Forecasting Stock Price Movements from Company Disclosures

Neslihan Dinçel
STUDENT NUMBER: 2058858

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:
Dr. Afra Alishahi
Dr. Peter Hendrix

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2021

Preface

I would like to thank my supervisor professor Afra Alishahi and my second reader Peter Hendrix for their patience and valuable feedback. Furthermore, I would like to thank my friends and housemates for their incredible emotional support during my master's year.

Forecasting Stock Price Movements from Company Disclosures

Neslihan Dinçel

In this thesis, official company disclosures (8-K filings) are analyzed to make predictions about stock price movements. The relationship between textual content and stock prices has been an attractive research topic among researchers and industry analysts. Text data ranging from social media posts to news and official company disclosures are used as input in these analyses. Complex nature of the stock market makes stock price prediction one of the most challenging prediction tasks. Competent works in literature are based not only on text analysis, but use hundreds of variables as features, text representations being only one of them. For the purposes of this analysis, market volatility index, market returns, company characteristics such as size and industry, and texts coming from disclosures are used as inputs to build a predictive model. To control for the systematic market risk and isolate the effect of the disclosures, stock returns are normalized by the expected stock returns. Combinations of features, texts, and taking into consideration the systematic risk are the distinguishing characteristics of the analysis. Findings confirm the challenging nature of the stock price prediction task. Nevertheless, majority baseline model is surpassed with two models tested. Some of the existing research uses long periods of data for training, while including dozens of financial ratios and indicators in the model. Considering the limited time and resources, a 28% increase in accuracy from majority baseline model can be interpreted as a good starting point. Adding new relevant variables, and increasing the amount of data have improved the performance, suggesting that further increase in both may continue to improve the performance.

1. Introduction

Efficient market hypothesis suggests that financial markets are perfectly liquid, and information is accessed instantly and equally by utility-maximizing rational investors (Malkiel and Fama 1970).

In reality,

i) markets are illiquid from time to time depending on macroeconomic cycles, and transaction costs introduce frictions to the market,

ii) investors do not have access to the information equally and at the same pace, and lack necessary skills to interpret the financial data, average investor significantly underperforming the market index (Barber and Odean 2002)

iii) behavioral finance studies have repeatedly shown that investors deviate from rational behavior (Barber and Odean 2002; Baker and Nofsinger 2002). As a result, there is room for profit for those who identify factors contributing in price changes correctly and in a timely matter.

Forecasting using accounting data (fundamental analysis) and forecasting using historical prices (technical analysis) have been widely used in predicting stock prices. Predictive powers of these models vary across sample periods, indicating that success of out-of-sample predictive models is a rare outcome due to "lucky" sample period. Despite the better long-horizon performances, auto-regressive models which regress past returns on yields explain less than 5% of monthly or quarterly return variances (Fama and French 1998). Papers utilizing regression models typically report in-sample performance, which compare the model's predictions of training period to the actual prices in that period. A comprehensive review over regression-based forecasts shows that 46 out of 51 models fails the out-of-sample tests (Goyal and Welch 2004). Due to widespread adoption of these models as well as changing market dynamics, explanatory and predictive powers of accounting data on stock prices have diminished over time (Francis and Schipper 1999; Setiono Strong 1998). The predictive power of historical data on DJIA index (Dow Jones Industrial Average, an index consisting of weighted average stock prices of 30 largest industrial companies in the U.S.) is fluctuating substantially across different time periods (Qian 2005). Results of these studies indicate that variables used in these models are not sufficient in explaining the variation in prices. With the developments in text mining and natural language processing, the ability to use text as the new variable to make predictions with current information and qualitative data has become possible, making text analysis a trending topic in finance.

News coming from formal SEC filings (company disclosures made through U.S. Securities and Exchange Commission), company press releases, secondary news, micro blogging and social media posts have a potential to be associated with buy and sell decisions, thus with price movements. Based on the hypothesis that formal company disclosures mandated by U.S. Securities and Exchange Commission are the most reliable primary news sources among these, the research questions of this thesis are:

1) Can we predict the direction of stock price movements after the announcements by analyzing company disclosures?

2) How does the prediction performance change after including other features such as company characteristics and volatility index in the model?

3) Which machine learning models and pre-processing steps yield better performance in capturing the relationship between company disclosures and stock prices?

RQ.1) The main research question is aiming to understand if there is a correlation between company disclosures and company's share price, after controlling for overall market change. Disclosing company's stock price is monitored within a time frame after the announcement, and the outcomes are labeled as "up" or "down". Each stock has a different Beta – covariance of the stock return relative to the market return, which implies that low beta stocks have less co-movement with the market than the high beta stocks. The overall market conditions and intrinsic characteristics of the individual stocks are taken into account during labeling, in the pursuit of isolating the effect of the announcement.

RQ.2) Stock prices correlate with multiple events and variables. Although it is not possible to include all stock-related and economic variables in this study, company size in terms of market capitalization, the industry company operates in, the stock exchange shares trade in, the security type, and market volatility measure VIX (volatility index by Chicago Board Exchange) is added as features to the model to see if there is a correlation between them and stock prices.

RQ.3) Machine learning model logistic regression, and neural network models Multi-layer Perceptron, LSTM and BERT are tested to understand which classifier works better for the problem at hand. LSTM and BERT are chosen to capture sequential information in the documents. All models are tested on both a separate document classification task using news articles, and stock price prediction task using 8-K forms. This allows for comparison and interpretation of the model behaviour in different tasks, which use inputs and outputs with different characteristics.

Findings suggest that we can predict the direction of stock prices by analysing event disclosures, and predictive performance does increase slightly with the inclusion of additional features. The details are discussed in the upcoming sections.

2. Related Work

The related work will be separated by the era and models used, however they differ in their scope and approach as well. Natural language based financial forecasting can typically take two forms: a) predicting the sentiment score of related texts and using these scores as input in predictive models, b) predicting change in stock prices as a direct result of document classification. Division by era and models shows that early studies use machine learning models such as random forest classifier and Naïve Bayes, while recent studies make use of deep learning models such as recurrent neural networks and transformers. Finally, more sophisticated document classification techniques are presented, which are proposed to overcome the memory problem in classifying long

documents. These models are not applied in financial domain yet, but have a potential to improve the representation of 8-K filings, which essentially are long documents.

