Natural Language Processing as a Predictive Feature in Financial Forecasting

By Jaebin (Jay) Chang
Thesis Advisor: Mark Liberman
Academic Advisor: Rajeev Alur

EAS499 Senior Thesis
School of Engineering and Applied Science
University of Pennsylvania
April 28th 2020

Table of Contents

1. Introduction

With no doubt, the stock market serves an important function in the overall economy. It not only works as a source of funds for businesses but also can be a main vehicle of wealth creation for individual and institutional investors. Specially, stocks allow small investors to participate in gains of companies through partial ownership. It is attractive because it generates average returns beyond the inflation rate, also serving as a good means to store wealth.

However, it is important to note that the stock market is not the economy itself but rather a good indicator of it. Its price is determined by the supply and demand of investors who are heavily influenced by market psychology and the public mood. A lot of research like [13] and [15] have shown the effect of public sentiment on financial performance. And this notion has served as the bedrock, on which natural language processing based financial forecasting techniques were developed. Natural language processing allows users to extract textual components or sentiments embedded in a piece of textual data. Many of the research related to this domain utilize the extracted features to predict stock price performance.

In this paper, we explore implementations of various NLP-based financial forecasting techniques and classify them according to common characteristics. In Section 2, we first discuss the core concepts related to the financial market, such as Efficient Market Hypothesis, and different perspectives towards investing. Then, we explore different algorithmic approaches to financial forecasting and show some of the mathematical concepts related to each approach. Next, Section 4 dives into details on how NLP-based prediction models are set up and run. To conclude, we discuss some of the limitations faced by researchers in this domain.

2. Understanding the Financial Market

2-a. Efficient Market Hypothesis and Random Walk Hypothesis

A critical foundation of modern stock market analysis is the efficient market hypothesis ("EMH") by Eugene Fama [1]. According to the hypothesis, the stock price immediately reflects all information publicly available [1]. However, it should be familiar to anyone who has ever invested his or her wealth in the stock market that the market is not as ideal as the EMH proposes. To tie the concept to reality, Eugene Fama stated that the market condition can exist in one of the three forms of EMH, namely strong, semi-strong, and weak [2]. First, the strong version describes a market condition, in which security prices reflect all sorts of information from public and historical data to private information [2]. Also, the form presumes that there are no costs to trade and get access to information, which is rarely true [2]. Fama acknowledged that this type of market rarely exists and is useful for theoretical purposes [2]. The next type is the semi-strong form [2]. In this form, stock price reflects all public and historical information, while private information does not affect price movements [2]. Finally, the weak form presumes that historical information is reflected, while any current or private information fails to influence the market [2].

A financial theory that goes hand in hand with the EMH is Random Walk Hypothesis, which was first applied to financial markets by Paul Samuelson in 1965 [3]. According to the hypothesis, asset prices move in random directions and thus are impossible to predict [4]. This theory has some resemblance to the semi-strong version of the EMH in that it presumes all public information is accessible to every market participant [5]. Samuelson stated that the randomness is achieved by the greed of investors who rely on all sorts of information to drive excess return [3]. By actively trading on new information every minute they are generated, the investors effectively "bake" the new information into the stock price, driving away profit opportunities as they do so [3]. Hence, the market cannot be predicted with already existing sets of information [6]. This idea is now very popular among academics and industry professionals, as the EMH and random walk hypothesis "have become icons of modern financial economics [6]." However, some experts challenge this idea and claim that the market can somewhat be predicted. Researchers like Qian et al.; Gallagher and Taylor; and Butler and Malaikah have demonstrated that the market can be predicted to some extent [7], [8], [9]. Qian et al. in 2007 showed that the Dow Jones Industrial Average index can be predicted with an accuracy of 65% by using a combination of multiple machine learning classifiers [7]. Gallagher and Taylor showed that there is a significant negative correlation between "inflation due purely to supply innovation" and real stock returns [8]. Butler and Malaikah, on the other hand, demonstrated that thinly-traded markets like ones in Kuwait and Saudi Arabia show "a significant departure from the random walk [9]."

