was to benchmark results using a traditional NLP method. A term frequency–inverse document frequency TFIDF vectorization approach was taken as it does not require a word limit and is not influenced by the length of the text. Instead it focuses on how important a word is to a document in a corpus. An XGBoost model was used to perform the categorization with accuracy, precision, recall and F1 score as evaluation metrics. This process was carried out before and after the TextRank algorithm was applied to measure if predictive power was reduced by the reduction in data.

A longformer model was trained to measure the effect a transformer model would have on the prediction and determine if an improvement could be established over the baseline model.The longformer model uses pre-trained tokenizer from the BERT model and weights from the pre-trained model to produce a result on new data. The longformer model was created in response to the limitations of other transformer models with respect to sequence length that the model can 'attend' to. The attention mechanism used by transformers such as the BERT model has a quadratically increasing memory and time consumption which means the pre-trained models have a limit of 512 words per sequence. There are some techniques such as the RoBERTa method which stream this sequence length in chunks but are limited to a small multiple of the usual sequence length. Due to the size of the documents in this dataset, even after the summarization technique has been applied a longformer model was the best transformer candidate for this problem. The global and sliding window attention mechanism employed by the longformer model, allows the memory and time consumption to scale linearly with window size. This allows the maximum window size of the longformer model to 4096 words which is more appropriate for the the longer text seen in this problem.

4 Design Specification

This project was carried out using the python programming language and the Google Colab environment was used in cases were advanced GPU power was needed. The TextRank algorithm used in this work relies on Glove embeddings from Standford NLP [2]. The cosign similarity was used to calculate the similarity between the embeddings and was implemented using the Numpy [3] package to improve computational speed based on the below:

$$CosignSimilarity = \cos\theta = \frac{A \cdot B}{|A|\,|B|}$$

[2]http://nlp.stanford.edu/

[3]https://pypi.org/project/numpy/

The NetworkX package [4] was used to construct the graph matrix from the similarity matrix constructed in numpy. The sentences were ranked according to this graph matrix and ordered in descending rank.

The distributed boosted gradient model XGBoost [5] was chosen as a baseline model to compare the transformer model as it has proven sucess in NLP classification tasks (Ullah et al.; 2021). TFIDF encoding was chosen as it has a vocabulary limit which allows the most impactful words to be chosen and represented by the vector while cutting out more common words with less signal.

The longformer models used pretrained BERT tokenization techniques from the huggingface library and was initialised with pretrained weights from longformer model. The model formatting is changed from the standard provided by the library to account for a multilabel classification task. This takes a respresentation of the text from a pooled output layer of the base model and attaches a classification head and changes the cross entropy loss function to a log loss function appropriate for the problem at hand. The model was created using the Pytorch, HuggingFace Transformers and Wandb packages. The PyTorch package [6] provided the infrastructure to modify the models final layers, the HuggingFace Transformer package [7] contained the BERT tokeniser and the pre-trained weights for the longformer model and the wandb package [8] provided dashboarding infrastructure to monitor model training. The model took roughly three hours to train on a NVIDIA-SMI GPU on the Google Colab cloud computing platform.

5 Implementation

The major output of this work is the final dataset used in the modeling. It consisted of 9614 rows with 493 individual companies almost evenly split between 10Q and 10K filings. Some 10Q filings were removed as there was no textual data in the sections considered in this work. The code that produced this data could be easily altered to provide variation in the years that the filings were collected from and change the time after publishing that a change in stock price should be seen in. The creation of the dataset was done on a personal laptop with an AMD Ryzen 7 Processor and 8GB RAM. It was compiled and cleaned using the python programming language from open source data repositories accessed via API. Similar ready to use datasets are not openly available as mentioned in work by Doucette and Cohen (2015) and a major benefit of this work will be the availability of this dataset and the code to modify it depending on timeframes of interest. While the data is openly available it took several days to download fully due to the constraints of the API rate limit.