2.1 Early Examples

Early examples of natural language based financial forecasting are dictionary-based sentiment analyses measuring the feeling a text conveys, based on pre-defined list of sentiment-ascribed words (dictionaries). These studies represent documents with sparse lexical features such as bag-of-words and n-gram methods, while using general-purpose dictionaries such as Diction and Harvard General Inquiry. AZFin text-based predictive system analyzing financial news articles, and Tetlock's analysis of Dow Jones Newswire and Wall Street Journal news are examples of these studies (Schumaker and Chen 2006; Tetlock, Saar-Tsechansky, and Macskassy 2008). In 2011 Lougrhan and McDonald build upon these works by exploring the performance of Harvard's psychological and sociological dictionaries (Loughran and McDonald 2010). They find that using dictionaries not designed specifically for finance often results in misclassification. They create 5 financial sentiment dictionaries based on their tone (positive, negative, litigious, uncertain) and a master dictionary, which is 2of12inf dictionary extended with words from 10-K and 10-Q files (annual and quarterly reports released by companies). This master dictionary is later widely used for financial sentiment analysis. In addition to these, Lougrhan and McDonald weight rare and important words more, and penalize less important, abundant words in the documents using Tf-idf method. Both Tetlock and Loughran & McDonald use traditional sentiment analysis predicting sentiment scores, and find that negative and uncertain expressions have more effect on prices than positives expressions. Learning algorithms used to train the classifiers in these early studies are Naïve Bayes, SVM and SVR, and their accuracies are slightly better than random guess.

2.2 Deep Learning Era

More sophisticated natural language-based financial forecasting (NLFF) with complex representations and learning algorithms have emerged only in the last few years. The vector representations have evolved from word counts to pre-trained word embeddings such as GloVe and Word2Vec, which aim to capture the semantic relationship between words in sentences and documents.

Discovery of Transformer deep neural network models (Vaswani et al. 2017) for handling sequential data and long-term dependencies, followed by the introduction of transformer based BERT language model (Bidirectional Encoder Representations from Transformers) have greatly improved the feature extraction capabilities, while gated recurrent, LSTM, and convolutional neural networks have improved prediction performances significantly (Devlin et al. 2018; Hiew et al. 2019; Lee, Gao, and Tsai 2020). Comparison of logistic regression, doc2vec, LSTM and CNN over a dataset from StockTwits, - a social media platform for investors, shows that CNN outperforms other models (Sohangir et al. 2018). Here sentiment analysis is performed by classifying messages as "Bullish" or "Bearish". An investor is considered Bullish if he or she believes that the stock price will increase, thus recommends purchasing shares, and Bearish if she recommends otherwise. The problem with reporting sentiment prediction score is that predicted sentiment may be noisy, not reflecting the actual behavior of the investors. Thus, the results may not predict the stock prices after all. Lastly, FinBERT, a

domain-specific word embedding model derived by training BERT on financial corpus (Corporate 10-K & 10-Q Reports, Earnings Call Transcripts and Analyst Reports) has outperformed BERT model in several tasks (DeSola, Hanna, and Nonis 2019; Yang, UY, and Huang 2020).

2.3 Document Classification

Although transformer based language models have longer memory in comparison to recurrent neural networks, they fail to perform well on sequences longer than a few hundred words. Vanilla transformer models are trained on corpus split into fixed length segments. As a result, inter-segmental context is lost during training. "Transformer XL", introduces segment level recurrence to transformer architecture, which holds the segment hidden state in cache and reuses it as an extended context while processing next segments (Dai et al. 2019). "Hierarchical Transformers" proposes sequence segmentation to solve the memory problem. The documents are split into overlapping segments, and segment contexts are stacked and fed into LSTM network, which outputs final document embedding (Pappagari et al. 2019). Another solution to increase the memory capacity is "Hierarchical Attention Network" for document classification, which has two levels of encoding: word encoder creating word vectors, and sentence encoder creating sentence vectors. It has two levels of attention mechanisms applied at the word and sentence-level (Yang et al. 2016).

In summary, natural language-based financial forecasting literature review reveals that models aiming to predict a sentiment score, which measure sentiment from annotated texts have higher accuracies, while models aiming to directly predict stock price movements have accuracies of 55% - 60%. Vanilla BERT-based models have high performance on sentiment analysis on relatively short social media data, but are inapplicable to longer texts. Studies work on improving the predictive power of text based models with numerical features, or use language processing output as sentiment score feature in a multi input financial forecasting model (Zoen Git Hiew et al. 2019; Lee et al. 2014). Advanced forecasting models use multiple variables along with text data, "earnings surprise" being the most predictive feature. Earnings surprise is the difference between earnings at the time of disclosure, and the average of analyst earnings forecasts. Earnings is perceived as a proxy of company's overall health and permanence, and is referred to as a performance measure of its business activities. Having extensive resources to gather and interpret the information, earnings forecasts generated and published by financial institutions are valued and monitored closely by investors, who take position based on these forecasts. When the reality does not meet the expectations, investors reverse their position, which impacts the share price. Lee and Surdenau find in their analysis that earnings surprise feature is the most predictive feature in their feature set, and modify their random forest model to guarantee that it is included in all generated decision trees (Lee et al. 2014). They use the model with only earnings surprise as their baseline, and gradually add other non-linguistic features. Finally, they add 8-K documents in the analysis, which improves the performance of non-linguistic model by 10%. Due to amount of time and resources required to gather earnings surprise data, this variable is not used in the analysis.

3. Methodology

3 different models will be used in the analysis. These are logistic regression, LSTM network and BERT language model.

Logistic Regression

Logistic regression is a machine learning model which is derived by setting right hand side of the linear regression equation equal to the logit of probability (logit(p)), i.e. logarithm of odds (log(p/(1-p))) in a binary classification problem. Proposed by mathematician Verhulst, its origins date back to 1838. In order to leave the probability value p alone, Sigmoid function (inverse logit) is applied to logit(p). By this way, sigmoid maps the output of linear function to probability values ranging between [0,1]. Gradient descent algorithm is used to train the model and update the parameters. Using non-linear sigmoid function in mean squared error formula results in non a non-convex loss function. In order not to get stuck in local minimum, cross-entropy loss is used, which is a convex loss function combining two different states of true class. Logistic regression is a linear classifier, with a decision boundary of p = 0.5. It can be regularized by including a penalty term in the loss function. Multinomial logistic regression extends the model by minimizing the loss across the entire probability distribution. Logistic regression is a simple and fast algorithm that works well in text classification. The limitation of logistic regression is that it assumes independent variables are not multicollinear, which may not be the case for the text data (Kowsari and Heidarysafa 2019).

LSTM

Although recurrent neural networks can in practice model the sequential dependency, problems of vanishing and exploding gradients makes their training infeasible. Long short-term memory network is a type recurrent neural network, which solves this problem by a gradient-based algorithm in an architecture enforcing constant (neither exploding nor vanishing) error flow through internal states of special units (Hochreiter and Schmidhuber 1997). It also works well with long dependencies by preserving the past information, and regulating the information flow through input, cell memory, forget and output gates. One of the problems encountered when using recurrent neural networks is that they do not involve parallel operations such as convolution. Long-term information has to travel through all cells before arriving to the current processing cell. Although this is reduced by LSTM's gate mechanisms, linear operations within LSTM blocks require large amounts of memory bandwidth, hence the training time can be a major challenge in LSTM.