Among non-random walk believers, there are those who argue that the stock market can be predicted because new information can be inferred from a pool of raw data even before the information comes out in the market. This camp is where natural language processing plays a

huge role. Gruhl et al. showed how sales of a product like a book can be predicted by counting the number of mentions of the product in online communities [10]. A similar research was conducted by Mishne and Glance in 2006, in which they predicted movie sales by utilizing sentiment analysis on blog data [11]. As another example, Choi and Varian demonstrated that it is possible to predict major economic indicators like unemployment claims and consumer confidence by using Google Trends data [12]. Although these approaches do not directly forecast stock price movements, they indirectly do so by predicting critical information that if released would influence the stock market. Recently, more research has been conducted to directly forecast stock market movements from publicly available textual data. For example, Bollen et al. showed that sentiment indicators extracted from twitter data can be a good predictor of daily stock price movements with accuracy of 87.6% [13]. Another famous example is the AZFinText system developed by Schumaker and Chen, which predicted stock movements with a directional accuracy of 57.1% by performing machine learning algorithms on textual representations, such as a bag of words, named entities, and noun phrases [14]. Similarly, Li et al. took a step further by utilizing both sentiment analyzer and news representation techniques to predict stock price movements [15]. All three literature will be covered in detail in later sections. As a more direct evidence of short term predictability, Gidofalvi showed that there is a 20 minute window of opportunity until a piece of information gets reflected on the stock market [16]. The result of this research is intuitive given that it takes time for investors to process new information.

2-b. Fundamental Approach vs. Technical Approach

The notion that the market can be predicted gave ways to mainly two approaches in investing: fundamental and technical approach. The fundamental approach assumes that the stock price of a company is driven by the performance and strategic positioning of the company [14]. Therefore, fundamental investors utilize various performance-related metrics, such as earnings per share, profit margin, revenue growth rate, and price over earnings ratio, to determine the attractiveness of the stock and predict its direction [17]. With these metrics, fundamental investors assign an "intrinsic value" to a security and compare that to the market value (or stock price) of the security [17]. If the intrinsic value is lower than the market value, then the security would be considered cheap and would signal a buying opportunity. Because the intrinsic value may change due to new information about the company, research like [10], [11], and [12] that utilize natural language processing to predict certain key indicators have proven to be useful in fundamental approaches. Currently, this method serves as the bedrock, on which many institutional investors like private equity funds, hedge funds, and mutual funds manage their money.

On the opposite end of the spectrum is the technical approach. It is deeply rooted in the belief that there are "trends" in stock prices that can be used to predict future stock prices [14]. Therefore, technical investors indirectly go against the concept of the random walk theory by claiming that although the random walk theory is valid, prices move in trends determined by market psychology and supply and demand of the underlying securities [18]. Moreover,

technical investors believe that the market already reflects all information related to fundamentals and thus support the EMH [18]. Therefore, to them, intrinsic values do not give any valuable insights into future stock prices. Research like [13], [14], and [15] can be classified as an extension of technical analysis, although they don't mention it explicitly on their paper. All three of them perform some sort of trend analysis on stock quotes over a certain period of time [13], [14], [15]. Ultimately, most research that utilizes sentiment analysis to predict stock prices, in my opinion, can be classified as technical analysis because the implicit value of a company should not change based on the sentiments of the public on its stock price. Also, the technical approach has close connections to market psychology, which is often the main outcome of any sentiment analysis.

3. Algorithmic Approaches to Financial Forecasting

There were many attempts to come up with an algorithm to predict stock price performance. Based on Zhang et al.'s work [19] on comparing different approaches to financial forecasting, the following segment is divided into three categories - historical price based, cross-correlation based, and natural language processing ("NLP") based approaches. NLP based approaches can be further broken down into two subcategories based on what was used as the main feature - textual components and sentiment score. Finally, an interesting approach of using a business network and an energy cascade model by Zhang et al. will be discussed as an extension of NLP approach. Because research in the area of financial forecasting focuses on reporting better implementations of certain techniques (as opposed to discussing theoretical concepts), for each section, a few highly cited papers were selected to represent how the approach is normally implemented.

