PREDICTING STOCK PRICE MOVEMENTS USING SENTIMENT AND SUBJECTIVITY ANALYSES

by

Andrew Kirby

A master's thesis submitted to the Graduate Faculty in Liberal Studies in partial fulfillment of the requirements for the degree of Master of Arts, The City University of New York

2021

PREDICTING STOCK PRICE MOVEMENTS USING SENTIMENT AND SUBJECTIVITY ANALYSES

by

Andrew Kirby

This manuscript has been read and accepted for the Graduate Faculty in Liberal Studies in satisfaction of the thesis requirement for the degree of Master of Arts.

Date	Kyle Gorman
Date	Gita Martohardjono

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

PREDICTING STOCK PRICE MOVEMENTS USING SENTIMENT AND SUBJECTIVITY ANALYSES

by

Andrew Kirby

Advisor: Kyle Gorman

In a quick search online, one can find many tools which use information from news headlines to make predictions concerning the trajectory of a given stock. But what if we went further, looking instead into the text of the article, to extract this and other information? Here, the goal is to extract the sentence in which a stock ticker symbol is mentioned from a news article, then determine sentiment and subjectivity values from that sentence, and finally make a prediction on whether or not the value of that stock will go up or not in a 24-hour timespan. Bloomberg News articles published between 2008 and 2013 were used as a data source, and prices of stocks were acquired using Yahoo Finance. News and information influence human behavior; constantly changing, the effects of this information on the market can be observed daily. This technology could assist people in making better decisions on how to invest their money.

INTRODUCTION

Where does value come from? How do people determine how much a thing is worth? How do people determine if something is worth less, or more, than it was worth in the past? It has been said that, in an ideal system of exchange, the answer to all these questions is relevant, truthful, and factual information. Notions of how the stock market reflects information come from the Efficient Markets Hypothesis, which states that the price of a stock comes from the aggregate of information available concerning it (Fama 1965). This hypothesis is the basis for this project, which assumes that financial news is relevant and connected to the price of a stock. While information generally encapsulates more than just news, ranging from forum posts, blogs, television, user-created videos, or memes, financial news remains meaningful since it is generally expected to be reliably sourced, with a motivated publisher taking responsibility for its content.

Financial news is intended to help investors learn about what is happening in the business world. This information can help to assist a person with making a short-term decision on whether or not to invest in a stock. While it is possible for a person to read an article and discern its sentiment and meaning, any one person is unable to do this for, say, a thousand articles every day. Information and markets are typically highly dynamic in nature, meriting near-constant attention and focus. As news propagates, it is possible to observe how it affects stock prices over time. A business creates something new, people like it, the news talks about it, and the price of the stock moves upward. Conversely, something could go wrong, such as a sudden product recall, which hits the news, and the stock price drops because attitudes toward the company have turned negative. For example, in an article posted on May 5, 2021, it was revealed that the Biden

administration was going to waive intellectual property rights for COVID-19 vaccines manufactured by certain pharmaceutical companies. The prompt result was the following:

> Stocks of major pharmaceutical companies that have produced vaccines, including Moderna, BioNTech and Pfizer, dropped sharply after news of the potential waivers first broke. Pfizer ended its trading day flat, while Moderna lost 6.1%; Johnson & Johnson shed a modest 0.4% (Macias 2021).

Note the meaningful short-term drop in Moderna's stock. With this in mind, the general problem here is that there are so many companies, and so much information, that it is nearly impossible for an individual working alone to keep track of it all. It is easy to catch major headlines such as this one; an average person with a non-finance day job likely does not have time to delve further, looking at what's happening with every company in the S&P 500 or beyond.

So, how can we make a program to help people understand stocks and decide what to do with them? How can we design something that can read many, many articles quickly, extract information relevant to a given stock, and make a prediction on what will happen to that stock's price? First, the program needs to be able to identify the name of the securities mentioned in the text. Then, the program will need to find the words around the stock, perhaps in a sentence or a paragraph. A way must be found for the program to interpret the text numerically. Next, the program needs to gather context: what's being said about the stock? How will the program be able to change this into information that it can understand? Finally, the program needs to know something about the price before and after the news takes place, it can make generalizations regarding how news has effects on the prices of securities.

To that end, below are some general techniques used in processing text and making predictions, and finally an overview of systems that have attempted to solve these problems already.

