LEBANESE AMERICAN UNIVERSITY

Stock Price Prediction via Sentiment Analysis on News Corpora

By

Maher Al Watchi Al Hayek

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

School of Arts and Sciences
August 2021

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS APPROVAL FORM

Student Name: Maher Al Watchi Al Hayek I.D. #: 201402278

Thesis Title: Stock Price Prediction via Sentiment Analysis on News Corpora

Program: Master of Science in Computer Science

Department: Computer Science and Mathematics

School: Arts and Sciences

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Master of Science in the major of Computer Science

Thesis Advisor's Name: Dr. Faisal Abu-Khzam

Signature: [redacted] Date: 14 / 08 / 2021
Day Month Year

Committee Member's Name: Dr. Ramzi Haraty

Signature: [redacted] Date: 14 / 08 / 2021
Day Month Year

Committee Member's Name: Dr. Victor Khachan

Signature: [redacted] Date: 14 / 08 / 2021
Day Month Year

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Maher Al Watchi Al Hayek

Signature: [redacted]

Date: 19 / 07 / 2021
Day Month Year

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Maher Al Watchi Al Hayek

Signature: [redacted]

Date: 19 / 07 / 2021
Day Month Year

Acknowledgments

This work could not have been possible without the generous assistance and support that I have been fortunate to receive.

I am more than grateful to my advisor, Dr. Faisal Abu Khzam, for his unrelenting academic and moral support that he has furnished me with throughout this challenging journey. It is chiefly due to his inestimable feedback and patient guidance that this endeavor has been brought to fruition. I would also like to thank the committee members, Dr. Ramzi Haraty, for his valuable comments and advice, and Dr. Victor Khachan, for granting us the much-needed linguistics expertise that this work relied on.

My friends and family were always there for me when I needed them most, and I am truly thankful that they have accompanied me during this lengthy project. Lastly, my path to the finish line would have been infinitely more tortuous had it not been for the incalculable support of my lifelong friend, Marc Nassar.

Stock Price Prediction via Sentiment Analysis on News Corpora

Maher Al Watchi Al Hayek

Abstract

The notion of stock market gains is an enticing one. For researchers, however, succeeding in developing a system that can predict market movements can be in and of itself an even more rewarding feat. With the rise of artificial intelligence in general, and machine learning-based sentiment analysis in particular, the dream of stock market prediction has never been closer to our grasp. By leveraging the massive amounts of news data being cranked out daily, we can gauge the market mood via sentiment analysis techniques. We develop a novel version of the random forest classifier infused with the powers of collocation and concordance, both of which borrowed from the field of linguistics. Our experimental analysis yields insightful and impressive results compared to other works in the literature. Our novel model achieves a whopping 85% accuracy in predicting stock movements.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning, Financial Forecasting, Natural Language Processing, Financial Corpora.

Table of Contents

List of Figures

List of Tables

Chapter One

Introduction

Ever since the economist Eugene Fama popularized the Efficient Market Hypothesis (EMH) in his PhD dissertation [11] and his seminal review of efficient capital markets [12], it has been widely accepted that all the information available for investors is already incorporated in, and reflected by, market prices. It therefore follows that any new set of information that makes its way toward investors would result in an update in prices, after said investors readjust their stance with regards to their assets portfolio. This stream of information, however, has been relentlessly expanding in bandwidth with the advent of the internet. A myriad of sources overwhelm investors with a deluge of non-stop new information including digital newspapers, news websites, specialized financial newswires (Dow Jones Newswires, Refinitiv, Bloomberg Terminal), forums, corporate financial disclosure platforms such as the U.S. Securities and Exchange Commission (SEC), stock message boards, and social media platforms such as Twitter, Reddit, Stocktwits, and more. For one, this plethora of data sources presents itself as an invaluable resource with enormous untapped potentials of information extraction and extrapolation opportunities. On the other hand, the task of information-gathering that the investors must face is not merely overwhelming, but outright impossible at a comprehensive scale. The process of automating the ingestion of information is therefore an enticing opportunity to jump on, to say the least. With the recent advances in the overarching field of artificial intelligence, it is not a question of if, but when, will machines be able to undertake the daunting task of understanding those streams of information and determining whether stock prices will continue riding their trend curve or hit an inflection point on their rollercoaster journey. The real value, therefore, lies not in our possession of mountains of data, but in our ability to mine those mountains for their precious gems. Simply put, making sense of our data is a modern-day superpower. In this work, we will endeavor to develop a system that consumes financial news corpora on one end, and provides a market prediction on the other. In particular, the system will try to mimick the behavioral aspect of market participants, in that it will strive to determine the sentiment that a certain news article would elicit from its reader — the investor — and consequently predict the market effect of said investor's resulting change of stance, or lack thereof, with regards to their assets portfolio.

The rest of this thesis is structured as follows. Chapter II touches on the preliminaries of stock market predictions, with and without the context sentiment analysis. Chapter III provides a comprehensive literature review of previous related work, tracing back the origins of the field. In chapter IV, we take a deep dive into several techniques used to perform sentiment analysis with the aim of predicting stock prices, and explain our novel approach. Chapter V presents our experimental analysis of these techniques and a comparative study of their performance. We conclude the thesis and discuss future work in chapter VI.

Chapter Two

Preliminaries

2.1 Stock Market Predictability

When it comes to financial forecasting, the stakes are quite high as companies that are better at financial forecasting have the potential to reap more rewards in today's volatile and ever-changing business environment. To that end, financial forecasting can be divided into two broad approaches: primarily, those that deal with predicting prices, market volatility, trading volume, etc., and, secondarily, those that deal with cyber security (detecting fraud) and managing supply chains. Going back to the increased rewards seen with companies that are better at financial forecasting, a central question that needs to be answered is where said increased rewards, or excess return, come from. One hypothesis that aims to explain this excess return seen with financial forecasting is the efficient market hypothesis [11]. This hypothesis posits that no type of fundamental analysis can generate excess returns – new information comes into the market and is immediately reflected in stock prices, thereby creating an information-efficient system. In contrast, behavioral economics, the field of economics concerned with the effects of psychological, cognitive, and social factors on decision-making, offers the hypothesis of adaptive markets: the asymmetrical profits can be explained by the asymmetry of available information between the participants of any given market. As for the type of information that should be used to create better financial forecasting systems, [19] note two possible mining techniques which are built on two different philosophical approaches: firstly, one could "see the future from the past" by applying data mining techniques to historical data, thus finding trends in said data which are likely to propagate further down the line. Secondly, the other mining technique is more focused on zoning in on the right type of information rather than predicting the future based on a variety of information. This technique employs text mining and other NLP techniques to accomplish its objective.

2.2 ML for Stock Market Prediction

In this section, we will survey the literature for stock market prediction based on machine learning techniques, aiming to predict future stock prices from past stock prices as opposed to predictions from textual news. The ML techniques chosen can be broadly classified into five groups based on the type of data they deal with: (1) time-series combined with non-time series and financial time series, (2) time-series alongside non-time series, (3) time-series alongside financial time series, (4) non-time series combined with financial time series, and (5) financial time series.

ML techniques that deal with all three types of data, namely, time-series, non-time series, and financial time series data (group 1) comprise the broadest group. This group includes 16 different ML techniques: artificial neural networks (ANNs), back propagation neural networks (BPNNs), fuzzy c-means (FCM), genetic algorithms (GAs), generalized regression neural network (GRNN), hidden Markov model (HMM), k-means, k-nearest neighbors (KNNs), long short-term memory (LSTM), multi-layer perceptron (MLP), recurrent neural networks (RNN), particle swarm optimization (PSO), self-organizing maps (SOM), radial basis function neural networks (RBF), random forest (RF), and support vector machines (SVM).