BERT

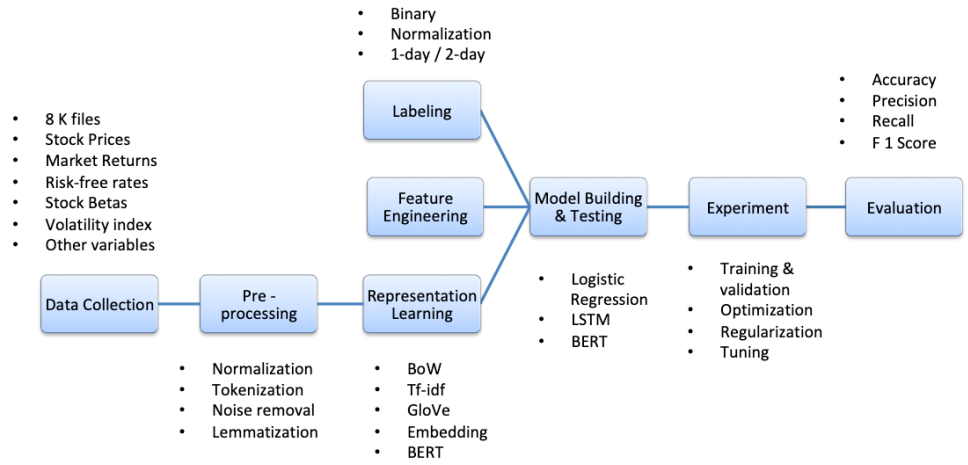
Bidirectional Encoder Representations from Transformers is a stack of pre-trained transformer encoders trained on unlabeled text by conditioning on both left hand right contexts. The pre-trained BERT model can be fine-tuned with an additional output layer to create state-of-the-art models for a wide range of tasks (Devlin et al. 2018). Two models proposed by the paper are BERT base (12 encoders stacked), and BERT large (24 encoders stacked), which are trained by performing masked language modeling and next sentence prediction tasks. In masked language modeling, a special MASK token is used to randomly corrupt the input sequence, after which model tries to predict

the masked words. This task trains the model to take into account and understand the context of the sentence. Next sentence prediction task involves predicting if two sentences follow each other or are unrelated. During training 50% of the time successive sentences are fed into the model, and 50% of the time random sentence pair is fed. This task trains the model to make sentence level prediction.

4. Experimental Setup

Experimental setup consists of data collection, pre-processing, labeling, feature engineering, feature extraction, training & validation & hyperparameter tuning, and evaluation on test set. The structure of the study is as follows:

Figure 1
Study Design



4.1 Data Collection

Multiple data types from different data sources are required to conduct the research. 8-K filings, company characteristics and economic indicators are used as features in the analysis, while stock prices, market returns, stock betas and risk free rates are used for labeling.

8-K Files

U.S. Securities and Exchange Commission requires that public companies disclose changes in material status on a current basis with 8-K filings, in addition to periodical annual and quarterly reports 10-K and 10-Q. Coming from the primary source, 8-K filings are considered to be reliable sources of news containing up-to-date information, since companies are required to disclose the material changes within 4 business days. Due to this time limit, 8-K reports are generally referred to as "current reports". The events disclosed through 8-K filings have material nature, and are therefore valuable to investors and shareholders in their decision-making. These events are grouped into 9 sections with sub-sections referred to as "items".

Example item content of an 8-K filing:

"Item 5.02. Departure of Directors or Certain Officers; Election of Directors; Appointment of Certain Officers; Compensatory Arrangements of Certain Officers On January 14, 2019, Ron Morrison, resigned as Executive Vice President and General Counsel of Impac Mortgage Holdings, Inc. (the "Company"). In connection with his departure, Mr. Morrison and the Company entered into a Separation and Release Agreement whereby the Company agreed to pay Mr. Morrison a severance payment of $300,000, permit the exercise of vested and outstanding options aggregating 132,700 with exercise prices ranging from $2.73 to $20.50 until the earlier of December 31, 2019 or the expiration date, and provide health benefits until December 31, 2019. Mr. Morrison and family members were also removed as beneficiaries to a life insurance policy for which the Company provides collateral."
https://www.sec.gov/Archives/edgar/data/1000298/000110465919002333/a19-3134_18k.htm

Financial services company FinnHub provides API to access the urls of SEC files from SEC Edgar database, where queries can be made using stock's trading symbol "Ticker", type of filing and date. For those who need bulk data, they have uploaded tables including SEC forms' url links from 1994-2021 in Kaggle. The url tables are downloaded from the page https://www.kaggle.com/finnhub/sec-filings as .csv files (kaggle.com).

Although 2020 data is more recent, to avoid distortions caused by massive stock market fluctuations during early stages of pandemic, the documents released in 2019 are preferred for the analysis. Several python libraries are tested to scrape the text content from the pages. The best method to scrape the data is using the remote browser controller Selenium WebDriver (The Selenium Browser Automation Project). Beautiful Soup 4.9.3 library is also tested for web scraping, and found incapable of handling the pages relying heavily on JavaScript (Richardson 2007).

During selenium web scraping, depending on the instantaneous change in Internet speed caused by fluctuations in connection, some pages do not open fully, thus their content cannot be scraped. These incidences do not exceed 5% of the requests. After multiple trials, it has been concluded that missingness in data take place completely at random, and do not affect the distribution of the documents. The document lengths and filing dates are uniformly distributed across the filings. The filings are independent events, and there is no correlation between documents. Therefore, web-scraping procedure of web driver is randomly sampling a data set for this study.

Market Returns, Stock Returns, Stock Betas

Nobel-price winning Capital Asset Pricing Model (CAPM) is used to calculate expected stock returns using excess market returns and stock betas (Sharpe 1964). CAPM is a linear model describing the relationship between the systematic risk of an asset and its return. Systematic risk is the market related risk, which cannot be diversified away, while unsystematic risk is the asset specific risk, which can be eliminated by diversification. Each asset is affected by the changes in market on a different scale. Therefore, each stock has a different systematic risk multiple (beta parameter in CAPM). Historical asset returns are regressed on market returns to find the systematic risk (beta parameter) of the asset.

CAPM equation:

$$E\left(R_i\right) = R_f + \beta_i\,\left(E\left(R_m\right) - R_f\right)$$

$E\left(R_i\right)$ = capital asset expected return
R_f = risk-free rate of interest
β_i = sensitivity
$E\left(R_m\right)$ = expected return of the market

$R_m - R_f$ – Excess Market Return:

Rm (market return) is the overall market return, calculated with weighted average returns of all stocks traded in NASDAQ, NYSE and AMEX. Rf (daily risk free rate) is calculated using 3-month treasury bill rate. Since daily T-Bill rates are close to zero in recent years, daily risk free rates are also zero. WRDS (Wharton Business School Data Services) provides the combined daily Rm-Rf referred to as excess market return ("exret") in percentage (WRDS 2021).