3-a. Historical Price Based Approach

Analogous to how the technical investing style relies on historical prices to forecast future prices, some algorithmic approaches use past prices as the main feature in their models. Like the technical investing style, this approach assumes that there is some form, linear or nonlinear, of relationship among prices over time. A popular approach that assumes a linear relationship is called the autoregressive integrated moving average or ARIMA, which is an extension of a time series prediction technique called the autoregressive moving average or ARMA [30]. ARMA takes in two parameters k and q and produces an equation for prediction as follows:

$$X_t = \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \sum_{i=1}^{k} \alpha_i X_{t-i} + \epsilon_t \quad \text{[30]}$$

Here, k and q respectively represent autoregression and moving average coefficients. X_t is the predicted value, while X_{t-i} are past values at time t-i. $\mathcal{E}_t$ represents the zero-mean noise term (or random error [29]) at time t, and α_i and β_i are linear regression coefficients for variables X_{t-i} and $\mathcal{E}_{t-i}$ [30]. In summary, this technique assumes that the current value of X is a linear combination of past k X values and past q random error values. And accordingly, ARMA(k, q) is a good way to predict values for stationary stochastic processes [30].

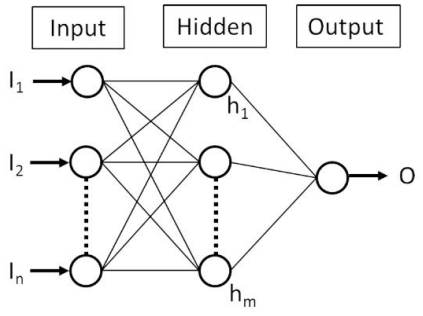
However, real world time series data, such as stock price data, are rarely stationary. Rather, they "may contain deterministic trends" that can be turned into stationary processes through a differential method [31]. For these kinds of time series data, ARIMA is needed to make predictions. As the name implies, ARIMA extends ARMA through an "integration" process [30]. Instead of treating a time series data as is, it transforms its values into one or more orders of differentials [30]. The equation for ARIMA is as follows:

$$\nabla^d X_t = \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \sum_{i=1}^{k} \alpha_i \nabla^d X_{t-i} + \epsilon_t$$ [30]

As it can be easily noted, the only difference from ARMA is the differential sign, ∇^d, where d represents the order of difference and $\nabla^d X_t = \nabla^{d-1} X_t - \nabla^{d-1} X_{t-1}$ [30]. It is also apparent that an ARIMA of k and q with difference order of zero is effectively the same as an ARMA of k and q [30].

According to Adebiyi et al., the ARIMA model is especially useful and robust in short term predictions and is "extensively used in the field of economics and finance [29]." In their research, they built an ARIMA prediction model for stock prices of Nokia and Zenith Bank [29]. For each stock, an Augmented Dicky Fuller (ADF) unit root test was run to identify the order of difference that will make the time series stationary [29]. For Nokia, it was one; for Zenith Bank, two [29]. Likewise, different combinations of parameters k and q were tried with the ARIMA model to identify values that would produce the smallest Bayesian information criterion and standard error of regression [29]. Accordingly, the best models for Nokia and Zenith Bank respectively were ARIMA (2, 1, 0) and (1, 0, 1) [29]. These models were used to predict stock price movements of Nokia and Zenith Bank [29]. The authors demonstrated the models' effectiveness through graphical representations and noted them to be "satisfactory [29]."

Another widely used approach that relies on historical price data is called the artificial neural network or ANN. In contrast to the ARIMA model, this approach has "the potential to capture complex nonlinear relationships between a group of individual features and the variable being forecasted [26]." A popular ANN model in the domain of financial forecasting is the back propagation neural network, which is a supervised machine learning technique that minimizes a predetermined error function through a gradient descent [32]. As shown in the figure below, an ANN model is composed of multiple nodes or neurons that together form the input, hidden and output layers [31], [32].



[32]

An edge between each node or "link" has a weight that represents the strength of the connection between the two endpoints [32]. An ANN model changes the values of these weights so that the error function, which is usually mean squared error, is minimized [32]. Once the weights have been set, the model can take in unknown data to predict output values [32]. One danger of running an ANN model is overfitting the training dataset, which would make the model more of a memorization model than a prediction model [31], [32]. However, it can be

prevented by limiting the number of hidden layers to one [31], starting with a few of neurons and moving your way up [33], or using mathematical techniques like the Bayesian regularization [32].