The second group, which comprises ML techniques that handle time-series + non-time series data, includes only two such techniques: hierarchical clustering and k-Medoids (PAM). The third group (time-series + financial time series) is broader than the second and includes autoregressive integrated moving average model (ARIMA), Gaussian processes (GP), and support vector regression (SVR). The fourth group (non-time series + financial time series) includes one sole member, a technique known as classification and regression trees (CART). Finally, the fifth group (financial time series alone) includes group method of data handling (GMDH), logistic regression (LR), Monte Carlo simulation (MCS).

The first group can further be divided into five sub-groups based on the purpose of the ML technique in question. Those that deal with classification and forecasting (sub-group 1) include artificial neural network (ANNs), k-nearest neighbor (KNN),generalized regression neural network (GRNN), long short-term memory (LSTM), multilayer perceptron (MLP), radial basis function neural networks (RBF), random forest (RF), recurrent neural networks (RNN), and support vector machine (SVM). The techniques whose purpose is to forecast solely (sub-group 2) are: back propagation neural network (BPNN) and particle swarm optimization (PSO). The techniques which hope to solely cluster data (sub-group 3) include fuzzy c-means (FCM) and k-means. Those ML techniques which have the triple purposes of clustering, classifying, and forecasting (sub-group 4) include genetic algorithms (GAs) and hidden Markov model (HMM). Finally, one technique deals with clustering and classification only (sub-group 5): self-organizing maps (SOM).

Among the classification and forecasting sub-group of group 1, GRNNs stand out as coming with a lot of advantages and few disadvantages. Indeed, the only notable disadvantage of GRNNs is their huge size which requires a lot of memory space to store; however, their advantages are numerous and include ease-ofimplementation, rapidity of the training process, ability to perform predictions in real-time, and high accuracy. In contrast, ANNs are sensitive to parameter selection, though they also have a high accuracy for modeling the relationship in data groups and the model itself is strong and can deal with noise and incomplete data. KNN is also robust to noisy training data but is computationally expensive and susceptible to sensitivities relating to the arrangement of the data. LSTM is one of the best models in terms of prediction and has a self-learning process, though it comes with a big disadvantage of lacking a procedure with which to save the working memory during data reads and writes. MLP can tackle challenging problems, but its convergence is quite slow and it is difficult to scale. RBF fixes most problems with MLP, being more stable, quicker in convergence than BPNN, faster than MLP, and robust to noisy output, though it lags behind MLP in terms of the classification process. RF is by far one of, if not the, most powerful of these methods, being also able to automatically handle missing values and working well with both discrete and continuous variables; however, it is also one of the most computationally expensive algorithms. RNNs have the unique advantage of displaying the temporal associations occuring between neural network's inputs and outputs, though they are difficult to train. Finally, SVM is very versatile, being able to work on various problems of classification, including high-dimensional variants, but, like ANNs, it is sensitive to parameter selection and also sensitive to outliers.

As for the second sub-group of group 1, the forecasting sub-group, BPNNs show many advantages but also many disadvantages: they have strong adaptability, a fast response, and high learning accuracy, though they are also sensitive to noise, have slow convergent speed, and their actual performance is based on initial values. In contrast, the PSO algorithm is easy to implement, but lacks a solid mathematical foundation, making it not very versatile.

Sub-group 3 of group 1, those techniques that solely cluster data, includes FCM and k-means, which both work well for searching spherical-shaped clusters, but both are also sensitive to noise, and FCM cannot handle high dimensional datasets, while k-means has poor scalability and the clustering quality is remarkably dependent on the initial choice of center elements.

As for sub-group 4, GAs are one of the most powerful algorithms, offering many benefits such as being able to handle noisy data, ability to search clusters with different shapes, and solving some issues of ANNs such as the definition of proper parameters. The only notable issue of GAs, which they also share with ANNs, is their sensitivity to parameter selection. HMM is a very powerful model with a strong statistical foundation and extreme versatility in terms of being able to handle high level information, though it also takes a long time to process and is heavily dependent on user assumptions, thereby introducing bias. Finally, the SOM algorithm, contrary to the GA, is reliable when it comes to the selection of parameters, provides favorable clustering results, and is a first-rate data-exploring tool, though it is sensitive to outliers and performs poorly on time series of differing lengths.

For the second group (time-series + non-time series): hierarchical clustering comes with the advantage of no parameter-setting, but it comes with poor scalability and is useful only for small datasets. In contrast, k-Medoids (PAM) comes with similar advantages to k-Means (searching spherical-shaped clusters), also being more reliable when encountering data noise and outliers as compared to k-Means, though, it, too, like hierarchical clustering, suffers from a scalability issue for larger datasets. With respect to the time series + financial time series group (3), ARIMA is considered to be one of the most competent techniques for forecasting in the fields of social science and works well for linear time series; however, it has little versatility, is slow, and, like HMM, is based on user assumptions. The GP model, in contrast, is quite flexible and also robust, but is also computationally expensive and entails black box compartments which might prove problematic to interpret. In comparison, SVR is well-suited to process multiple inputs, is powerful for financial time-series prediction, and can tackle the overfitting problem, though it is also sensitive to users' parameters.

The fourth group (non-time series + financial time series) includes CART, which can model nonlinearity very well, and, compared to the GP model, its results are very interpretable. However, it is a rather unstable model. Finally, the fifth group (financial time series alone) includes group method of data handling (GMDH), which handles noise very well and has a high accuracy, though it does not consider the input-out relationship well and generates a complicated polynomial even for much simpler systems. Similar to CART, LR can handle nonlinearity well, being able to handle complex nonlinear patterns, although it is sensitive to outliers and must come with strong assumptions. MCS can also model complex systems, is fast, shows which inputs contributed the largest effect on results, and is very versatile. However, MCS is unidirectional, does not allow for the linkage between data and parameters to be interactive, and does not allow for backward reasoning.

2.3 Sentiment Analysis

To better be able to predict the overall attitude of a text writer concerning their topic, the field of natural language processing makes use of sentiment analysis techniques. This type of analysis allows one, more generally, to ascertain whether a text has positive or negative opinions concerning the topic in question. More specifically, sentiment analysis can detect whether or not a specific opinion is spam and whether or not there are several entities being discussed in a text and, if so, how they relate to one another. Of course, these goals are not easily achievable, so the field of sentiment analysis relies on countless novel developments, especially recent advancements in text mining, to help power it.

Opinion mining, which mines texts for positive or negative opinions, can be performed according to three different hierarchical levels: document-level, sentence- and phrase-level, and entity- and aspect-level. Simply put, document-level sentiment analysis assumes at a base level that a single document has either positive or negative sentiments concerning a particularly known target. The goal thus becomes to discover whether the document in question has positive or negative sentiments with regards the target.

The second-order sentiment analysis technique makes use of sentiment analysis to be applied to sentences and phrases. The analysis focuses on specific words, phrases, or sentences, where their sentiments are noted and later on aggregated to have the overall score for the document in question. In terms of sentence-level analysis more specifically, sentences are either considered as being positive, neural, or negative. From this kind of scoring, the overall sentiment of the document can be discovered. However, sentences are not only classified in terms of polarity (positive, neutral, or negative); they are also classified based on their subjectivity and objectivity because both kinds of sentences would be treated differently by the sentiment analysis technique. For example, a sentence such as "This computer comes with 256 GBs of RAM" is considered an objective sentence, thus should usually be labeled as a neutral statement of fact. However, it could be considered a positive statement if all other computers on the market come with less than 128 GBs of RAM. Therefore, these kinds of sentences are barred from influencing the overall result of the document analysis – usually by being removed altogether after a classification system is able to differentiate them from other kinds of sentences. The third-order sentiment analysis technique occurs at the entity- and aspect-level. In this order, it is important to distinguish sentences which relate to the main entity in question from those that do not. For example, if a text about a cleaning product contains the following sentence "Stains are very annoying and are everywhere," this sentence needs to be differentiated from the overall entity of the text. Indeed, while such a sentence is negative, it is actually stressing the exigent need for there to be a cleaning product that can remove said stains.