The data was cleaned by removing numbers, extra whitespace, and textual contraction. Cleaning specific to this source of data was carried out with headings and numerical codes removed. Punctuation was removed after sentences had been split into lists for the TextRank process. The TextRank process was carried on the dataset and the data was split

[4] https://networkx.org/
[5] https://xgboost.readthedocs.io/
[6] https://pytorch.org/
[7] https://huggingface.co/
[8] https://pypi.org/project/wandb/

into train 66% and test 33%. The response variable was then reformatted depending on each models requirements using label encoding. The XGBoost Classifier model was run on a CPU with a learning rate of 0.3 and a max depth of 6. The longformer model was trained via transfer learning using the pretrained weights. The learning rate was 0.00002, with a total batch size of 36 and 6 gradient accumulation steps which resulted in 512 optimization steps. Due to the computational needs of the model training, higher batch size or gradient accumulation would have exceeded the GPU capabilities of the Google Colab environment.

6 Evaluation

The goal of this work to provide a decision support tool that could be used by stock market analysts and allow them to make decisions on portfolios without dedicating large amounts of time to reading the multiple filings that are published by companies each year. This does not require the prediction of an exact stock price but a general sense of how the company is performing based on an insiders take of company performance. The Up, Down, Stay categorical output was designed to give this decision support in a consise way. As a result of this output the evaluation metrics that will be used to evaluate success will be overall accuracy and precision, recall and F1 scores for each individual category.

6.1 TextRank Summarization

It is difficult to evaluate the success of a text summarization technique without a labeled dataset from which to extract metrics. This problem cannot provide a metric for success, however the TextRank algorithm has shown success in similar problems Park et al. (2022). When the summarized documents were manually reviewed they seemed to have adequately summarized important information from the longer filings text. The following is the sentences deemed most important in the first long document that was summarized in the dataset and give a strong indication of the points made in the longer document.

key fiscal measures of our performance for the second quarter and first half of are summarized below complemented with prior-year acquisition activity average second-quarter production of thousand barrels of oil equivalent per day set a new record for the company and represents an increase of percent from second-quarter....

6.2 XGBoost - Baseline Model

There were several reasons a baseline model was required in this work. The first was after the dataset has been established - the presence of predictive signal needed to be established before further work was carried out. The second reason was to measure any signal loss that took place before and after the TextRank summarization took place. When TFIDF + XGBoost classifier models were used on the raw, long-form text the accuracy was 56%. Considering this was a three category problem, this was considered reasonable result to determine that the text contained signal. This dropped a little over 1% after text summarization to 54.5%. Which suggested that most of the signal in the text was retained during the TextRank process. Full results are below in Table 1. The classes for this data are unbalanced, measures to address this haven't been trialled in this experiment but have the possibility of improving results. Best results were seen with the largest

class, stocks which have gone down in price. The 70% F1 score and the 95% recall are particularly encouraging results here suggesting the model is unlikely to give false negatives.

Table 1: Results of XGBoost classifier with TFIDF encoding post TextRank summary

	precision	recall	f1-score	support
Down	0.55	0.95	0.70	1469
Stay	0.50	0.00	0.01	345
Up	0.40	0.07	0.12	855
Accuracy			0.54	2669

The poor results seen in recall in the "Stay" category suggest there is likely a better way to categorize stock price movement as the model struggles with the 1% Stay category for stock prices which do not move over the time period.A suggested solution here could be to normalize the stock price, tying it to the overall performance of the market on the day in question. This might improve results by highlighting stock that may have over or under-performed by bucking the market trends on a particular day.