Several research papers have been devoted to predicting the stock market with this approach. Most notably, Ticknor in 2013 created a "Bayesian regularized artificial neural network" that addressed the issue of overfitting in stock market predictions and returned mean absolute percentage errors or MAPEs of 1.0507% and 1.4860% in predicting stock price movements of Microsoft and Goldman Sachs respectively [32]. He also showed that the model is "as good as both the fusion model and the ARIMA model in Hassen et al. [34], [32]." In Zhang et al.'s research, a neural network was set up to be compared with the results of other approaches [19]. Their neural network showed directional accuracies of 54.8% and 55.1% for upward and downward movements, which turned out to be far lower than other methods explored [19]. In a research by Kwon and Moon, it was shown that an ANN model can be combined with a genetic algorithm to produce outstanding performance over the traditional "buy and hold" strategy [26].

3-b. Cross-correlation Based Approach

Another common way to predict stock price movements is to use a cross-correlation based approach. It is based on the belief that "assets in similar sectors will have similar behavior due to the fundamental environment [20]." Such belief has been well supported by researchers like Anton and Polk, who through their research showed that "stocks are connected through their common fund ownership [21]." They also found that the covariance of these stocks are greater for smaller companies than larger ones [21]. Xing et al. also attributed connected movements to limited attention investors have, which allows certain information about a specific firm to affect other firms as well [20]. Moreover, there were studies that demonstrated that predictions using the random matrix theory can represent the cross-correlations among companies fairly well [23] [24], indicating that "there exist cooperative behaviors of the entire market [25]."

In this approach, the stock volatility of one stock and its correlation with a connected stock are used to predict the volatility of the connected stock [19]. Zhang et al. achieved this kind of prediction by calculating the Pearson correlation between a pair of two connected stocks [19]. A Pearson correlation, sometimes called just as "correlation" or "Product Moment Correlation Coefficient," is a correlation metric that is commonly used to show linear relationship between two variables [22]. Its formula is simply "dividing the covariance by the product of the standard deviations [22]." Using this metric, Zhang et al. identified the top five firms with the highest correlation to a testing firm based on training stock price data [19]. Then, the directional movements of the top five firms were used as input to train a support vector machine (SVM) classifier, and the trained classifier was used to predict stock price directions of the testing firm based on testing data [19]. The research utilized "a library for support vector machines (LIBSVM) with a radial basis function (RBF) kernel [19]." Further details on SVM will

be covered in a later section. But, Zhang et al. claimed that this approach can be extended to competitive relationships [19]. If one stock is a competitor of another in certain dimensions, the stock's upward movement would induce the downward movement on the other [19]. However, Zhang et al. acknowledged that this approach is limited in that it only captures one side of many relationships a company could have [19]. Therefore, he proposes a different approach called "Business Network Approach," which will also be discussed more in detail in a later section.

Similarly, Kwon et al. implemented a system that predicts stock price movements based on the correlation between highly connected stocks [25]. In this research, the correlation was defined using the following formula,

$$c_{ij} := \frac{\langle r_i r_j \rangle - \langle r_i \rangle \langle r_j \rangle}{\sqrt{(\langle r_i^2 \rangle - \langle s_i \rangle^2)(\langle r_j^2 \rangle - \langle s_j \rangle^2)}}$$ [25]

where $<r_i\ r_j>$ is a "temporal average over the given period and r_i at a give time is defined as $\ln(x_i \text{ at time } t + \Delta t) - \ln(x_i \text{ at time } t)$ and "x_i at time t"is the closing price of a stock x at time t [25]. Their research focused on identifying the top K firms in Korea Stock Exchange (KSE) that have the highest correlation value [25]. For example, Samsung Electronics was shown to be closely related to Samsung Electro-Mechanics, Donguanam semiconductor, Samsung Techwin, Samsung SDI, and Trigem computer, all of which are either a subsidiary or a close partner of Samsung Electronics [25]. In this research, for each of the K companies identified, 75 input variables related to technical indicators were extracted based on inputs used in their previous research [26], [27], in which they tried to predict stock price movements using auto-correlation approaches like neuro-genetic hybrids and genetic ensembles of recurrent neural networks [25]. The variables were subsequently fed into a Feature Selection Genetic Algorithm ("FGSA"), which selected significant variables from the set of 75 * K variables [25]. The team compared the result of this approach with those of their previous research on auto-correlation approaches and came up with a result that of 274 cases, the FGSA based model produced 136 cases of better results than conventional "buy and hold" strategy, while an auto-correlation approach like RNN produced only 104 cases [25].