2.4 The Effect of Sentiment on the Market

As history's wildest stock market swings have proven, it is difficult to reconcile the central economic theory of "homo economicus," or rational man, with these unexplainable variations in the stock market. These stock market upheavals, which defy conventional explanations, include the Great Crash of 1929, the Black Monday crash of October 1987, and, most well-known to investors nowadays, the Dot.com bubble of the 1990s. To reconcile these crashes with economic theory, the field of behavioral finance, which focuses on the psychological, cognitive, and social determinants of financial decisions, has come up with two important assumptions: firstly, that there exists a class of investors who are driven by their sentiment in opposition to the determinants of rationality. The second assumption is that the rational actors who should be able to drive down the prices usually do not do so because they do not meet the aggression of the first group with the same amount of aggression. Instead, as stock market crashes have shown, the sentimental investors drive prices up so high that the rational investors are eventually forced out of business.

Nowadays, the central assumptions of behavioral finance are well-attested to and the question is now how to predict the effects of investor sentiment on the stock market. One psychologically-laden approach is to study how certain cognitive biases or psychological traits can affect investor sentiment; notable among said biases and traits are overconfidence, conservatism, the anchoring effect, confirmation bias, and the recency bias. This approach can then be used to predict the overall trends in the stock market.

More broadly, the research literature has shown that younger, more financially-distressed, and more volatile stocks are more likely to receive investor sentiment because the investors are more likely to see their potentially biased beliefs being confirmed. In addition, these stocks are harder to arbitrage by the more rational investors.

The overall conclusion is thus that the more volatile a stock, the more likely investor sentiment is to play a role. More stable and predictable stocks, such as bond-like stocks, are less likely to garner the interest of sentimental investors and would see many more “arbitrageurs,” or more rational investors who bring speculative prices down, in their ranks. In contrast, volatile stocks would be more appealing to sentimental investors.

Chapter Three

Literature Review

Natural Language Financial Forecasting, i.e. the field of applying natural language processing techniques to predict the performance of financial markets, is a relatively recent field of study with early research efforts emerging between the years 2000 and 2010. Upon deeper investigation, one can also identify even earlier research works dating back as far as the 1960s and 1970s, albeit with a much weaker resemblance to the current form of NLFF and therefore might not truly qualify as related work. This chapter will, however, briefly trace back the origins of the field in an effort to paint a more holistic and comprehensive picture.

Even before the emergence of automated sentiment analysis techniques, quantifying the relationship between market sentiment in the news and the corresponding price movements was already being practiced manually by some financial analysts. In his book published in 1965 [28], Merrill studied the effect of tragic news on stock behavior. He was indeed able to confirm that sentiment in the news is reflected in the stock market. For example, he tracked the effect of the deaths of several presidents each of which causing a bear market in its wake. More closely resembling the current form of the field, Niederhoffer's pioneering 1971 work analyzed the relationship between the sentiment in news headlines and the stock market [31]. As an analog to modern sentiment analysis, his work aimed at ranking headlines on a seven-point good-bad scale in addition to classifying said headlines by untrained observers into twenty categories upon receiving general instructions on how to do so — a precursor to our modern classification techniques. The analysis found that stock market effects had a strong tendency of occurring in the 2 days directly following a significant headline. It was also observed that, during the days 2-5 following an extremely negative event, the stock market experiences a rise in prices correcting for the previous slump.

Edging closer to the 2000s, a 1998 study by Wysocki [40] aimed to ascertain whether or not message-posting volume on the Web was related to stock market activity; a sentiment analysis via investors' enthusiasm was conducted. The relationship was proven to exist as overnight posting volume affected trading volume on the next day; firms that had six key characteristics, with one important characteristic being extreme past returns and accounting performance, experienced a higher volume of messages. In the same year, another study by Wuthrich [39] aimed to predict stock price movements based on news scraped from the web. Sentiment analysis was conducted by counting the occurrences of certain keywords (unigrams, bigrams, trigrams) and assigning weights. The paper assessed several learning techniques, notable among which were rule-based, nearest neighbor, and neural net techniques.

Lavrenko et al., in their 2000 study [20] match news articles with stock market trends. The paper sought to build a language model for each type of trend so as to be able to later classify new articles into a particular type of trend. Following this classification, investors would receive recommendations for articles in which the trend in each article would be made clear by the language model. Thus, investors would see great benefits in terms of the decision-making process. In 2003, Fung [13] noted that there was a lack of research in mining news articles in multiple concurrent times series as opposed to single time series. The work aimed to create a systematic framework that included trend discovery, alignment of articles to time series, feature extraction and feature weighting, time series relationship discovery, and model generation. It would then be possible to predict stock trends based on the newly broadcasted news articles.

Another study that analyzed message boards' effects on markets was Antweiler's work in 2004 [3], which analyzed stock message boards for sentiment. The study used Naive Bayes to classify text as bullish (positive), bearish (negative), or neither. The "naive" assumption was that words were independent, yet it proved quite successful in practice. Fast forward to 2009, Schumaker and Chen developed the AZFinText system [33] for stock market prediction using textual analysis of breaking financial news. They employed a machine learning approach tasked with prediction, using several different textual representations: bag of words, noun phrases, and named entities. Furthermore, their system used a support vector machine (SVM) derivative specifically designed for discrete numeric prediction and suitable for models comprised of different stock-specific variables. The system found that the proper noun scheme outperformed the oft-used bag-of-words models across all metrics.

A later study, in 2011, aimed to predict stock market trends based on social media messages; particularly, Bollen et al. went for predicting stock market trends based on Twitter moods [7]. The study sought to find the correlation between large-scale Twitter feeds and stock prices from the Dow Jones Industrial Average (DJIA) over a certain period of time. As for the methods, the study made use of two tools for tracking moods: OpinionFinder, which measures mood in a polarity based scheme — positive versus negative — and Google-Profile of Mood States (GPOMS), classifying mood states into 6 dimensions (Happy, Calm, Vital, Kind, Sure, and Alert). In a similar vein, Mittal's 2012 work [29] also aimed to predict stock market movements using Twitter sentiment analysis. Sentiment analysis and machine learning methods were used to correlate public sentiment with market sentiment. The study developed this approach by making use of Self Organizing Fuzzy Neural Networks and applying them on Twitter data and Dow Jones Industrial Average values.

A 2014 study by Li et al. [23] attempted to base stock market prediction on news sentiment analysis by constructing a sentiment space from two well-known sentiment dictionaries: the Loughran-McDonald financial sentiment dictionary and the Harvard psychological dictionary. The analysis outperformed bag-of-word models in both the validation set and the independent testing set. That same year, Lee [21] developed a system that predicts changes in companies' stock prices as a reaction to financial events reported in 8-K documents. The approach was complementary to traditional financial (numerical) prediction techniques and helped increase its accuracy by 10%.

Another novel approach, introduced by Li in 2015, used tensor-based learning to predict stock movements [22]. The tensor-based stock information analyzer (TeSIA) ingests stock data and textual data (news and messages from discussion boards) and represents all three information streams in the form of tensors. Tucker decomposition is then performed to eliminate noisy data and identify inherent relations between the different modes in the tensors. The reconstructed tensors are then given to the model to predict movements in the stock prices. Another more recent sentiment analysis technique was applied by Atzeni in 2017 [5]. The study conducted sentiment analysis on financial micro-blogs and headlines from financial news; multiple classifiers were trained on two distinct sets of data. The method was ultimately tested with lexical and semantic features, in addition to a hybrid solution involving both.

In 2018, Chiong conducted sentiment analysis using SVM and PSO on news disclosures to predict financial markets [9]. The results were very promising since the sentiment analysis was conducted during the pre-processing stage, which reduced the feature dimensions by a very large margin. In terms of machine learning approaches, Atkins, in his 2018 work [4], constructed a machine learning model making use of the Latent Dirichlet Allocation technique to incorporate information from Reuters news feeds, and Naive Bayes classifiers to forecast the future price movements in two stock indices: NASDAQ and DJIA. LDA was employed as a means of feature reduction to classify and categorize the news articles into their corresponding topics.