6.3 Longformer - Transformer Model

Surprisingly the longformer model performed worse than the baseline model with an overall accuracy of 35.7%. There a number of reasons that could account for this drop in performance. One of the main benefits of a transformer model is that is can process text sequentially. This allows the model to take the context of words into account rather than treating them as individual data points. In most benchmark NLP problems, this ability has shown marked improvement over other non-transformer methods. It is possible that the TextRank sentence ranking process removed some of the flow of the text and hampered the main benefit of this type of model. It is also possible that the type of data that the longformer model is trained on wikipedia data, which varies significantly in tone and contents from corporate legal filings causing the discrepency between expectations and the final results. Full results are in Table 2

Table 2: Results of longformer classifier

	precision	recall	f1-score	support
Down	0.52	0.46	0.49	1469
Stay	0.11	0.22	0.15	345
Up	0.30	0.23	0.26	855
Accuracy			0.36	2669

The model does not seem to be harnessing much predictive power at all with reduction in Down and Up categories and a small improvement in the Stay category, however this is such a small improvement that is likely just noise. The training parameters chosen for the

longformer model were chosen to relieve some of the compute pressure, with other hyper-parameters causing the system to crash during training. Despite this, it is not likely that hyper-parameter changes would improve results. The loss value during training dipped slightly during training and then leveled out. Suggesting the model was not picking up much signal during training.

7 Conclusion and Future Work

The goal of the project was to improve on the baseline model using the Longformer transformer model architecture. The XGBoost baseline model unexpectedly outperformed the transformer model. In the future it would be prudent to place more emphasis on the XGBoost classifier model with TFIDF text encoding. The limitations outlined in the evaluation section give some possible reasons for this.

The novel contribution of this project, aside from the data set creation, is that the methodology of using the TextRank summarization along with the Longformer transformer model were not able to capture the signal from the data as well as the baseline despite the success of transformer models in other benchmark tasks. A methodology of this type has not been trialled previously on SEC Filings data in order to predict stock movement. Although this experiment was not successful, negative findings are useful in that they help define the boundaries of methodologies that do not result in improved outcomes. In the future, there is hope that another researcher may have more success building on these finding and provide a tool to enable more efficient stock analysis.

Improvements could be seen by increasing the amount of data used to train the model. The data also has a class imbalance, and stratified sample could improve performance overall. There was also not much emphasis placed on feature creation in this work which could be beneficial in the future.

A contribution from this work is the creation of a dataset which collected textual data from filings for ten years along with the associated stock price movement. This will be made available for further work. It can also be easily and quickly modified to show stock movement for a different timeframe after filings published. An interesting line of inquiry would be to retry the baseline model at different timeframes after filing to determine how long it would take to see the affect in the stock price.

The major limitations of the data set are that the set timeframe for price movement, the fact the data was compiled through an API and that not all text available was included. The response variable in the data was calculated for the price movement between the date the filing was published and 30 days after that date. This can be easily modified using the code provided in the project however for the dataset as a stand-alone asset, the timeframe is fixed.

The data was compiled using the EDGAR and Yahoo Finance APIs. As a result, the compiled data was not manually checked for accuracy and missing data. The completeness and validity of the data is dependent upon the quality of the APIs and cannot be assured by the work in this project.

The dataset was created by selecting four major and relevant sections of the filings; however the entirety of the filing could not be captured in the dataset due to limitations in storage and compute capabilities.

References

Beltagy, I., Peters, M. E. and Cohan, A. (2020). Longformer: The long-document transformer.

Boonpeng, S. and Jeatrakul, P. (2016). Decision support system for investing in stock market by using oaa-neural network, *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)* pp. 1–6.

Burke, J. and Gunny, K. (2022). Sec comment letters and 10-k accounting and linguistic reporting complexity, *Journal of Accounting, Auditing Finance* **39**(3): 1653–1688.

Chang, L. S., Most, K. S. and Brain, C. W. (1983). The utility of annual reports: An international study, *Journal of International Business Studies* **14**(1): 63–84.

Coelho, J., D'almeida, D., Coyne, S., Gilkerson, N., Mills, K. and Madiraju, P. (2019). Social media and forecasting stock price change, *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* **2**: 195–200.