β – Beta:

Stock beta is the covariance of the stock movement relative to the market movement. It is the degree of correlation between stock return and market returns. All stocks combined have a beta of 1, which implies that a portfolio consisting of all stocks will have the same return as the market return, which is in fact calculated by combining all stocks. The idea is that low-beta stocks have less co-movement with the market than the high beta stocks. Large companies for instance have a beta close to 1, since they move more or less in parallel with the market. Utility and tobacco companies on the other hand have betas closer to zero, since they are not heavily affected by market movements. Finally, negative beta stocks move in the opposite direction of the market. WRDS provides daily stock betas as ratios.

$E(R)$ – Stock return:

Actual daily stock returns are compared with expected daily stock returns derived from the CAPM equation. The difference between the two is used for labeling. If daily stock return is higher than expected, the instance is labeled as positive, while if it is less than expected, it is labeled as negative. If the release time of the 8-K form is before 16:00, same day return is used. For disclosures released after 16:00 on the other hand, next business day's return is used.

VIX

"The VIX Index is the measure of constant, 30-day expected volatility of the U.S. stock market, derived from S&P 500® Index (SPX℠) call and put options. It is one of the most recognized measures of volatility – widely reported by financial media and closely followed by a variety of market participants as a daily market indicator." - (Cboe 2021)

The motivation behind using this variable is to include an indicator of the economic condition and financial outlook, as the stock prices are greatly affected by the external conjuncture.

Daily volatility index data is used for the same day for which the returns are taken, e.g. same day VIX for same day returns and next business day VIX for next business day returns. The daily VIX data is taken from Federal Reserve Economic Data (FRED 2021).

Additional features

Company characteristics such as market size and industry, stock characteristics such as exchange code, share type, and filing characteristics such as item numbers are used as additional features in the analysis. All data except for item numbers are from (WRDS 2021). Items are extracted from the documents and one-hot encoded for the analysis.

4.2 Pre-processing

Different types of pre-processing procedures are conducted for different models, as they require inputs in varying formats. First, and common step is to extract only the item contents, which start with the first item and stops before the signature. This way, all the headers, meta data, tables and addresses are removed from the data. Python 3.9.5 (Python.Org 2021) regular expression operations module is used to match the text between the first item and signature. Less than 5% of the data don't have a matching text, which is the randomly missing data coming from selenium web scraping.

Another regex pattern is used to extract the item numbers from the documents. These are later turned into item feature through one-hot encoding. Item information is then used as additional feature during classification. Punctuation (non-alphanumeric characters), new line characters, and multiple spaces are removed using regex library.

Longest document in the dataset has 68220 tokens. The distribution of documents in terms of number of tokens is as follows:

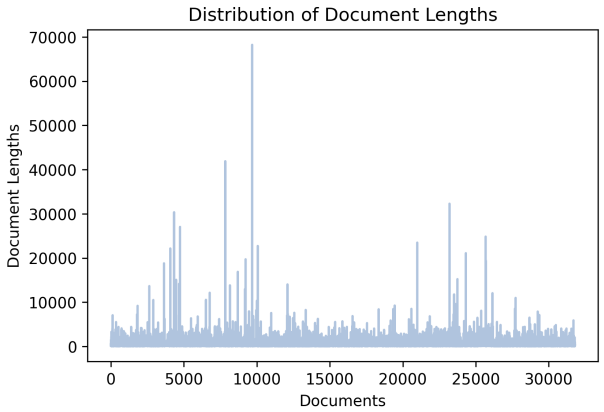


Figure 2
Document lengths

We can clearly see that long documents are outliers. These are excluded from the analysis for the following reasons:

1) Outliers are not representative of the population, distort the distribution of the documents, and have a potential to distort the outcomes.

2) Feature extraction and training of the models become computationally infeasible.

3) These documents contain too many "items" i.e. many news bundled together. Measuring the collective meaning of multiple news is challenging for a machine-learning model.

The document corresponding to 90th percentile of documents in terms of length has 962 tokens. The documents longer than this cut off value are removed from the data. The new distribution of the document lengths is close to uniform distribution. Mean document length of 90th percentile is 460 words.

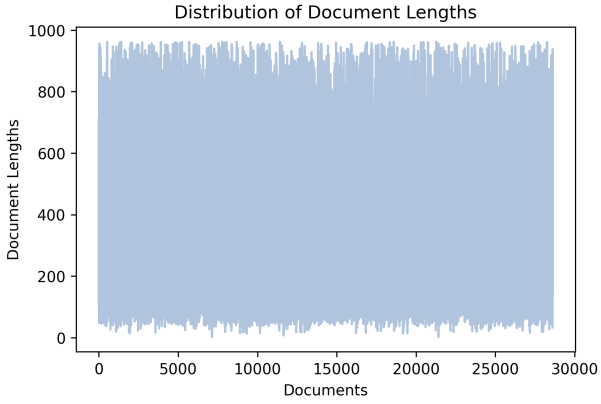


Figure 3
90th percentile document lengths

Logistic regression pre-processing

To prepare the texts for logistic regression model, words are lowercased and top words are removed using nltk library's (NLTK3.6.2 2021) list of English stop words. Special html characters, tags and punctuation is removed using regex library. Words with contractions are separated to capture negations. Words are lemmatized with nltk library, which uses WordNet lexical database. Lemmatization is chosen over stemming, as it considers the context, and transforms words into their closest base synonym. Documents are split into words and characters using nltk library's WordPunctTokenizer() function.

LSTM pre-processing

Pytorch (Facebook 2021) is used both for pre-processing and model building, since it has all the necessary classes to efficiently implement both tasks. Data is split into training and validation with stratification using sklearn train-test split (Scikit-Learn 2021). Pytorch's torchtext library handles the Out of Vocabulary words by replacing them with an Unknown token, and handles variable length input sequences through *packed padding sequence*, which pads the sequences into equal lengths, but ignores the padding token during output generation in the recurrent neural network. With the *field class*, the pre-processing pipeline is defined: tokenization with spacy tokenizer (Honnibal and Montani 2017), lowercasing, and arranging the iterator to have batch

as its first dimension. TabularDataset is used to access the raw training and test data using these fields as parameters. Vocabulary is built with training set texts and labels, using GloVe embeddings (Pennington, Socher, and Manning 2014).

In text only model, BucketIterator is used as batch iterator to group similar length sequences together. This minimizes the padding, which reduced the training time. In multi-input model, training and validation iterators are initiated separately for text embeddings and numerical features. Index is taken as reference point to keep the matching occurrences of text and features.

BERT pre-processing

BERT model requires different pre-processing steps to maximize the performance of the model. Transformers library by HuggingFace is used for both pre-processing and model building (Wolf et al. 2019). The item numbers and titles are removed manually, leaving only the actual content of disclosures. Punctuation is not removed, as BERT tokenizer replaces them with special tokens. Sentences longer than 30 words are removed, which causes loss of information, but increases accuracy. Since the objective is not Named Entity Recognition or Part-of-Speech tagging, "BERT uncased" is used, as suggested by Google (GitHub 2021) BERT uncased tokenizer lowercases the text before tokenization, converts whitespace characters to spaces, removes the accents, splits non-alphanumerical characters by adding whitespace on both sides, and uses WordPiece algorithm, which tokenizes the text into meaningful sub words maximizing the likelihood of the training data (PaperswithCode 2021).