Later in 2019, Li [24] developed a differential privacy-inspired long short-term Memory (DP-LSTM) model for stock market prediction relying on financial news. The model integrates the news snippets as concealed information items and incorporates distinct news streams into the differential privacy mechanism. Using the financial news documents, the sentiment autoregressive moving average model (ARMA) is devised, after which an LSTM-based deep neural network is built consisting of LSTM, VADER, and the DP mechanism.

Another deep learning based work in 2019 was conducted by Souma [34]. The method relied on the Wikipedia and Gigaword corpora and used global vectors for word representation to feed into the TensorFlow deep learning framework. Moreover, the method used RNN and LSTM approaches with stock data from Thomson Reuters and Dow Jones Industrial Average (DJIA). A novel approach by Hiew in 2019 [17], different from the aforementioned dictionary- and machine learning-based approaches, used BERT-based financial sentiment analysis and LSTM-based stock return prediction with a feature of non-linear mapping. The framework so created proved to be more general and comprehensive.

A more general and comprehensive application of machine learning techniques was performed by Lim in 2020 [25], where different machine learning techniques were evaluated in terms of their ability in predicting the sentiment of the readers toward business news headlines. The study compared text classification approaches with recurrent neural network (RNN) approaches. For the text classification approach, multilayer perceptron (MLP), complement naive Bayes, multinomial naive Bayes, and decision trees were analyzed. For the RNN approach, the authors assessed the common RNN architecture and the encoder-decoder architecture in forecasting the sentiment. As one of the most recent endeavors in the field, research by Muthukumar in 2021 [30] forecasted the stock price by using a stochastic time series model for predicting financial trends using NLP. The paper proposed a novel deep learning model known as STGAN (Stochastic Time-series Generative Adversarial Network) to learn correlations among textual and numerical data over time.

Chapter Four

NLP for Stock Market Prediction

Techniques for predicting stock market movements by applying natural language processing on news streams have been increasing in number ever since interested practitioners found potential in such endeavors. Approaches for financial sentiment analysis can be roughly categorized into four classes: probabilistic inferences, regression models, neural networks, or some hybrid technique. Moreover, as a preprocessing step of training for sentiment analysis, such approaches can make use of generic dictionaries, domain-specific dictionaries, or solely rely on statistical and machine learning-based methods. Widely used generic dictionaries include the Harvard GI (General Inquirer) [35] employed in early works such as [36] and [37], and Hart's Diction [14] [15]. However, since the use of general dictionaries resulted in faulty classifications of a significant number of financial keywords, attention turned to domain-specific dictionaries such as the Loughran-McDonald dictionary [27] and the Henry Word List [16], that can achieve superior classification results on financial corpora. For a more holistic coverage of such lists and dictionaries, see [26].

In this chapter, we will flesh out the various techniques we will use. We will explain their inner workings, their strengths and shortcomings, and then empirically compare their performance in the next chapter. We will first, however, lay down the foundational natural language processing and machine learning methods to be used in our work.

4.1 NLP Infrastrutcure

Natural language processing work often consists of a pipeline of several tasks chained together such that the input corpus passes through several stages before any desired outcome is obtained on the other end. Such deployed pipelines often include, but are not restricted to, a morphological analysis stage (stemming, lemmatization, and part-of-speech tagging), a syntactic analysis stage (parsing), and a semantic analysis stage (named entity recognition).

4.1.1 Morphological Analysis

Stemming and Lemmatization

Stemming and lemmatization are text normalization techniques aiming at reducing words, or tokens, to their root or base form so as to achieve consistency among the many possible variations of a word, i.e. its inflected forms. Figure 4.1 showcases the various forms of the word "start" reduced to their common root form. In this case, this same result can be achieved through stemming or lemmatization. For other words, however, this might not be the case.

$$\left.\begin{array}{lcl}\text{starts} & \longrightarrow & \text{start}\\ \text{started} & \longrightarrow & \text{start}\\ \text{starting} & \longrightarrow & \text{start}\end{array}\right\} \quad \text{start}$$

Figure 4.1 Word Normalization

Take for example Figure 4.2 and 4.3, displaying two weakness of stemming, as compared to lemmatization. The former displays over-stemming, where three unrelated words are incorrectly stemmed to the same root form, while the latter shows under-stemming, where related words do not yield the same stem. Both of these incorrect behaviors are rectified when using lemmatization instead of stemming, since the former relies on the vocabulary of the language and can even take into consideration the word's part of speech to better decide which part of the word to trim, if any.

$$\left.\begin{array}{lcl}\text{universe} & \longrightarrow & \text{univers}\\ \text{university} & \longrightarrow & \text{univers}\\ \text{universal} & \longrightarrow & \text{univers}\end{array}\right\}\ \text{univers}$$

Figure 4.2 Over-stemming

$$\left.\begin{array}{lcl}\text{data} & \longrightarrow & \text{dat}\\ \text{datum} & \longrightarrow & \text{datu}\end{array}\right\}\quad \textit{inconsistent}$$

Figure 4.3 Under-stemming

Since lemmatization makes use the language's vocabulary and the word's part of speech, it can normalize seemingly different tokens into their real root form. See for example Figure 4.4, showing that the lemmatization procedure is able to recognize the four different forms of the verb "to be" and reduce them to their root form. Moreover, since lemmatization can look at a word's surrounding context and therefore determine its part of speech, it can, for example, distinguish between the use of the word "meeting" as a noun and as a verb. Consequently, the lemmatizer can refrain from trimming the "-ing" ending if its part of speech was a noun, as opposed to trimming it when it is encountered as a verb, thereby preserving the distinction in meaning.

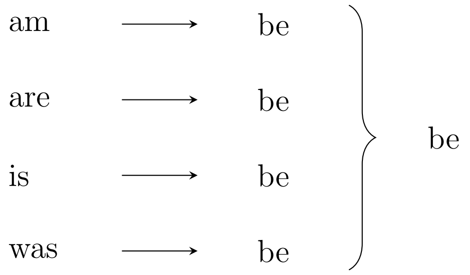


Figure 4.4 Lemmatization

4.2 A Novel Random Forest Approach

The ensemble machine learning technique of employing many decision trees, aggregating them, and averaging out their votes, is known as a random forest model. It was proposed by Breiman in his seminal paper [8], building atop previous work by Amit et al. in [2], work by Ho in [18], and by Dietterich in [10]. In Figure 4.5 below, a single decision tree is shown. See Figure 4.8 for a depiction of a random forest model.

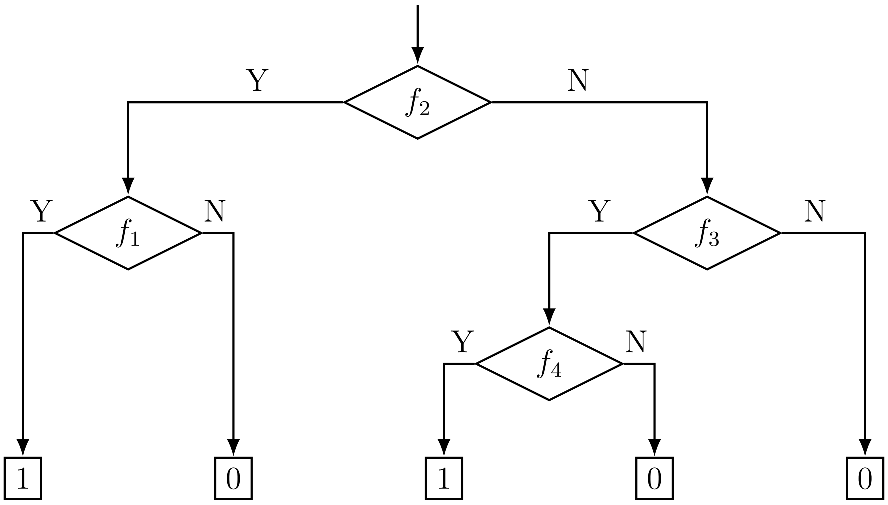


Figure 4.5 Decision Tree in a Random Forest

The decision tree is comprised of decision nodes of a boolean nature such that a soon-to-be-classified data point entering from the root node would be subject to choosing one of two directions labeled Y and N for *yes* and *no* respectively. The decision question at each decision node has to do with a constraint on a certain feature of the data. Following the feature extraction process, the model now has a bank of features to make use of during the learning process. The model then has to try several — or, all possible — value constraints for every available feature. The resulting choice at each node then forms a decision tree to be used in the testing phase or in the actual prediction phase. A new set of data, never before seen by the model, is then given as input to the decision tree, after which we expect the classification mechanism to have been learned.