Common approaches of processing text, changing it into numbers that a program can understand, consist of using Bag of Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) (Henrique et al., 2019). The Bag of Words approach consists of counting up all the words and putting them into vectors so that they can be computed into other forms of information. TF-IDF takes this a step further, by giving less importance to words that occur very frequently while not having a lot of useful content, like ‘the’ and ‘a’ or ‘it’. It also takes into account the quantity of documents in a document set in order to determine which words get more value (Eisenstein 2019, Ch. 2). The TF-IDF approach was introduced by Spärck Jones (1972).

In order to find the stocks and the sentences that talk about stocks, it is necessary to perform entity linking. Entity linking involves looking through text data and linking pieces of text that are present in a knowledge base, which is a collection of entities or things that are known and relevant to the search (Broscheit 2020). One form of this knowledge is a gazetteer, listing stocks and the associated companies that need to be found in pieces of text such as news articles. Gazetteers must be created, rather than learned, as in some cases of Named Entity Recognition. In a stock market, companies are represented with something called a *ticker symbol,* of a length no longer than four letters, that typically correspond to the company’s name in an identifiable, intuitive, or occasionally whimsical way. Some examples include: Microsoft, known as MSFT; Exxon Mobil, known as XOM; or Petco, known as WOOF. Sometimes a stock ticker can correspond to many forms of a name, such as AAPL. AAPL can refer to Apple, Apple, Inc.,

Apple Computer(s), or more colloquially, to its general location Cupertino, in the same way that the White House can be a common moniker for the US executive branch. Company names can also change over time, such as when Google (or Google, Inc., or even 'The Goog') changed its name to Alphabet in 2015 as it expanded into other domains of business. Names abstract in different ways; challenges can arise as the limits of the power of a knowledge base or gazetteer are reached.

Subjectivity measurements involve determining the degree to which a parcel of text is subjective or objective. This area of study is also commonly referred to as opinion mining, and tools available often integrate this feature with sentiment analysis. This is accomplished by using a model to make a prediction on another piece of text (Murray & Carenini 2009).

Sentiment Analysis is a measure of how positive or negative a parcel of text is. This can be accomplished in a variety of ways. Some examples include using data like tweets with happy or sad emoticons, product reviews with ratings, or a lexicon with positive and negative words to generate a quantity that corresponds with how positive or negative the item of text is (Eisenstein 2019, Ch. 4). Sentiment analysis can also be approached as a text classification problem, where machine algorithms are put to work to detect positive or negative sentiment. For example, film review data can be used to train a model to predict sentiment of other pieces of text (Pang & Lee 2004).

Logistic regression is a common tool used for classification. It is used with binary dependent variables, which have two values as in 0/1 or True/False. As a side effect of classification, it

estimates the posterior probability of true and false labels. using a scoring function for base features (like word vectors) and a binary label (Eisenstein 2019, Ch. 2).

Other machine learning techniques for classification commonly used in this area of study are neural networks, support vector machines (SVM) and random forests. Neural networks attempt to mimic biological forms of information processing. SVMs work by increasing the dimensions of a training vector and then classifying using a line. Random forests are a type of decision tree that helps to optimize classifications (Henrique et al., 2019).

One system for predicting stock outcomes using finance news, AZFinText, has been able to not only predict the trajectory of security prices, but do so with meaningful accuracy and robust returns in an investing simulation. This system used data from news, along with stock quotes and information from analysts, a noun phrase extraction technique, and a variation on SVM called *support vector regression for classification* (Schumaker & Chen, 2009).

Another system of the above kind used news to predict price movements in the Korean Exchange. In this system, the authors converted the text of the entire article into TF-IDF vectors, assessed these vectors for statistical relevance, and used an SVM tuned with grid search for classification. The goal was to evaluate the degree to which a news article caused a stock to change (Nam & Seong, 2019).

Another group of authors created a system which performs an entity-linking process, searching for mentions of the Yahoo, Microsoft, and Facebook companies in a diverse set of news sources

including but not limited to Reuters, the Wall Street Journal, and Google Finance. This study also uses TF-IDF vectors (including what appears to be the entire article), Naive Bayes with an 'opinion mining based sentimental dictionary' for classifying sentiment, and K-Nearest Neighbors for classifying whether stocks will rise or fall. The time interval for the price change is the closing price of the date in which the article was written and the closing price of the previous date (Khedr & Yaseen 2017).