Pre-processed and shortened text sequences are prepared for the model by tokenizing, padding and truncating the sequences to a maximum length of 150 tokens. 93% of the processed documents are below 150 words, 7% is truncated. Calling the tokenizer returns a dictionary with input ids, attention masks and input type ids. Input ids and attention masks, along with labels are wrapped together to create the data loader, which randomly samples batches.

4.3 Feature Engineering

Unique ticker list (stock symbols) are taken from the final training data. These symbols are used to make queries for stock data and market data from WRDS. Return column in market data is processed by removing the percentage sign from the end of return values, turning them from string to float, and from percentages to ratios.

Continuous Features

Some companies have more coverage and publicity, i.e. are covered by more analysts, investment companies and individual investors. They are expected to be affected less from the disclosures, since most of the information is already out there. Therefore, company size is chosen as a feature to be included in the analysis. Share price at the time of disclosure is multiplied with the number of shares outstanding (shares outstanding refers to all shares held by investors) to get the market capitalization, which is used as a proxy for company size. Volatility index (VIX) and market size are standardized using sklearn StandardScaler (Pedregosa et al. 2011). This ensures that data in different scales do not bias the model by influencing it disproportionately. The scaling is done by centering all data points with sample mean and dividing by sample standard deviation. Training and test sets are standardized separately, to avoid any information of test set to leak in training set.

Categorical Features

Categorical features used in the analysis are industry of the company, item numbers in disclosures, share type, and exchange code. Originally there are approximately 600 unique sectors in the dataset, however these sectors are subsectors nested in parent sectors. Typical sector code (NAICS) consists of six-digit numbers, starting from parent to child from left to right. To avoid the problem of "curse of dimensionality", subsectors are merged into parent sectors by taking first two digits of the codes. The sector numbers are looked up from NAICS (North American Industry Classification System) Association's official website (NAICS 2021), and distribution is plotted.

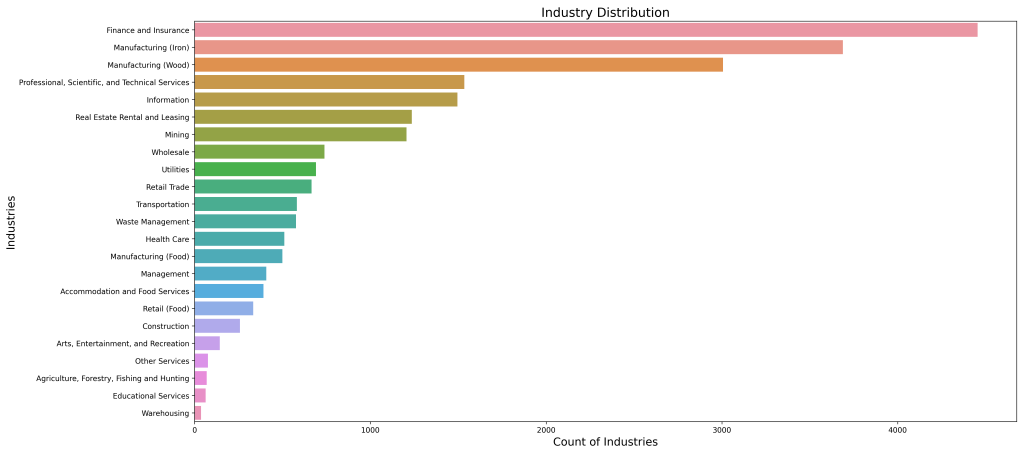


Figure 4
Distribution of Industries

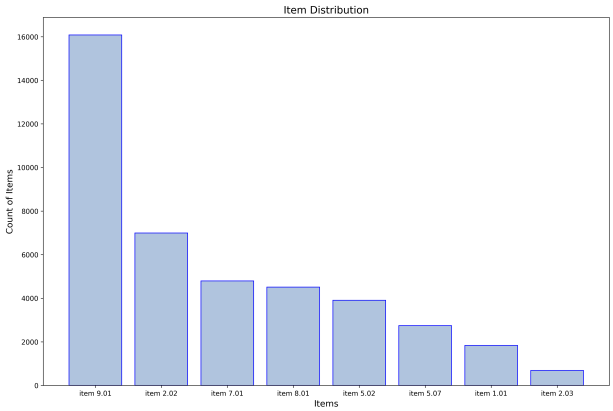


Figure 5
Distribution of Items

Rare items are excluded from the graph. Most occurring items from left to right are Financial Statements and Exhibits, Results of Operations and Financial Condition, Reg-

ulation FD Disclosure, Other Events, Departure of Directors or Certain Officers, and Submission of Matters to a Vote of Security Holder.

Three exchange codes in dataset represent NYSE, NASDAX and AMEX stock exchanges. There are 13 share codes, which indicate the type of share, such as common share, certificates, units etc. All categorical features are dummy coded, and their first columns are removed to exclude redundant “reference group” and reduce dimensionality.

4.4 Labeling

Expected returns are calculated through the CAPM formula, and instances are labeled according to stock return being smaller or larger than the expected return. If the stock price has gone up more than their expected change based on market change and their covariance, it is labeled as "positive", and "negative" otherwise. To see the time horizon the model predicts best, two different time intervals are tested, one of them taking the same day price change after the announcement, the other one taking the same day and next day cumulative price change (two-day price change).

4.5 Representation

Bag of Words

Word counts are used to vectorize the documents. In text-only model, no vector size limits are set in order to capture all the words. These vectors have the length of training set vocabulary. To capture some of the sequential information, Bag of n-grams method is used with n=3. In multi-input model with texts and features, vector size is limited to most important words, to have a balanced representation of both texts and numerical features. 4000 is chosen as length, after different maximum feature arguments are tested.

Tf-idf

From bag-of-ngram matrix, Tf-idf representations are created, which normalize the word counts (term frequency) with log transformed inverse document frequency. In multi input model, which uses texts and non-linguistic features as input, the representations are extracted directly from tokenized texts (BoW matrix now has a limited length). Tf-idf penalizes the words that appear in every document, which do not add a distinguishing value in classifying the documents, while giving more weight to words that are distinguishing.

GloVe

In LSTM model, pre-trained vector embedding Glove (Global Vectors) is used to represent the texts. Training set vocabulary is built with GloVe 6B 100d. GloVe is chosen to capture the global statistics along with local statistics. It is an unsupervised learning algorithm trained on Wikipedia and Gigawords, an English newswire archive, which captures the semantic relationship between words in a (V, V) shaped co-occurrence matrix, V being the vocabulary size. These representations are then passed into embedding layer in the neural network architecture.

BERT

While GloVe assigns single representation for words regardless of the context, Bidirectional Encoder Representations from Transformers (BERT) represent words based on the context. Through masked language modeling it aims to predict randomly masked words by looking at words before and after them, and through attention mechanism it understands the context. The first layer of the model built is pre-trained BERT-base uncased layer, which takes the texts as inputs and returns the hidden state of the classification token.