Normally, for feature extraction, the text is tokenized into single words and is then vectorized. In other words, for each word in each corpus, the frequency of that word's occurrence is registered. The result is a vector representing each text with the frequencies of its word content. The problem, however, is that this approach does not take into consideration the context in which each token appears. The same word can be present in two different contexts and be associated with two opposite market sentiments. This is one of the weaknesses of the vectorization process as described above. For an example, see Figure 4.6 below. The same word "decreased" can have completely different connotations when put in context. Here, the negative sentiment context is highlighted in red: "stock price decreased," whereas the positive sentiment is shown in green: "losses decreased."

Facebook's **stock price decreased** as investors...

Facebook's **losses decreased** year-on-year as investors...

Figure 4.6 Same Word, Different Context

With the traditional vectorization approach, the machine learning model would be confused. For example, the token "decrease" will appear in news items associated with a decrease in stock prices while also appearing in news items associated with an increase in stock prices. For the model to learn a statistical correlation between the two, it will probably lose this feature as it will consider it not really informing. This is where collocation and concordance add value to our approach. Instead of our model vectorizing tokens based on their appearance on their own, we can vectorize bigrams and trigrams that take into account the collocation of certain words. For example: prices increased compared with losses increased, revenue decreased compared with liabilities decreased. These collocations present themselves as much stronger feature candidates for the machine learning model to learn from in the training phase, and then identify these patterns in the testing or deployment phase.

Analysts expected Facebook's stock price to **decrease**, **but it did not**...

Figure 4.7 Concordance: context matters

Further, the context in which a token occurs can play a pivotal role in determining the overall semantics of the sentence. Here we borrow the concept of concordance from linguistics, which is an index of principal word occurences along with their adjacent surrounding context. To give an example, as shown in Figure 4.7, the word "decrease" occurs in a sentence holding positive sentiment when it comes to stock market news. The feature would lose statistical significance since in some sentences it would be correlated with an upward stock trend while in others it would be associated with a downward trend. Including the surrounding context clears up the confusion for the ensemble learning algorithm. It would then be able to distinguish these cases where a seemingly a negative word is strangely associated with a positive trend in the stock market.

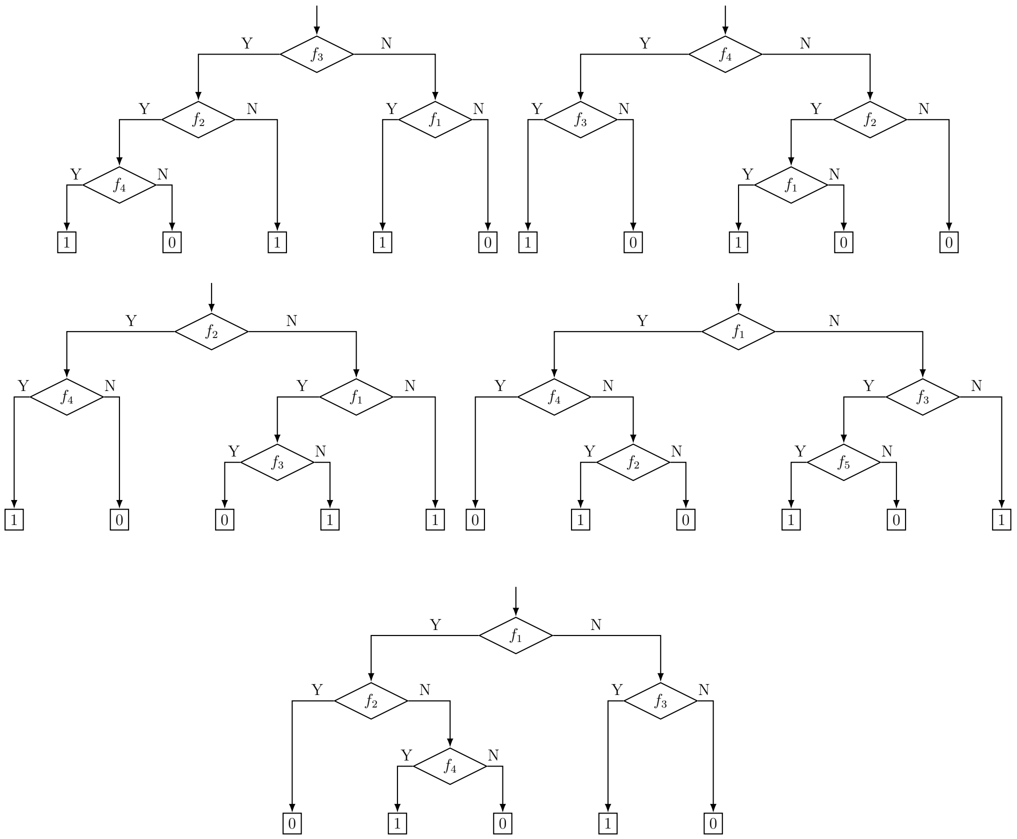


Figure 4.8 Random Forest

Chapter Five

Experimental Analysis

To better determine which of the previously discussed approaches fares better in practice, we conduct the following experimental analysis. We employ each method on the below dataset, and present the relevant results in a comparative format. Results include confusion matrices (true positives, true negatives, false positives, false negatives), accuracy scores, misclassification rates (error rate), precision, recall (sensitivity), specificity, prevalence, and F_1 scores.

5.1 Dataset

We experiment on publicly available data from Yahoo Finance. The dataset is comprised of daily data of stock prices accompanied by daily news headlines. The dataset is an amalgamation of both data sources, aligned with respect to their corresponding dates. The format is thus perfectly suitable for our purpose of predicting stock price movements for a certain day, based on that day's news headlines.

5.1.1 Stocks Data

The stocks portion of the dataset includes stock prices from the Dow Jones Industrial Average (DJIA) from August 08, 2008, till July 01, 2016. Or in ISO format, from 2008-08-08 to 2016-07-01. The data was sourced from Yahoo Finance and is comprised of a date column, columns for Open and Close prices, columns for High and Low prices for the day, and a Volume column. See Table 5.1 for an incomplete data sample showcasing the format. To visualize market movements via candlestick charts, see Figure 5.1 for a market overview throughout 2009, 5.2 for the first quarter of 2012, and 5.3 for the last two quarters of 2015.

5.1.2 News Data

The textual portion of the data consists of a curated set of daily news headlines. Matching each day in our stock prices data, the top 25 news headlines on that day were collected and aligned with the remainder of the dataset. These strings of text were then normalized in several ways. A simple lowercasing was first applied to facilitate later tasks. Lemmatization was then chosen over stemming for its superior capability of returning the correct word roots. The preprocessing stage ended with a final step of punctuation and stop word removal.