3-c. Natural Language Processing Based Approach

The last category in this section is the Natural Language Processing ("NLP") based approach. Normally, literature reviews on this category like [15], [20], [28] do not break it down further into subcategories, but based on my observation of research in this area, this section is broken down into two parts - cases when components of textual data like noun phrases or proper nouns are used as features of a prediction model and cases when sentiment scores extracted from textual components are used as features. The last subcategory is about the business network approach that combines the concept of cross-correlation and sentiment scores to create a prediction model that relies on a business network graph.

3-c-i. Textual components as features

The most direct way of using textual data for a prediction model is to use its textual components as the main features. This is in contrast with the indirect way of calculating sentiment scores first and then using the scores to train a model. One of the first attempts to dissect textual data and use them to predict stock price data is that of Wuthrich et al. in 1998 [35]. In their research, they came up with a list of keywords that were "provided once by a domain expert and judged to be influential factors potentially moving stock markets [35]." For each keyword in the list, the number of occurrences was counted per document and then transformed into a weight that ranged from 0 to 1 and then stored as a time series data [35]. The time series was then used to predict stock price movements of the Dow Jones Industrial Average [35]. The result of this experiment was disappointing, as the accuracy only reached up to 45% [35]. However, the authors noted that their system sometimes reached accuracy of 60% for shorter prediction windows of a few weeks [35]. Similarly, Schumaker et al.'s AZFinText system directly took in textual components and used them as features for predicting stock prices [5]. The authors performed a support vector machine classification on different forms of textual representation, like the bag of words, noun phrases, and named entities, which were extracted from textual sources, such as financial news articles, company filings, and analyst reports [5]. These textual representations will be covered in more detail in a later section.

On the other hand, research like those of Mittermayer [36] and Zhai et al. [37] utilized the vector space model to classify financial articles into "Good News," "Bad News," and "No Mover" that respectively correspond to rise, fall, and no change in the stock price. In a vector space model, textual data of a document is directly fed into a training algorithm in the form of a vector containing TF * IDF values for each term in the document [36]. Here, a TF of a term in a document represents the number of term's occurrences in the document [36]. An IDF of a term t can be calculated as $\log(N / DF_t)$, where DF_t is the number of documents in the corpus that contains the term t and N is the number of documents in the corpus [41]. TF * IDF thus assigns a higher value to terms that appear frequently in a document but do not show up in other documents and assigns a lower value to terms that appear rarely in a document but show up in many documents [36]. Both research used a support vector machine model to train the vectors against historical stock price performance, and Zhai et al.'s system demonstrated a fairly high accuracy of 64.7% [36], [37].

3-c-ii. Sentiment score as a feature

An approach using sentiment scores as features differs from the approach of feeding textual components into a training model in that a training in the former approach does not directly utilize textual data. However, textual data still plays a huge role in the prediction process because the sentiment scores are built based on textual components. The key idea behind this approach is that "emotions, in addition to information, play a significant role in human decision-making [13]." According to Bollen and his team, the evidence for this can be

found in numerous literature like [38], [39], [40]. Specially in R.J. Dolan's work [38], it was claimed that "emotions are less encapsulated than other psychological states as evident in their global effects on virtually all aspects of cognition."

In this approach, how sentiment scores are generated significantly influences the result of a prediction. Some research like [13], [19] utilized already-classified sentiment lexicons from publicly available tools. The robustness of these tools have been explored by research like [43], while some research like [15] and [42] contended that financial sentiment lexicons vary significantly from the lexicons from publicly available tools. For example, words like "crude" are usually used to indicate industry jargons like "crude oil" in financial context, but the publicly available lexicons would normally classify them as negative expressions [15]. Loughran and McDonald were able to identify similar cases for 75% of negative words found in a general sentiment lexicon [42]. Also, the opposite can be true where neutral words in general lexicons like "bull" are positive expressions in financial context [15]. Therefore, research like [15] trained their own sentiment lexicons and used them to create their own weighted sentiment scores.