Lastly, a system by Ding et al. (2014) features a dependency parser that picks up relevant noun phrases in order to find relevant companies mentioned in sentences. It then uses WordNet to extract lemmatized forms of words, converts them into vectors with TF-IDF, and then performs classification using SVM. It detected about 1,800 'instances' with article titles and sentences from the articles. This set of instances were then split into training and testing. For prices, the authors used a day, a week, and a month after the publish date of the article (Ding et al., 2014).

METHODOLOGY

This model will be a supervised linear binary classifier. Given that a security was mentioned in a Bloomberg article on day n, this model will attempt to predict whether that security's price will rise or fall on the next trading day, $n + 1$.

As a baseline, I used a 'beta nonsense model' with 5000 sets of 5 random integers numbering 1-5 for features. This predicts that security prices will increase roughly 65% of the time. McNemar's Test is used to compare the proposed model to the baseline.

The corpus used in this project consists of financial news articles from Bloomberg,[1] with articles from between 2008 and 2013. These articles contain finance news about companies from around the world. They also contain information about when the article was published and the author. This dataset was first compiled and used in Ding et al. 2014.

For entity linking, I used SpaCy's `PhraseMatcher`[2] to locate stock ticker entities, along with the sentence which contained them. The `PhraseMatcher` works by using 'rule-based matching', meaning that rules must be made for the matcher to work around. In simple terms, the rule stipulates that the program should match entities listed in the gazetteer to identical entities found in each article, and extract the sentence in which the entity was mentioned. It will then go through each document and produce matches and their sentences, which I then passed into a data frame. The intention is to go directly into the part of the text that mentions the stock, and extract that information for classification. I used a gazetteer to populate the matcher with a list of companies from the S&P 500, which was gathered from Wikipedia[3] in March 2021.

	A	B	C	D
1	ticker	names_1	names_2	names_3
2	MMM	3M Company	The 3M Company	3M Co.
3	ABT	Abbott Laboratories	NaN	NaN
4	ABBV	AbbVie Inc.	AbbVie, Inc.	AbbVie, Incorporated
5	ABMD	Abiomed	Abiomed Inc.	Abiomed, Inc.
6	ACN	Accenture	Accenture plc	NaN
7	ATVI	Activision Blizzard	Activision Blizzard, Inc.	Activision Blizzard Inc.
8	ADBE	Adobe Inc.	Adobe Inc	NaN
9	AMD	Advanced Micro Devices	NaN	NaN
10	AAP	Advance Auto Parts	NaN	NaN
11	AES	AES Corp	Applied Energy Services	AES Corporation

Figure 1: An example of the information contained in the gazetteer. Column A contains the stock tickers, and the others contain the variations on the names of each company. NaNs are to fill in empty spaces.

[1] Bloomberg News corpus obtained from: https://github.com/philipperemy/financial-news-dataset
[2] SpaCy `PhraseMatcher`: https://spacy.io/api/phrasematcher
[3] S&P 500 at Wikipedia: https://en.wikipedia.org/wiki/S%26P_500

To obtain prices, I used the `yahoofinancials`[4] Python module to extract stock price information from a given date. I gathered the opening price of the stock on the date in which the article was written, and then the closing price of the stock on the next trading day. I then took the difference of these two values and made a classification based on this value.

For sentiment analysis, I used NLTK's VADER[5] (Valence Aware Dictionary for Sentiment Reasoning) as a model for evaluating sentiment polarity. VADER uses a weighted sentiment lexicon and is tuned for 'microblog-like contexts'. This model is then tested on human-annotated gold standard data to measure effectiveness (Hutto & Gilbert, 2014). For evaluating subjectivity, I used TextBlob.[6] TextBlob evaluates subjectivity using a model trained on annotated movie review data from IMDB.

I created text vectors from the sentences surrounding the named entities using the `TfidfTransformer` from scikit-learn.[7] I then concatenated these vectors with the sentiment and subjectivity values acquired above. I used unigram features here, as a default.

Each article in the corpus was stored in a directory named for the date in which the article was written. This date information is then supplied to the `yahoofinancials` module. From there, I obtained the opening price for the stock on the date in which the article was written and the closing date on the following date for the price of the same stock. I then obtained the difference between these two prices. If the price rose, the features were classified as a 'buy!'. If the price

[4] `yahoofinancials`: https://pypi.org/project/yahoofinancials/
[5] NLTK VADER: https://www.nltk.org/howto/sentiment.html
[6] TextBlob: https://textblob.readthedocs.io/en/dev/
[7] TF-IDF: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

remained the same or fell, the features were classified as a ‘sell!’. If a price was not able to be obtained for the following 24 hours, the data item was dropped. As a result, weekends and holidays on which no trading occurs were not included in the data. If no articles were found containing a given stock ticker on a given day, then no sentence was grabbed, and no features or classification assigned. For an example of the data generated, see Figure A in the Appendix.