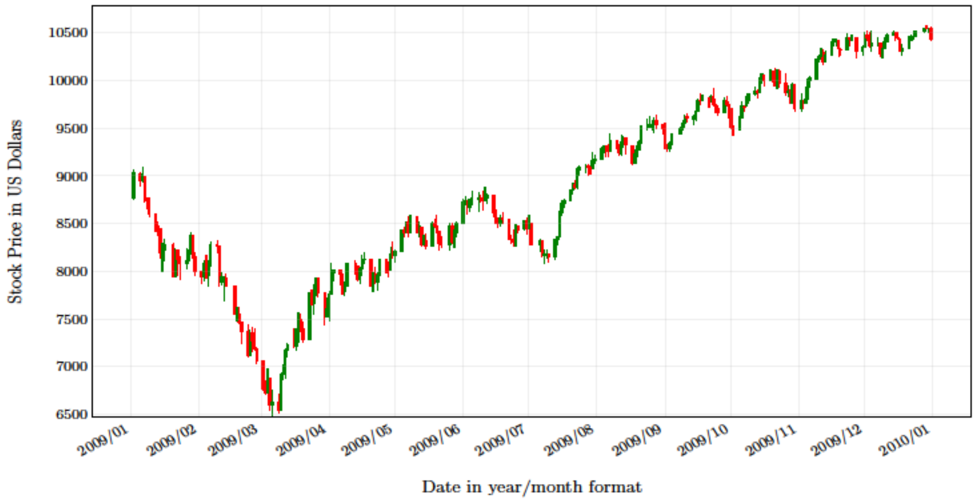


Figure 5.1 Candlestick chart of DJIA movements in 2009

Table 5.1 Stock Price Movements — Dow Jones Industrial Average

Date	Open	High	Low	Close	Volume
2008-08-08	11432.089844	11759.959961	11388.040039	11734.320312	212830000
2008-08-11	11729.669922	11867.110352	11675.530273	11782.349609	183190000
2008-08-12	11781.700195	11782.349609	11601.519531	11642.469727	173590000
2008-08-13	11632.80957	11633.780273	11453.339844	11532.959961	182550000
2008-08-14	11532.070312	11718.280273	11450.889648	11615.929688	159790000
2008-08-15	11611.209961	11709.889648	11599.730469	11659.900391	215040000
2008-08-18	11659.650391	11690.429688	11434.120117	11479.389648	156290000
2008-08-19	11478.089844	11478.169922	11318.50	11348.549805	171580000
2008-08-20	11345.94043	11454.150391	11290.580078	11417.429688	144880000
2008-08-21	11415.230469	11476.209961	11315.570312	11430.209961	130020000
2008-08-22	11426.790039	11632.129883	11426.790039	11628.05957	138790000
...	...	...	...	...	...
2016-06-17	17733.439453	17733.439453	17602.779297	17675.160156	248680000
2016-06-20	17736.869141	17946.359375	17736.869141	17804.869141	99380000
2016-06-21	17827.330078	17877.839844	17799.800781	17829.730469	85130000
2016-06-22	17832.669922	17920.160156	17770.359375	17780.830078	89440000
2016-06-23	17844.109375	18011.070312	17844.109375	18011.070312	98070000
2016-06-24	17946.630859	17946.630859	17356.339844	17400.75	239000000
2016-06-27	17355.210938	17355.210938	17063.080078	17140.240234	138740000
2016-06-28	17190.509766	17409.720703	17190.509766	17409.720703	112190000
2016-06-29	17456.019531	17704.509766	17456.019531	17694.679688	106380000
2016-06-30	17712.759766	17930.609375	17711.800781	17929.990234	133030000
2016-07-01	17924.240234	18002.380859	17916.910156	17949.369141	82160000