A classical example of using publicly available sentiment lexicons, the research by Bollen et al. extracted public sentiments from a large collection of Twitter tweets and applied the scores as features to predict the stock price movement of Dow Jones Industrial Average ("DJIA") [13]. It is important to note that unlike other research papers that are focused on company-centric predictions this research predicted overall market movements by using sentiments of the general public [13]. The extraction of sentiments was performed using mainly two tools - OpinionFinder and GPOMS [13]. OpinionFinder outputs a positive or negative sentiment based on twitter tweets on a given day and thus can be outputted as a time series of a ratio between positive and negative sentiment [13]. On the other hand, GPOMS provides granular sentiments by outputting six dimensions of sentiment - "calm, alert, sure, vital, kind, and happy" [13]. And these tools will be discussed more in detail later. These sentiment scores were then correlated to historical stock price performance of DJIA to predict changes in DJIA price [13].

Another example is that of Zhang et al [19]. Their research utilized Topsy, which is a twitter textual data analysis platform developed by Topsy Lab with tweets from 2006 [44]. The platform is now discontinued from the internet due to Apple's acquisition of Topsy Lab in 2013 [44]. The platform tags each tweet into three categories - "highly influential," "influential," and "no label". The authors assigned weights of 3, 2, and 1 respectively to tweets [19]. Along with Topsy, the authors also utilize OpinionFinder to assess sentiment polarities and sentiment strengths in tweets [19]. By combining all three metrics, the authors represent sentiment of a company with the following formula [19].

$$En(v_k) = \frac{\sum_{oe \in OE_k^+} w_a(oe) \cdot strenght(oe) - \sum_{oe \in OE_k^-} w_a(oe) \cdot strenght(oe)}{\sum_{oe \in OE_k^+} w_a(oe) \cdot strenght(oe) + \sum_{oe \in OE_k^-} w_a(oe) \cdot strenght(oe)} \quad [19]$$

Here, OE^+ and OE^- respectively represent a set of positive and negative tweets about a given company *k* [19]. $w_a(oe)$ represents the weights assigned by Topsy, and *strength(oe)* represents the strength of a tweet that is extracted from OpinionFinder [19]. Based on the calculated sentiment score, Zhang et al.'s sentiment analysis based system predicts upward stock movements if the score is higher than the positive threshold predetermined by training [19]. It predicts downward movements if the score is lower than the predetermined negative threshold [19].

In contrast to the previous two studies, the research by Li et al. [15] internally generated their own lexicon from financial texts. And based on the lexicon, they built two sentiment scores to represent optimistic and pessimistic sentiment towards a stock [15]. The scores are calculated in the following way.

$$M_s^+ = \sum_{i=0}^{\tau}\sum_{j=0}^{K}\frac{P_{i,j}\times W_j}{l_i}\times T_i \quad M_s^- = \sum_{i=0}^{\tau}\sum_{j=0}^{K}\frac{N_{i,j}\times W_j}{l_i}\times T_i \qquad [15]$$

Here, M_s^+ represents the score measuring the optimistic mood of a stock s, while M_s^- represents the pessimistic mood [15]. P_{ij} and N_{ij} are the number of positive and negative words (determined by their lexicon) in document j on i-th day after the release of the document, and l_i is "the total number of the words in the postings in the i-th day [15]." W_j is a weight of posting j that is determined by the number of clicks the document received [15]. According to the authors, this method allows the model to give more weights to documents that appear to have more authority on the internet [15]. T_i is a time factor that dampens the effect of the sentiment aroused by the document over time [15]. The weight factor in this research is similar to the approach taken by Zhang et al. in that they both attempt to give more weights to documents that have more influence on public sentiment [15], [19].