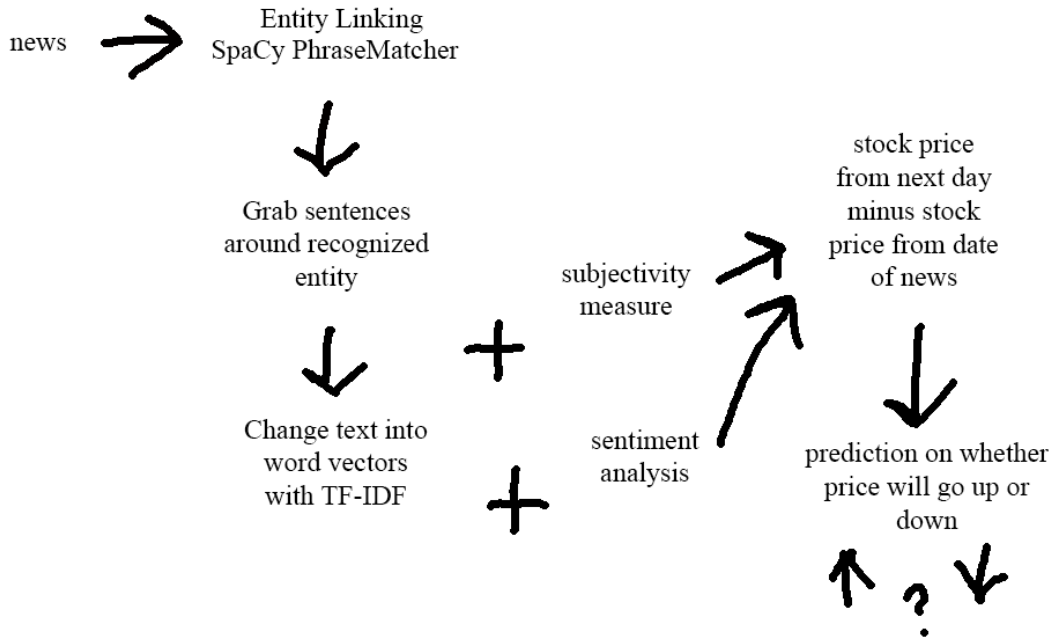

Figure 2: Diagram of this system.

For training, development, and testing, I used a 70-10-10-10 split; 70% for training, 10% for development, 10% for testing, and 10% for secret testing. The dataset contained 5,649 total data points. After partitioning of the data, each data slice was shuffled. The classifier was run five times, with five different seeds (333, 4, 15, 21, 81), and the median was taken for the final measure. The development, test, and secret test sets all include data from news that occurred after the training set. The secret test set was opened and processed in isolation from the training, development, and testing sets. The non-secret test sets were opened and processed together,

using the process described above. Figure 3 below illustrates what is described here, along with time spans of each set.

	TRAIN	DEV	TEST	SECRET
LENGTH	3954	564	564	564
TIME SPAN	2008-01-02 to 2010-10-07	2010-10-07 to 2010-10-13	2010-10-13 to 2011-09-07	2013-10-22 to 2013-11-26
PARTITION	0.7	0.1	0.1	0.1

Figure 3: Information on the size, date intervals of the articles included in the partition, and partition ratios.

For classification, I used a logistic regression with the following parameters: L1 penalty, a C = 0.1, and the `liblinear` solver, which gave the best results on held-out data. If mentions of a given stock occurred in multiple articles on the same day, those data points were included, along with identical 'buy!' or 'sell!' tags. On some occasions, a sentence would contain mentions of two securities. In this case, the same sentence was included twice, one for each individual security, along with the corresponding tags.

Some stock ticker symbols that interfered too much with entity linking were eliminated from the list of ticker symbols that the program searched for, such as A (Agilent Technologies) or IT (Gartner, Inc.). These were dropped because 'A' could be the article 'a' or the stock ticker, and the matcher would erroneously acquire the sentence even though that sentence wasn't talking about that stock. The same goes for IT: the acronym commonly refers to information technology, and the matcher would pick up any sentence mentioning 'IT', thinking it was referring to Gartner, Inc. when, in reality, it is not the case. Additionally, some other tickers were resulting in errors because the Yahoo Financials library was unable to acquire prices for them: GM, NWS,

NWSA, and UA. Dropping these did not meaningfully affect the accuracy or results. An example of this has been included in Appendix Figure B.