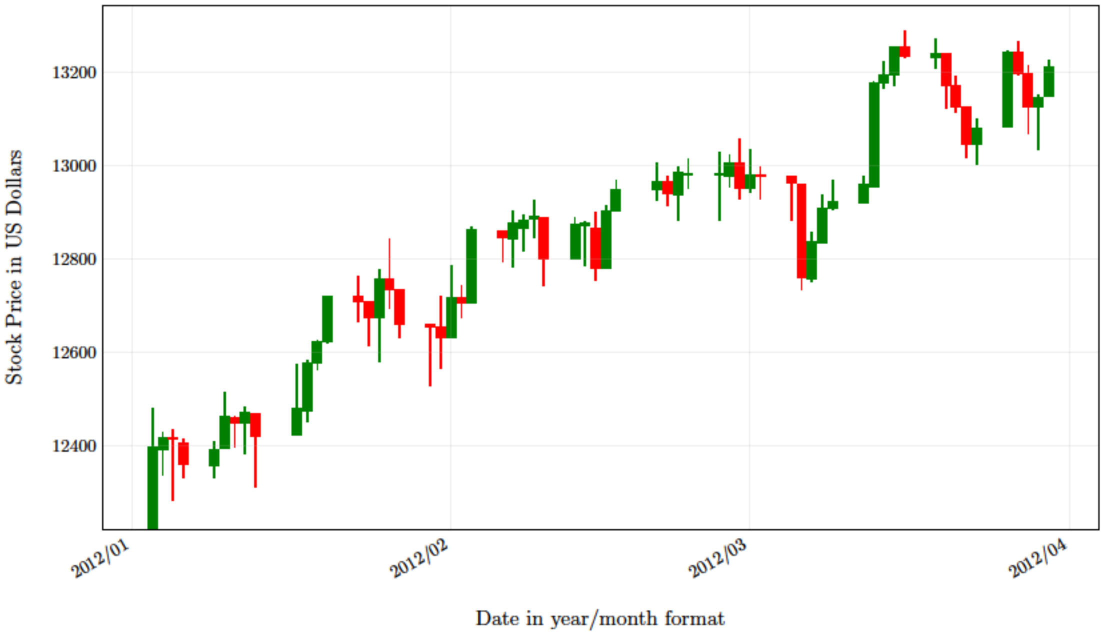


Figure 5.2 Candlestick chart of DJIA movements in Q1 of 2012

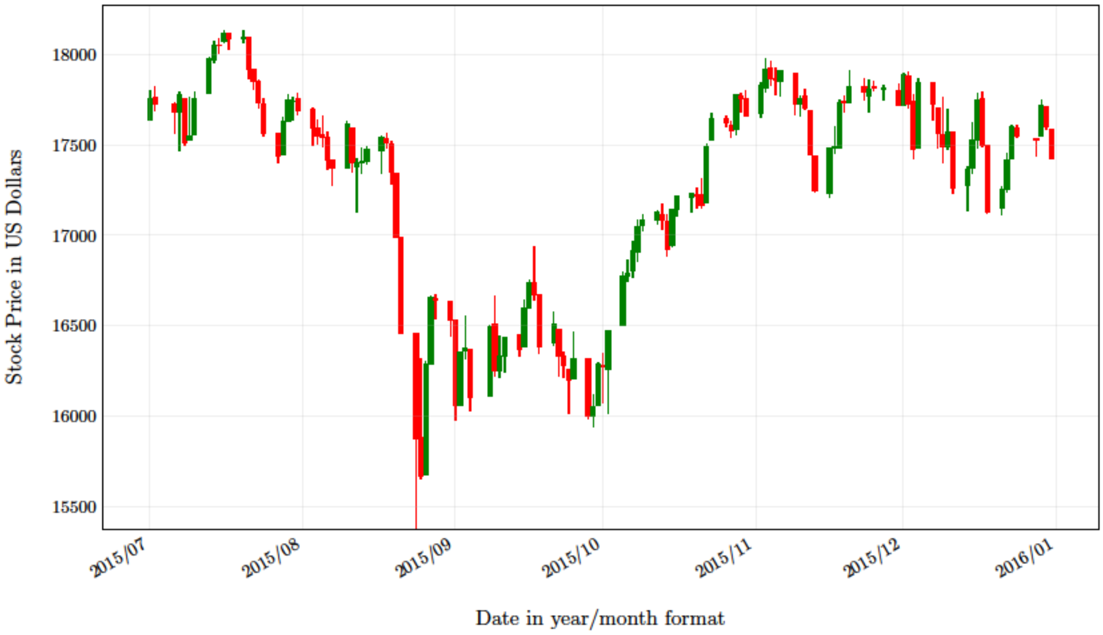


Figure 5.3 Candlestick chart of DJIA movements in Q3 and Q4 of 2015

5.2 Experimentation & Results

For predicting the stock price movements from each day's top news headlines, and after going through the preprocessing steps of text normalization (lowercasing, lemmatization, punctuation and stop word removal), we split the dataset into a train set and a test set, as is customary for such applications. We elect a split point at the date of 2015-01-01. We therefore have a training dataset spanning from the date of 2008-08-08 till 2014-12-31, i.e. six years and roughly four months. The test dataset then spans from 2015-01-01 until 2016-07-01 — a time period of one year and a half. Naturally, the train set is the portion that gets seen by the model and trained on, while the test dataset consists of "never before seen" data for the model that serve as a simulation of a realistic scenario for testing the performance of the model "in the wild." Following these steps, we then collect each day's top 25 headlines into one string ready for ingestion by our classification model — the Random Forest algorithm. Our approach is then expected to turn the daily news corpus into a numerical representation based on the n-gram parameter specified beforehand. A text vectorizer takes the daily corpus as input and transforms it into a numerical vector representation. If the n-gram parameter is set to 1, then the vectorizer only considers unigrams; that is, it only considers single word tokens as features. On the other hand, one can specify a higher number for the n-gram parameter, such that bigrams or even trigrams can be considered as features for our classification model to work on in the next phase of our approach. Indeed, this method of simplifying the corpus representation and transforming it into a numerical format is the bag-of-words model, where the features to be extracted from the text are the frequencies of each token's occurrence in said text. Note that the token here can be a unigram, bigram, trigram, or a higher n-gram variant. At this point, our dataset is ready for the final phase. We can now use the current form of our dataset as an input for our random forest classification model.

Some of the parameters that the random forest model requires are the number of estimators — the number of trees in the forest — and the criterion parameter which specifies the method to use to assess the quality of a split that the model has to make at a certain node based on its impurity. The first of the two options for the criterion parameter are Gini (or the Gini index) represented by Equation 5.1 below, where p_j is the probability of choosing an item from class j. The Gini index therefore measures the probability of mislabeling an element of the dataset when it is labeled at random.

$$GiniIndex = 1 - \sum_j p_j^2 \tag{5.1}$$

The alternative method used as a measure of impurity or information-gain is Entropy, displayed in Equation 5.2 below, where p_j remains the probability of choosing an element from class j.

$$Entropy = -\sum_j p_j \log_2 p_j \tag{5.2}$$

The other model parameter to consider is the number of estimators in our ensemble technique, i.e. the random forest algorithm. In general, since the random forest algorithm is an ensemble technique which relies on averaging out over many decision trees, then the more we use trees, the better our results will get. However, benefits only accrue up to some point where adding further trees does not result in better accuracies, in addition to the accumulation of performance penalties that make it hard to justify the addition of more and more decision trees to the model. As will be apparent in the following results we obtained, diminishing returns is very much a phenomenon that occurs with increasing the number of estimators in a random forest model.

We first try the model by only considering unigrams in the text vectorization phase and with only 5 decision trees with the criterion set to be entropy. The classification report is shown below in Table 5.2, indicating an accuracy of 81%. Precision, recall, and the F_1 score are also shown for predictions of negative price changes, and predictions of non-negative price changes, i.e. either the price increased day-on-day, or stayed the same.

Table 5.2 Random Forest Classification Report 1

	precision	recall	F_1 score	support
negative price change	0.80	0.81	0.81	186
non-negative price change	0.81	0.80	0.81	192
accuracy			0.81	378
macro avg	0.81	0.81	0.81	378
weighted avg	0.81	0.81	0.81	378

We then experiment by increasing the number of decision trees used in our model — the number of estimators. Tables 5.3, 5.4, and 5.5 below, show classification reports with the number of estimators parameter set to 10, 20, and 30, respectively. As is apparent from these results, each jump in the number of decision trees used resulted in a 1% increase in classification accuracy.

Table 5.3 Random Forest Classification Report 2

	precision	recall	F_1 score	support
negative price change	0.79	0.85	0.82	186
non-negative price change	0.84	0.79	0.81	192
accuracy			0.82	378
macro avg	0.82	0.82	0.82	378
weighted avg	0.82	0.82	0.82	378

Table 5.4 Random Forest Classification Report 3

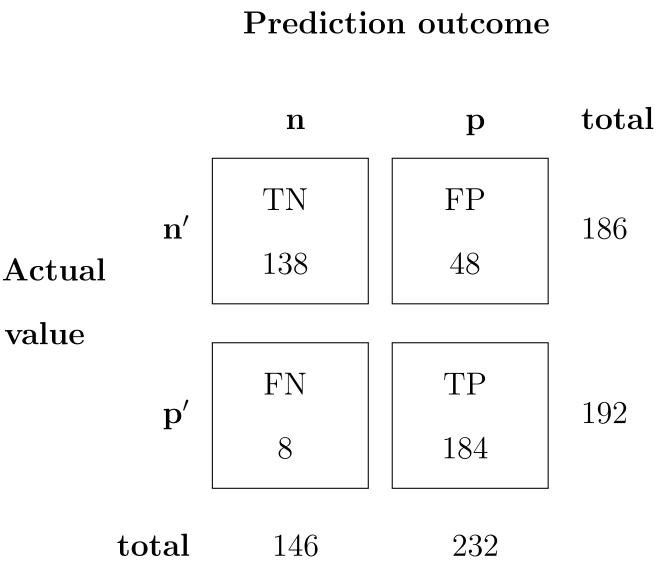
	precision	recall	F_1 score	support
negative price change	0.83	0.83	0.83	186
non-negative price change	0.83	0.83	0.83	192
accuracy			0.83	378
macro avg	0.83	0.83	0.83	378
weighted avg	0.83	0.83	0.83	378

Table 5.5 Random Forest Classification Report 4

	precision	recall	F_1 score	support
negative price change	0.84	0.82	0.83	186
non-negative price change	0.83	0.85	0.84	192
accuracy			0.84	378
macro avg	0.84	0.84	0.84	378
weighted avg	0.84	0.84	0.84	378

To squeeze out the last bits of performance from our model, we can include bigrams and trigrams to be treated as features to enhance the classification process. Table 5.6 shows the results where bigrams are taken into consideration, and Figure 5.4 shows the confusion matrix.

Figure 5.4 Random Forest Confusion Matrix



		Prediction outcome		
		n	**p**	**total**
Actual value	**n′**	TN 138	FP 48	186
	p′	FN 8	TP 184	192
	total	146	232	

Table 5.6 Random Forest Classification Report 5

	precision	recall	F_1 score	support
negative price change	0.95	0.74	0.83	186
non-negative price change	0.79	0.96	0.87	192
accuracy			0.85	378
macro avg	0.87	0.85	0.85	378
weighted avg	0.87	0.85	0.85	378

With all the aforementioned tweaks incorporated into our model, the algorithm achieves an accuracy of 85%, higher than all previous attempts. The confusion matrix sheds some light on the strengths and weaknesses of our classification process. It displays the performance in terms prediciton outcomes versus the actual values. The following abbreviations were used: TN for true negatives, FN for false negatives, TP for true positives, and FP for false positives. Our approach is stronger in predicting negatives, as can be seen with the low number of false negatives. On the other hand, it has a weakness, at least on this particular dataset, in predicting positives, as can be seen with a relatively high number of false positives.

Chapter Six

Conclusion and Future Work

Owing to the rewarding nature of predicting stock market movements, several works have attempted distinct methods to ultimately reach higher accuracy scores. We join forces with the field of linguistics to develop a novel approach — an amalgamation of the random forest classifier and the two concepts of collocation and concordance. This fusion, which, to the best of our knowledge, has not been previously attempted in the literature, resulted in an impressive 85% accuracy in predicting stock price movements based on sentiment patterns detected in the news corpora.