Coyne, S., Madiraju, P. and Coelho, J. (2017). Forecasting stock prices using social media analysis, *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* pp. 1031–1038.

Ding, M., Zhou, C., Yang, H. and Tang, J. (2020). Cogltx: Applying bert to long texts, *Advances in Neural Information Processing Systems* **33**: 12792–12804.

Doucette, J. A. and Cohen, R. (2015). Content of annual reports as a predictor for long term stock price movements, *FLAIRS: The International Florida Artificial Intelligence Research Society Conference* pp. 416–421.

Fataliyev, K., Chivukula, A., Prasad, M. and Liu, W. (2021). Stock market analysis with text data: A review.

Gite, S., Khatavkar, H., Kotecha, K., Srivastava, S., Maheshwari, P. and Pandey, N. (2021). Explainable stock prices prediction from financial news articles using sentiment analysis, *PeerJ Computer Science* **7**: e340.

Gudakahriz, S.J., a. M. A. and Mahmoudi, F. (2022). Opinion texts summarization based on texts concepts with multi-objective pruning approach, *The Journal of Supercomputing* .

Khadjeh Nassirtoussi, A., Aghabozorgi, S., Ying Wah, T. and Ngo, D. (2014). Text mining for market prediction: A systematic review, *Expert Systems with Applications* **41**(16): 7653–7670.

Kraus, M. and Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning, *Decision Support Systems* **104**: 38–48.

Lesmy, D., Muchnik, L. and Mugerman, Y. (2019). Doyoureadme? temporal trends in the language complexity of financial reporting, *SSRN Electronic Journal* .

Li, E. X. and Ramesh, K. (2009). Market reaction surrounding the filing of periodic sec reports, *The Accounting Review* **84**(4): 1171–1208.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks, *The Journal of Finance* **66**(1): 35–65.

Malagrino, L. S., Roman, N. T. and Monteiro, A. M. (2018). Forecasting stock market index daily direction: A bayesian network approach, *Expert Systems with Applications* **105**: 11–22.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* pp. 404–411.

Mihalcea, R. and Tarau, P. (2008). Optimizing text summarization based on fuzzy logic, *Seventh IEEEIACIS International Conference on Computer and Information Science* pp. 347–352.

Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y. and Dehak, N. (2019). Hierarchical transformers for long document classification, *CoRR* .

Park, H. H., Vyas, Y. and Shah, K. (2022). Efficient classification of long documents using transformers, *The Association for Computational Linguistics 2022* .

Patalay, S. and Bandlamudi, M. (2021). Decision support system for stock portfolio selection using artificial intelligence and machine learning, *Ingénierie des systèmes d information* **26**: 87–93.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, *Knowledge-Based Systems* **89**: 14–46.

Schnaubelt, M., Fischer, T. G. and Krauss, C. (2020). Separating the signal from the noise – financial machine learning for twitter, *Journal of Economic Dynamics and Control* **114**.

Shynkevich, Y., Mcginnity, T., Coleman, S. and Belatreche, A. (2015). Predicting stock price movements based on different categories of news articles, *2015 IEEE Symposium on Computational Intelligence for Financial Engineering Economics* pp. 703–710.

Ullah, H., Ahmad, B., Sana, I., Sattar, A., Khan, A., Akbar, S. and Asghar, M. Z. (2021). Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews, *CAAI Trans. Intell. Technol.* **6**: 251–264.

Xu, B., Zhang, D., Zhang, S., Li, H. and Lin, H. (2018). Stock market trend prediction using recurrent convolutional neural networks, *Natural Language Processing and Chinese Computing* pp. 166–177.

Yadav, A., Maurya, A. K., Ranvijay, R. and Yadav, R. (2021). Extractive text summarization using recent approaches: A survey, *Ingénierie des systèmes d information* **26**: 109–121.

Ţiţan, A. G. (2015). The efficient market hypothesis: Review of specialized literature and empirical research, *Procedia Economics and Finance* **32**: 442–449.