3-c-iii. Business Network Approach

Business network approach in financial forecasting is a relatively new concept proposed by Zhang et al. in 2015 [19]. The concept tries to capture multi-faceted relationships companies can have with each other by creating a "business network" that maps out competitive or collaborative relationships among companies on a graph [19]. On the graph, each node represents a company, and an edge with weight *w* represents a relationship between two companies with negative weight indicating competitive relationship and positive weight indicating collaborative relationship [19]. Zhang et al. relies on an automatically generated a business network of firms in the U.S. that was made by Lau et al. [19]. They tie this concept of business network with the concept of energy cascading model ("ECM"), and they define ECM in the following way: "Given a business network and a set of source nodes, the total net energy cascading to a sink node is determined by the energy propagated from all the source nodes through all the propagation paths. During energy cascading, the types of energy can be changed according to the nodes and the edges attached to a propagation path [19]." At a high level, an upward movement of stock price is captured by positive energy, and a downward movement by negative energy [19]. The ECM calculates the stock price movement of stock X by using a

predetermined set of source nodes - those that emit or propagate the first wave of energy calculated by their actual stock movement and sentiment scores of those nodes - and the node of stock X that acts as a sink node and receives all energy propagated from nodes along all possible paths from the source nodes to the sink node [19]. Essentially, every node in the ECM emits its own energy that is a combination of its own internal energy, represented by a sentiment score on the corresponding company, and external energy that flowed in from other nodes through edges in the network [19]. However, as an exception, the external energies of source nodes are captured by stock price movements of corresponding stocks [19]. The sentiment score that represents internal energy of nodes is calculated from readily available sentiment analysis tools - in this case, Topsy and OpinionFinder [19]. At the end of the cascade model, the total net energy received by the sink node determines whether the stock price of X will go up or down [19]. If the total net energy is greater than the positive threshold determined by training, then the stock price would be predicted to go up [19]. If the total net energy is less than the negative threshold determined by training, then the stock price would be predicted to go down [19]. The exact algorithmic theory behind the ECM is beyond the scope of this paper. But, what is important to note here is the use of natural language processing (sentiment analysis) to measure the internal energy of nodes.

Zhang et al.'s research showed that the ECM approach outperforms other systems, namely cross-correlation, artificial neural network, and sentiment-based systems [19]. According to their results, the artificial neural network system performed the worst [19]. This suggests that previous stock price movements alone cannot be the predictor of future stock price movement, which refutes the core thesis behind the technical investing [19]. The authors noted that the sentiment-based system did not produce accurate predictions that other literature have reported [19]. The cross-correlation approach demonstrated fairly good outcomes of 61.9% and 62.1% for upward and downward predictions [19]. This is roughly 4% higher than the result of Schumaker's AZFinText System [19]. Most importantly, Zhang et al. reported that the ECM system showed the highest accuracy of 67.7% and 67.4% for upward and downward predictions [19].

4. Natural Language Processing based Financial Forecasting

In this section, some of the key steps behind NLP based financing forecasting models will be explored. It will first cover some pre-processing techniques that many researchers perform in their research. Then, we will discuss common features that are used in NLP based financial forecasting. Also, some of the most used algorithms for training and predicting stock price movements will be explored. Lastly, we will discuss how performance is evaluated in these research papers.

4-a. Pre-Processing Set-up

4-a-i. Technical Analysis and Prediction Window

According to Schumaker et al., "almost all techniques start off with a technical analysis of historical security data by selecting a recent period of time and performing linear regression analysis to determine the price trend of the security [14]." Normally, a regressed prediction based on historical price is used as a benchmark for evaluating performance of a model. Schumaker et al. admitted that this violates the random walk theory, which states that previous stock price data cannot be a predictor of future price movement [14]. However, he claimed that short periods like 20 minutes are short enough for "weak predictive ability" to remain, citing Gidofalvi's work [16]. For example, in their research, for each article, a prediction for the next 20 minutes was made based on stock price data of 60 minutes prior to the release of the article [14]. This regressed prediction was then used as a benchmark to compare three NLP based models he created [14]. Bollen et al. also used a regressed prediction as the baseline for comparison [13]. In this research, historical three day prices were used to regress on the future index price of DJIA to predict daily closing stock price [13]. This goes against Gidofalvi's notion of "weak predictive ability" window that prediction of stock price is only allowed for very short time frames. Bollen et al. did not acknowledge these points in their research, but many researchers like the authors of [35], [36], [37], [45], [46] ignored this notion and made predictions on a daily basis. Although some researchers like Chatrath [47] who restricted predictions to less than 20 minutes, Antweiler and Frank [48] showed that some textual data like "message posting does help to predict volatility both at daily frequencies and also within the trading day."