4.6 Model Building & Testing

Before the actual experiments, the models are tested on a document classification dataset. AG's news classification dataset is used to test the models (groups.di 2021). The dataset has 120 000 training examples consisting of 4 balanced categories. Categories "World" and "Business" are filtered and "Sports" and "Sci-Tech" are discarded to make the dataset contextually resemble the financial texts. The mean news length of filtered training set is 38 words, titles included.Three different combinations of model architectures and feature representations are tested on this dataset.

Logistic Regression

Sklearn linear model logistic regression is fit into both bag-of-ngram features and Tf-idf features, and results are evaluated. The function implements L2 regularization by default. The choice of penalty parameter, and parameter C, which specifies the strength of regularization are optimized using grid search with k-fold cross validation (GridSearchCV) where k = 5.

BERT

BERT-base uncased model is downloaded from Transformers library's Automodels. A model is built, first layer being the pre-trained BERT model, which outputs the state of the CLS token – a token added in the beginning of sequence as a *classification token*, followed by two fully connected layers with dropout in between. To use the pre-trained weights, parameters of the base model are *frozen*, i.e. not updated through back propagation. Softmax activation is used to make the model compatible with both 2-class classification and 3-class classification. Although the classes are fairly balanced in, class weights are passed to loss function to handle the class imbalance. Negative log-likelihood loss is applied after Softmax layer, which is equivalent to using Pytorch's CrossentropyLoss without softmax activation. In training, Adam optimizer with weight decay regularization (rather than classic L2 regularization) is used, as it yielded slightly better results.

LSTM

An LSTM network, which takes GloVe embeddings as inputs is implemented with Pytorch nn Module. Inside the model, GloVe representations are first passed through the embedding layer. Then for each batch, the documents are padded to equal lengths and packed together using pack_padded_sequences() and packed sequences are passed through bi-directional LSTM blocks with dropout regularization. Finally, last forward and backward hidden states are concatenated and fed to the fully connected layer. Bidirectional LSTM makes use of both preceding and follow up words in the sequence, by forward and backward prediction. Sigmoid is applied to the output of fully con-

nected layer as final activation. In multi input classification, numerical features are concatenated with the hidden states before the fully connected layer. Similar to the BERT model, Adam optimizer with weight decay is used during training. A diagram of model architecture combining textual and non-textual features is presented:

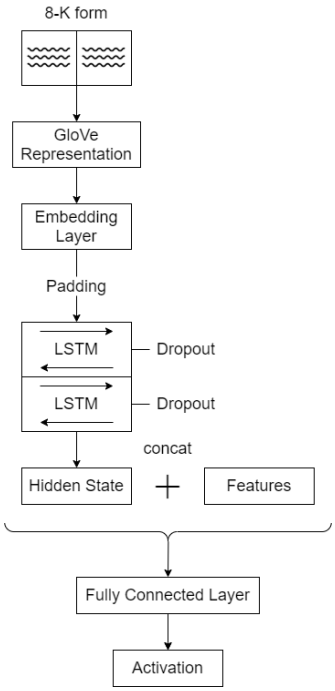


Figure 6
Multi-input LSTM model architecture

Model Testing

Before conducting the experiments, bag-of-n-gram features + logistic regression, Tf-idf n-gram features + logistic regression, GloVe embeddings + LSTM network, and BERT model are applied to AG news classification dataset.

The performance on AG dataset:

Table 1
Classification Results

Model	Accuracy
BoW Logistic Regression	0.92
Tf-idf Logistic Regression	0.92
GloVe LSTM	0.96
BERT	0.96

Logistic regression model has similar performances with Tf-idf and BoW features. LSTM and BERT models slightly outperform logistic regression. We can conclude that 60.000 training examples and short sequence lengths (mean = 38) enable the models

learn well. Models are validated through this testing process before the experimental stage.

4.7 Experiments

After confirming that models work, they are used on company disclosures and non-linguistic data in the experimental stage. To answer two research questions, three different experiments are conducted. 8-K filings are processed and labeled through the steps described in the methodology. A total of 17.361 data points are used for training and validation, and 4341 for testing.

Experiment 1: Prediction using 8-K filings as input

In this experiment, 8-K filings are processed and labeled through the steps described in the methodology. Logistic regression, LSTM, and BERT-based language model are trained and validated using these 8-K representations and labels (supervised learning). Motivation behind this experiment is to identify stand-alone value of 8-K files in predicting stock prices.

Experiment 2: Prediction using features as input

In this experiment, only the additional features market capitalization of the security, volatility index, security type, security exchange, industry of the registrant (disclosing company), and item numbers are used as inputs. The size of feature vector consisting of continuous and one-hot encoded categorical variables is 70. The motivation behind this experiment is to identify stand-alone value of selected features in predicting stock prices. The models used in this experiment are logistic regression and multilayer perceptron. Since non-linguistic features are fed only to fully connected layers in LSTM and BERT models, multilayer perceptron model is chosen for this experiment.

Experiment 3: Prediction using 8-K filings and features as input

In this experiment, text features, continuous numerical features and categorical features are used in combination for the analysis. The motivation behind this experiment is to find out if combining the features improves the stand-alone predictive performances. Having the lowest accuracy among the classifiers, BERT model is excluded from the analysis in this experiment.

4.8 Training & Hyperparameter tuning

Logistic Regression

Data is split into test and validation sets with a ratio of ratio of 80%:20%. Logistic regression models with different combinations of the penalty parameters L1, L2 and elasticnet, and parameter C, which specifies the strength of regularization is fit in the training data and best parameters are chosen through k-fold cross validation (GridSearchCV) where k = 5. In the first experiment using only documents as features, the best performing parameters are C = 0.1 and L2 regularization for the model using bag-of-words features, and C = 0.001 and L2 regularization for the model using Tf-idf features. In the second experiment using the non-linguistic features, best performing parameters are C = 0.1 and L2 regularization as the regularization method. In the third experiment using the combination of texts and non-linguistic features, the best performing parameters are C

= 0.1 and L2 regularization for the model using bag-of-words features, and C = 1.0 and L2 regularization for the model using Tf-idf features.

LSTM

Training set of 17.361 data points are further divided into training and validation sets using a split ratio of 80%:20%, resulting a training set of size 13.888, and validation set of size 3.473. Combinations of different dropout rates, optimizer learning rates, beta1 parameters for adam optimizer (momentum parameter optimizing the convergence speed), and learning rate scheduler gammas (learning rate decaying factor), numbers of LSTM blocks, numbers of units (hidden state dimensions) in LSTM blocks and whether to include bias term in LSTM and fully connected layers are used as hyperparameters in a grid search. The parameters creating the model with highest validation accuracy is dropout rate = 0.25, bias = True, number of LSTM layers = 2, L2 regularization rate = 0.0, learning rate = 0.0001, hidden state dimension = 64 units, Adam's first moment beta = 0.9, Adam's second moment beta = 0.999, learning rate scheduler gamma = 0.5.