We evaluate the performance of our model in different conditions by tuning the hyperparameters and observing the effects of each change. As expected, by increasing the number of estimators in our model, or, in other words, by increasing the number of decision trees in our random forest classifier, we note an increase in prediction accuracy. We further investigate by alternating between the entropy criterion and the gini index criterion as another hyperparamter for our approach. Comparing the results, we observe that the addition of the mechanisms associated with collocation and concordance did indeed contribute to the increased accuracy score.

An additional step in our experimental analysis was comprised of inspecting the confusion matrix for our classification model. Our approach was found to be susceptible to a significant dose of false positive predictions, while remaining reliable in terms of negative predictions as shown by the low number of false negatives in the confusion matrix.

It would be interesting to try combining our approach with algorithms other than the random forest classifier. The often-used logistic regression, for example, could potentially benefit from taking into consideration collocations and concordance. Deep learning based approaches such as recurrent neural networks in general, or long short-term memory in particular, or even convolutional neural networks might also be fertile ground for the seeds of linguistic methods such as the ones we used in our work. This thesis was a step in the right direction of joining the fields of computer science and linguistics under the umbrella of natural language processing and sentiment analysis; it remains to be seen if further steps in that direction will be taken to further advance the state-of-the-art.

Bibliography

[1] Charu C Aggarwal. “Opinion mining and sentiment analysis”. In: *Machine learning for text*. Springer, 2018, pp. 413–434.

[2] Yali Amit and Donald Geman. “Shape quantization and recognition with randomized trees”. In: *Neural computation* 9.7 (1997), pp. 1545–1588.

[3] Werner Antweiler and Murray Z Frank. “Is all that talk just noise? The information content of internet stock message boards”. In: *The Journal of finance* 59.3 (2004), pp. 1259–1294.

[4] Adam Atkins, Mahesan Niranjan, and Enrico Gerding. “Financial news predicts stock market volatility better than close price”. In: *The Journal of Finance and Data Science* 4.2 (2018), pp. 120–137.

[5] Mattia Atzeni, Amna Dridi, and Diego Reforgiato Recupero. “Fine-grained sentiment analysis on financial microblogs and news headlines”. In: *Semantic Web Evaluation Challenge*. Springer. 2017, pp. 124–128.

[6] Malcolm Baker and Jeffrey Wurgler. “Investor sentiment in the stock market”. In: *Journal of economic perspectives* 21.2 (2007), pp. 129–152.

[7] Johan Bollen, Huina Mao, and Xiaojun Zeng. “Twitter mood predicts the stock market”. In: *Journal of computational science* 2.1 (2011), pp. 1–8.

[8] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[9] Raymond Chiong et al. “A sentiment analysis-based machine learning approach for financial market prediction via news disclosures”. In: *Proceed-*

ings of the Genetic and Evolutionary Computation Conference Companion. 2018, pp. 278–279.

[10] Thomas G Dietterich. “Ensemble methods in machine learning”. In: *International workshop on multiple classifier systems.* Springer. 2000, pp. 1–15.

[11] Eugene F Fama. “Efficient market hypothesis”. In: *Diss. PhD Thesis, Ph. D. dissertation* (1960).

[12] Eugene F. Fama. “Efficient Capital Markets: A Review of Theory and Empirical Work”. In: *The Journal of Finance* 25.2 (1970), pp. 383–417. ISSN: 00221082, 15406261. URL: `http://www.jstor.org/stable/2325486`.

[13] G Pui Cheong Fung, J Xu Yu, and Wai Lam. “Stock prediction: Integrating text mining approach using real-time news”. In: *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.* IEEE. 2003, pp. 395–402.

[14] Roderick P Hart. “Redeveloping DICTION: theoretical considerations”. In: *Progress in communication sciences* (2001), pp. 43–60.

[15] Roderick P Hart and Craig Carroll. *DICTION: The text-analysis program.* 2011.

[16] Elaine Henry. “Are investors influenced by how earnings press releases are written?” In: *The Journal of Business Communication (1973)* 45.4 (2008), pp. 363–407.

[17] Joshua Zoen Git Hiew et al. “BERT-based financial sentiment index and LSTM-based stock return predictability”. In: *arXiv preprint arXiv:1906.09024* (2019).

[18] Tin Kam Ho. “The random subspace method for constructing decision forests”. In: *IEEE transactions on pattern analysis and machine intelligence* 20.8 (1998), pp. 832–844.

[19] B Shravan Kumar and Vadlamani Ravi. “A survey of the applications of text mining in financial domain”. In: *Knowledge-Based Systems* 114 (2016), pp. 128–147.

[20] Victor Lavrenko et al. “Language models for financial news recommendation”. In: *Proceedings of the ninth international conference on Information and knowledge management*. 2000, pp. 389–396.

[21] Heeyoung Lee et al. “On the Importance of Text Analysis for Stock Price Prediction.” In: *LREC*. Vol. 2014. 2014, pp. 1170–1175.

[22] Qing Li et al. “Tensor-based learning for predicting stock movements”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.

[23] Xiaodong Li et al. “News impact on stock price return via sentiment analysis”. In: *Knowledge-Based Systems* 69 (2014), pp. 14–23.

[24] Xinyi Li et al. “DP-LSTM: Differential privacy-inspired LSTM for stock prediction using financial news”. In: *arXiv preprint arXiv:1912.10806* (2019).

[25] Seong Liang Ooi Lim et al. “Examining machine learning techniques in business news headline sentiment analysis”. In: *Computational Science and Technology*. Springer, 2020, pp. 363–372.

[26] Tim Loughran and Bill McDonald. “The use of word lists in textual analysis”. In: *Journal of Behavioral Finance* 16.1 (2015), pp. 1–11.

[27] Tim Loughran and Bill McDonald. “When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks”. In: *The Journal of finance* 66.1 (2011), pp. 35–65.

[28] Arthur A Merrill. *Behavior of prices on Wall Street*. Analysis Press, 1965.

[29] Anshul Mittal and Arpit Goel. “Stock prediction using twitter sentiment analysis”. In: *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)* 15 (2012).

[30] Pratyush Muthukumar and Jie Zhong. “A stochastic time series model for predicting financial trends using nlp”. In: *arXiv preprint arXiv:2102.01290* (2021).

[31] Victor Niederhoffer. “The analysis of world events and stock prices”. In: *The Journal of Business* 44.2 (1971), pp. 193–219.

[32] Mehtabhorn Obthong et al. “A survey on machine learning for stock price prediction: algorithms and techniques”. In: (2020).

[33] Robert P Schumaker and Hsinchun Chen. “Textual analysis of stock market prediction using breaking financial news: The AZFin text system”. In: *ACM Transactions on Information Systems (TOIS)* 27.2 (2009), pp. 1–19.

[34] Wataru Souma, Irena Vodenska, and Hideaki Aoyama. “Enhanced news sentiment analysis using deep learning methods”. In: *Journal of Computational Social Science* 2.1 (2019), pp. 33–46.

[35] Philip J Stone et al. “The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information”. In: *Behavioral Science* 7.4 (1962), p. 484.

[36] Paul C Tetlock. “Giving content to investor sentiment: The role of media in the stock market”. In: *The Journal of finance* 62.3 (2007), pp. 1139–1168.

[37] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. “More than words: Quantifying language to measure firms’ fundamentals”. In: *The journal of finance* 63.3 (2008), pp. 1437–1467.

[38] Hrishikesh Vachhani et al. “Machine learning based stock market analysis: A short survey”. In: *International Conference on Innovative Data Communication Technologies and Application.* Springer. 2019, pp. 12–26.

[39] Beat Wuthrich et al. “Daily stock market forecast from textual web data”. In: *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218).* Vol. 3. IEEE. 1998, pp. 2720–2725.

[40] Peter D Wysocki. “Cheap talk on the web: The determinants of postings on stock message boards”. In: *University of Michigan Business School Working Paper* 98025 (1998).

[41] Frank Z Xing, Erik Cambria, and Roy E Welsch. “Natural language based financial forecasting: a survey”. In: *Artificial Intelligence Review* 50.1 (2018), pp. 49–73.