The code for this project, along with this article, are in the link at the footnote below.[8]

RESULTS

TF-IDF DEV:	**TF-IDF + SENTIMENT DEV:**	**TF-IDF + SUBJECTIVITY DEV:**	**ALL DEV:**
0.298	0.298	0.298	0.298
TF-IDF TEST:	**TF-IDF + SENTIMENT TEST:**	**TF-IDF + SUBJECTIVITY TEST:**	**ALL TEST:**
0.298	0.298	0.298	0.298
TF-IDF secret:	**TF-IDF + SENTIMENT secret:**	**TF-IDF + SUBJECTIVITY secret:**	**ALL secret:**
0.447	0.447	0.447	0.447

BETA NONSENSE MODEL	**MCNEMAR P-VALUE**
0.65	< .001

Figure 4: Accuracy scores with logistic regression, taken from a median of five runs. 5649 total data points; 70:10:10:10 training/development/testing/secret testing split.

The median accuracy remained constant for the development and test sets at 0.298, despite the addition of additional features. Including bigrams, rather than unigrams, produced no change. Doing the same with BoW (as opposed to TF-IDF) produced no change. Performing an identical logistic regression on the secret test set returned an accuracy of 0.447. The beta nonsense model returned an accuracy of .605, a score higher than the other sets indicated above. According to this test, the model was significantly worse than baseline ($p < .001$).

[8] GitHub link: https://github.com/andrewlkirby/CL_MA_Thesis

DISCUSSION

When comparing my model to the beta nonsense model, it is apparent that my system is ineffective, producing accuracies that were worse than flipping a coin. Adding additional features did not modify results. Changing the TF-IDF matrices from sparse to dense also had no effect when combining with my other features. Earlier versions of the model were able to make better predictions with around 20% of the current quantity of data, with an 80:20 partition. However, the inclusion of additional data and correct data shuffling on the partitions shows that the earlier promising results were not what they appeared to be. Because of the uneven distribution of data over time, and the uneven distribution of data points picked up, the development set included a very short time interval, yet returned identical prediction results. The secret test set returned different accuracy scores. The only difference between it and the development and test sets was that it was opened and processed separately from the other sets, in its own separate CSV file. This is a likely indicator of an issue with the data handling; future experiments should include data that should be separated into their own respective CSV files, and then loaded into dataframes separately, as was done with the secret test set. These scores are worse than that of 'All News' indicated in the table below in Ding et al. (2014). The authors of that study used the same dataset as this one. 'All News' refers to any mention in any article of the company in question.

Google Inc.		
Company News	**Sector News**	**All News**
Acc	Acc	Acc
67.86	61.17	55.7
Boeing Company		
Company News	**Sector News**	**All News**
Acc	Acc	Acc
68.75	57.14	56.04
Wal-Mart Stores		
Company News	**Sector News**	**All News**
Acc	Acc	Acc
70.45%	62.03%	56.04%

Figure 4: Results from Ding et al. (2014, p. 1420). The accuracy scores, indicated in column 'Acc' in All News are roughly similar to the ones produced here.

CONCLUSION

The intention of this project was to go further than predicting changes in the Dow Jones industrial average by using specific stocks. I had also hoped to use text from within an article, rather than its headline and subheader, to make a vague prediction about whether a stock's price would go up or down in a day. However, one obvious but important conclusion is to be made here: having a sufficient quantity of data is necessary to come to the right conclusions. While there was some initial excitement when working with sets of 100 and 1,400 data points, it became clear as I added additional data that what I had seen initially was not quite accurate. Adding additional features also had initial effects that ultimately diminished to nil as the quantity of data increased, possibly because the TF-IDF (or bag of words) matrices contained too many values for the other features to have a meaningful effect. A small cause for optimism is that I was able to achieve comparable (if not great) results to another study using significantly simpler techniques.

For future work, it would be helpful to attempt to predict the magnitude or degree to which a stock price increased or decreased. More data should be added from diverse sources. More features could be added. One set of features could include past trends of the stock and the degree to which news has been affecting it. Some stocks may only move around a little with ordinary news, while a significant event occurring could have a huge effect on the news. An explicit sentiment or subjectivity analysis using labeled domain-specific financial news data could also prove to be a richer, more impactful feature to integrate into the system. Another set of features could be added that would include a set of tags indicating which parts of the article are relevant to the stock mentioned.