Table 2
LSTM hyperparameters

Hyperparameter	Grid	Selected
Dropout rate	[0.2, 0.25]	0,25
Use Bias	[True, False]	True
Number of BiLSTM blocks	[2, 4, 6]	2
L2 regularization lambda	[0.0, 0.01]	0.0
Learning rate	[0.001, 0.001, 0.0001]	0.0001
BiLSTM hidden state dimension h	[64, 128]	64
Learning rate scheduler gamma	[0, 0.2, 0.5]	0.5

MLP

Finally, hyperparameters of feedforward network using non-linguistic features are tuned by grid search through 3-fold cross-validation. Combinations of batch sizes = [16, 32, 64], number of epochs = [10,20,30], and dropout rates = [0.10, 0.20, 0.30] are tested, and best validation accuracy is achieved with batch size = 32, dropout rate = 0.2, number of epochs = 10.

4.9 Results

Results of Experiment 1 (8-K filings):

Table 3
Experiment 1.

Model	Accuracy
BoW Logistic Regression	0.73
Tf-idf Logistic Regression	0.73
GloVe LSTM	0.69
BERT	0.51

Results of 2-day price change labels are presented, as they are better predicted by the model compared to 1-day price change labels. Results indicate that simple models are better classifiers for this experiment. The majority baseline of 58% is surpassed by logistic regression (+26%) and LSTM (+19%) models. Training on long input sequences is problematic for BERT classifier. BERT is designed to create word and sentence-level representations, thus they fail to create document representations. When complete input sequences are used, BERT classifier yields an accuracy lower than random guess (~ 0.48). Best performance is achieved when long paragraphs are deleted from the texts, and max input length is set to 150. However, due to information loss caused by text deletion, BERT still underperforms other classifiers.

Results of Experiment 2 (non-linguistic features):

Table 4
Experiment 2.

Model	Accuracy
Logistic Regression	0.60
Feedforward Network	0.59

Results indicate that stand-alone value of numerical and categorical features are negligible in comparison to majority baseline.

Results of Experiment 3 (8-K filings & non-linguistic features):

Table 5
Experiment 3.

Model	Accuracy
BoW Logistic Regression	0.73
Tf-idf Logistic Regression	0.74
GloVe LSTM	0.70

Logistic regression model with Tf-idf features yields better results than LSTM model in both experiments 1 and 3. The majority baseline of 58% is surpassed by logistic regression (+28%) and LSTM (+21%) models. It can be concluded that vanilla LSTM model is incapable of keeping the context in memory, when used in long documents. Both LSTM and logistic regression accuracies increase slightly with the use of additional numerical features. In the next page, the classification report of best performing model (logistic regression using Tf-idf features and non-linguistic features) is discussed in detail.

Classification report of logistic regression with Tf-idf and non-linguistic features:

Table 6
Logistic regression classification report:

	precision	recall	f1-score	support
negative	0.72	0.58	0.64	1792
positive	0.74	0.85	0.79	2549
accuracy			0.74	4341
macro avg	0.73	0.71	0.72	4341
weighted avg	0.73	0.74	0.73	4341

The model has a tendency to predict majority class "positive" more often. The percentage of positive classes is 58%, while percentage of negative classes is 42%. Classification report reveals that model tends to predict negative instances as positive, thus recall of the negative class is lower. Positive classes on the other hand are correctly predicted as positive more often, thus recall of the positive class is high. F1-score, which combines the measures precision and recall is a more reliable classification metric than accuracy in imbalanced class distribution. The weighted average F1-score of the model is 0.74, which is an increase of 28% from majority baseline.

5. Discussion

The results show that although transformer and recurrent neural network models perform well on other classification tasks, they fail to achieve similar results in 8-K form analysis, length of the documents being the major obstacle preventing models from learning. The complex nature of the stock market mandates that forecasting models take multiple variables into account. When other variables are added in the model along with the information in documents, predictive performance has increased, suggesting that a more inclusive model can yield to further improvement. In literature, the problem of variables is dealt with by adding variables such as earnings surprise, or 1-week, 1-month, 1-quarter moving averages. Earnings surprise is considered to be a highly predictive feature, however its calculation is beyond the scope of this study given the time and resources allocated. Moving averages on the other hand allow integrating time series data in the models, which may improve the performance. Apart from other explanatory variables, increasing the size of training and validation data may also improve the performance. Due to resource constraints, 6-month data is scraped and used in this research. The models are tested on AG News dataset with a training set size of 60.000, which has yielded accuracies above 0.92. Most studies referenced in this study use over a hundred thousand documents spanning over the years. Lee and Surdenau provide a link for 10-year 8-K document, EPS, and moving averages data(Lee et al. 2014). The data however belongs to years 2002-2012 with entirely different market dynamics, and the results may not applicable to the current market. In addition, the EPS and moving averages in the dataset belong to S&P 500 stocks only. To build a more generalizable model, this is not preferred in this analysis.

One of the most valuable findings is that numerical and categorical features selected for the analysis are not predicting stock prices alone. The best models using only these

features have improved majority baseline by 2% , the best performing model using only texts have improved it by 26 %, and the best model using both texts and features have improved it by 28%. The intuition behind this is that the more context we have , the more powerful text features become. Same information can mean different things in different times and market conditions, which is why volatility index of the market is included. Similarly, an information may mean different things for different companies, which is why industries and market sizes are included.

The closest works to this study are Lee and Surdenau's "On the Importance of Text Analysis for Stock Price Prediction" and Masoud's "Attention-based Stock Price Movement Prediction Using 8-K Filings". Both of these studies aim to predict the direction of stock-price changes using 8-K filings, rather than predicting sentiment. Both use additional non-linguistic features earnings surprise, and 1-week, 1-month moving averages of the stock price to improve the performance of their models. They perform 3-class classification, and their best performing models achieve 54.4 and 57.02 test accuracies respectively. 3-class classification results of this study surpass these two, best performing model logistic regression yielding a 0.66 test accuracy. The accuracy may increase further if earnings surprise and moving averages are included, and the amount of data (time horizon) is increased. During labeling, both studies have normalized stock price changes by change in market price, which ignores the stock's sensitivity to changes in the market. Making use of financial theory, by introducing CAPM in the normalization process is a distinguishing characteristic of this study, and possible reason for better results.

The performance of attention-based networks can be improved by increasing their memory capabilities with *Hierarchical Attention Networks*, *Transformer XL* or *Hierarchical Transformer* architectures (Yang et al. 2016; Dai et al. 2019; Pappagari et al. 2019). Other possible solutions to memory problem is to get representations for each item content i.e. each event separately, rather than the entire document, and use attention mechanism to attend for relevant events, as suggested by Masoud (Masoud 2019). The performance of BERT model may be improved by using domain-specific BERT FinBERT by Prosus AI, which is trained on financial corpus (Prosus.Ai 2021). However the problem of document lengths will remain. It is worth to mention that there are more sophisticated asset pricing models than CAPM, which can be used to normalize the stock returns more accurately. Finally, building a multi-class model trained on correctly identified 3-class labels would be a more desirable solution from a practical point of view. In a binary model, even the slight increases are classified as positive, and vice versa, which is not profitable as an investment strategy considering the transaction costs. In summary, including more explanatory variables, increasing the amount of data, improving the neural network models, and applying a more sophisticated financial theory during labeling process are the possible actions to improve the results.

6. Conclusion

The main focus of this research is to test whether text analysis can be used in predicting the direction of the stock prices. As a secondary research question, the effect of including additional features in the language-based model is examined. To understand the stand-alone values of documents and non-linguistic features, three experiments are carried out. Logistic regression, LSTM and BERT models are trained on documents (Experiment 1), non-linguistic features (Experiment 2), and combination of documents and non-linguistic features (Experiment 3). Documents are represented with BoW, Tf-idf, GloVe,

and bidirectional transformer encoder. The findings suggest that there is a correlation between company disclosures and short-term stock price changes, which can be used to build a predictive model. The numerical and categorical features chosen improve the performance when added to this model, but do not predict the prices by themselves. Due to the lengths exceeding the memory capacities of traditional LSTM and BERT models, the price changes are best predicted with Tf-idf features and logistic regression. Trained to create representations for words and sentences, BERT has failed to create good representations of the documents. Nevertheless, accuracies of LSTM and logistic regression models trained only on documents have surpassed the majority baseline by a large margin. Additional features have improved the results of all models tested in third experiment, bringing the weighted F1-scores of logistic regression classifier to above majority baseline by 28%.

The study is a step towards understanding how new information is reflected on prices. It aims to understand investor perception and reaction to new information, contributing in the field of behavioral economics. By building a predictive model, it provides a tool for making informed investment decisions. Such trading corrects the prices by moving them towards the accurate predictions (Haeberle 2015). Correct pricing is necessary for healthy functioning of financial markets, which contributes in the economy by providing liquidity, funding and incentives for the companies.

The study provides a basis for the experiments combining textual information and numerical features in predicting stock price movements. It is distinguishing for combining financial theory with machine learning by including capital asset pricing model in the calculations. It includes all U.S. stocks instead of S&P 500 stocks, which expands the scope and improves generalizability. Further research addressing the factors discussed in discussion section will help developing a better understanding of the relationship between the events, company characteristics, market conjuncture and stock prices.

References

Baker, H Kent and John R Nofsinger. 2002. Psychological biases of investors. *Financial services review*, 11(2):97.

Barber, Brad M and Terrance Odean. 2002. Online investors: do the slow die first? *The Review of Financial Studies*, 15(2):455–488.

Cboe. 2021. Cboe global markets. https://www.cboe.com/.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Facebook. 2021. Facebook ai. https://ai.facebook.com/.

Fama, Eugene F and Kenneth R French. 1998. *Dividend yields and expected stock returns*. University of Chicago Press.

Francis, Jennifer and Katherine Schipper. 1999. Have financial statements lost their relevance? *Journal of accounting Research*, 37(2):319–352.

FRED. 2021. Federal reserve economic data. https://fred.stlouisfed.org/.

GitHub. 2021. Google-research/bert. 2018. google research. https://github.com/google-research/bert.

Goyal, Amit and Ivo Welch. 2004. A comprehensive look at the empirical performance of equity premium prediction. *NBER Working Paper No. 10483. JEL No. G12, G14*.

groups.di. 2021. Ag's corpus of news articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Haeberle, Kevin. 2015. Stock-market law and the accuracy of public companies' stock prices. *Colum. Bus. L. Rev.*, page 121.

Hiew, Joshua Zoen Git, Xin Huang, Hao Mou, Duan Li, Qi Wu, and Yabo Xu. 2019. Bert-based financial sentiment index and lstm-based stock return predictability. *arXiv preprint arXiv:1906.09024*.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. Https://doi.org/10.1162/neco.1997.9.8.1735.

Honnibal, M. and I. Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.

kaggle.com. Sec filings 1994-2020. https://kaggle.com/finnhub/sec-filings.

Kowsari, Jafari Meimandi and Mendu Heidarysafa. 2019. Barnes, and brown,". *Text Classification Algorithms: A Survey," Information*, 10(4):150.

Lee, Chien-Cheng, Zhongjian Gao, and Chun-Li Tsai. 2020. Bert-based stock market sentiment analysis. In *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, pages 1–2, IEEE.

Lee, Heeyoung, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. 2014. On the importance of text analysis for stock price prediction. In *LREC*, volume 2014, pages 1170–1175.

Loughran, Tim and Bill McDonald. 2010. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of finance*, 66(1):35–65.

Malkiel, Burton G and Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417.

Masoud, Mohamed. 2019. Attention-based stock price movement prediction using 8-k filings. *Stanford*.

NAICS. 2021. Search naics codes by industry. https://www.naics.com/search-naics-codes-by-industry/.

NLTK3.6.2. 2021. Natural language toolkit. https://www.nltk.org/.

PaperswithCode. 2021. Google's neural machine translation system: Bridging the gap between human and machine translation. https://paperswithcode.com/paper/googles-neural-machine-translation-system.

Pappagari, Raghavendra, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844, IEEE.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher,

M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
Prosus.Ai. 2021. Home. https://prosus.ai/.
Python.Org. 2021. Python language reference, version 3.9.5. https://www.python.org/.
Qian, B. 2005. Rasheed k. hurst exponent and financial market predictability. In *Proceedings of the Second IASTED International Conference on Financial Engineering and Applications*, volume 209.
Richardson, Leonard. 2007. Beautiful soup documentation. *April*.
Schumaker, Robert and Hsinchun Chen. 2006. Textual analysis of stock market prediction using financial news articles. *AMCIS 2006 Proceedings*, page 185.
Scikit-Learn. 2021. Scikit-learn 0.24.2 documentation. https://scikit-learn.org/stable/about.html#citing-scikit-learn.
Setiono Strong, Norman, Bambang. 1998. Predicting stock returns using financial statement information. *Journal of Business Finance & Accounting*, 25(5-6):631–657.
Sharpe, William F. 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19(3):425–442.
Sohangir, Sahar, Dingding Wang, Anna Pomeranets, and Taghi M Khoshgoftaar. 2018. Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, 5(1):1–25.
Tetlock, Paul C, Maytal Saar-Tsechansky, and Sofus Macskassy. 2008. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467.
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.
WRDS. 2021. Wharton research data services. https://wrds-www.wharton.upenn.edu/.
Yang, Yi, Mark Christopher Siy UY, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.
Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.
Zoen Git Hiew, Joshua, Xin Huang, Hao Mou, Duan Li, Qi Wu, and Yabo Xu. 2019. Bert-based financial sentiment index and lstm-based stock return predictability. *arXiv e-prints*, pages arXiv–1906.