While this system predicts based on a 24-hour period, in which a person would rapidly buy and sell stocks within that period, a more sophisticated system could incorporate information over a longer time interval, and assume that the person does not have infinite funds. In essence, the assumption here is that someone is buying a fixed unit of a purchasable stock if the system predicts ‘buy!’, and then sells that unit on the following day. The same would apply to a ‘sell’ prediction. Consequently, this assumption ignores trading fees and other related costs. In addition, this assumption relies on a hypothetical individual with an endless quantity of money. To that end, an improved system could account for such fees, and interpret stock units of many values, and tag them accordingly. This improved system could also simulate trading gains and losses over time, and simulate a real person’s finite supply of money. In addition, weekends and holidays could be included in the price data, increasing fidelity to a more realistic situation for the data to reflect. A simulation could also be run to verify the efficacy of the system monetarily.

Trading stocks based on algorithms has become significantly more commonplace in recent years. These algorithms, produced at great effort and expense by trading firms, are unavailable to the general public. By making this system and others like it available to the general public, it is my hope that people will have access to decision-making technology that could give them the same advantages that these firms have.

APPENDIX

ids	sents	string_ids	dates	buy?
GOOG	Google Inc. (GOOG US) agreed to purchase Zagat Survey LLC, a rival restaurant review and ratings service.	TICKER	2011-09-07	1
MSCI	The MSCI All-Country World Index of shares rose as much as 2.8 percent.	TICKER	2011-09-07	1
IBM	Full-Year Forecast IBM signed 13 contracts greater than $100 million last quarter, down from 16 a year ago.	TICKER	2010-04-20	0

Figure A: Example of data:
Stock tickers are found, along with the sentence in which they occurred. This is connected to the date of occurrence. A 1 on 'buy?' indicates the stock price was up the next day; a 0 means the stock price did nothing or went down. The column `string_ids` indicates the type of entity matched. In this case, looking at the column `ids`, we see that the entity GOOG is a ticker symbol.

ids	sents	string_ids	dates	buy?
MS	Lockyer developed the proposal after reviewing solicitations from Morgan Stanley (MS) , Citigroup Inc. (C) , Wells Fargo & Co. (WFC) and RBC Capital Markets in late 2009 and early 2010, according to documents from the treasurer	TICKER	2011-09-07	1
C	Lockyer developed the proposal after reviewing solicitations from Morgan Stanley (MS) , Citigroup Inc. (C) , Wells Fargo & Co. (WFC) and RBC Capital Markets in late 2009 and early 2010, according to documents from the treasurer	TICKER	2011-09-07	0
WFC	Lockyer developed the proposal after reviewing solicitations from Morgan Stanley (MS) , Citigroup Inc. (C) , Wells Fargo & Co. (WFC) and RBC Capital Markets in late 2009 and early 2010, according to documents from the treasurer	TICKER	2011-09-07	0

Figure B: An example of multiple stocks being mentioned in one sentence.

REFERENCES

Broscheit, S. (2020). Investigating entity knowledge in BERT with simple neural end-to-end entity linking. *arXiv preprint arXiv:2003.05473*.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2014, October). Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1415-1425).

Eisenstein, J. (2019). *Introduction to Natural Language Processing*. MIT Press.

Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, *38*(1), 34-105.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226-251.

Hutto, C., & Gilbert, E. (2014, May). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 8, No. 1).

Khedr, A. E., & Yaseen, N. (2017). Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications*, *9*(7), 22.

Macias, A. (2021, May 6). *U.S. backs waiving patent protections for Covid vaccines, citing global health crisis*. CNBC. https://www.cnbc.com/2021/05/05/us-backs-covid-vaccine-intellectual-property-waivers-to-expand-access-to-shots-worldwide.html

Murray, G., & Carenini, G. (2009). Predicting subjectivity in multimodal conversations. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 1348-1357).

Nam, K., & Seong, N. (2019). Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market. *Decision Support Systems*, *117*, 100-112.

Pang, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)* (pp. 271–278).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825-2830.

Schumaker, R. P., & Chen, H. (2009). A quantitative stock prediction system based on financial news. *Information Processing & Management*, *45*(5), 571-583.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28,* 11–21.