4-a-ii. Feature Space Limitation

Having too many features in a classification algorithm can be problematic, as the algorithm can get too slow and inefficient [28]. Pestov in his research called this "the curse of dimensionality [49]." Therefore, almost all research experiments that involve learning or classification algorithms preprocess raw data to reduce the number of features [28]. The most basic step for dimensionality reduction is to get rid of stop-words that bring no informational gain to the model, and this step is observed in almost all models mentioned in this paper. For example, Schumaker and Chen eliminated stop-words like "a" and "the" that carry semantically no meaning to make the bag of words approach more effective. For some cases, researchers like

Wuthrich et al. specified a "predefined dictionary" of terms they are looking for [35]. In their research, they consulted experts in the financial domain to come up with a predefined list of keywords, such as "bond strong," "dollar falter," and "dow rebound [35]." Similarly, Peramunetilleke and Wong also utilized a fixed set of keywords provided by industry experts [50]. In other cases, researchers used features that appeared more than a certain number of times [14]. In Schumaker and Chen's research, only "terms that appeared three or more times in an article" were used as features for training a classifier [14]. Finally, some experiments like that of Zhai et al. [37] and Mittermayer et al. [36] only selected top "k" terms with the highest weights or associated values. There are many other ways besides the ones mentioned here, but the ones above are the most common techniques [28].

4-a-iii. Textual Sources

Schumaker and Chen divided textual sources into two types - internally generated by companies and externally generated [14]. Internally generated sources are most easily found in SEC filings, and they offer useful information about the company's future performance through forward looking statements typically found in the discussion section of 10-K or 8-K filings [14], [51]. Some examples of research that relied on internal documents include research by Butler and Keselj [52] and Li et al. [53]. In Butler and Keselj's research, they retrieved annual reports from the investor relations page of each company website [52]. Similarly, Li et al. retrieved annual reports and quarterly reports from the EDGAR website and specifically utilized the "Management's Discussion and Analysis" section as textual data [53].

On the other hand, externally generated sources provide "more balanced view of the company" and can be further broken down into multiple types - analyst created, news outlets, news wire services, and discussion boards [14]. Here, it is important to note that news outlets and news wire services are different in that news outlets like Financial Times publish information "at specific time intervals" while news wire services like Yahoo Finance publish information "as soon as it is publicly released or discovered [14]." Because of their "timely release" and ease of gathering information, news wire services are widely used for algorithmic financial predictions, and Schumaker and Chen utilized Yahoo Finance to collect the articles required for the research [14]. In doing so, they avoided the "company in passing" problem by collecting articles based on stock tickers on Yahoo Finance [14]. Unlike Schumaker's usage of news wire only, many experiments like those of [35], [46], [48], [54] used a mix of news outlets and news wire services. For example, Rachlin et al. used today.reuters.com as a news wire service and Forbes as a news outlet [48]; Tetlock et al. pulled textual data from multiple sources, such as Wall Street Journal, Dow Jones News, and Factiva news service [54]. Finally, some research like Das and Chen's [45] relied only on textual data from discussion boards.

Another increasingly popular source of textual data is Twitter and other social networks. For example, Bollen collected tweets from February 29th 2008 to December 19th 2008, which sums up to 9,853,498 tweets posted by 2.7 million Twitter users [13]. They filtered for posts that contained expressions that normally portray moods, like "i feel" and "makes me [13]." Also,

they took out spam-like posts by filtering out posts with URL links to other websites [13]. Similarly, Vu et al. [55] and Zhang et al. [19] also utilized Twitter textual data to make stock price predictions.

4-b. Feature Extractions

4-b-i. Bag of Words

According to Schumaker et al., "the bag of words approach has been used as the de facto standard of financial article research primarily because of its simple nature and its ability to produce a suitable representation of text [14]." This is supported by the fact that so many researchers listed in this paper like [14], [36], [45], [46], [48], [50] used the approach in their research. The bag of words approach is basically "breaking the text up into its words and considering each of the words as a feature [28]." For example, a text like "I love cars because cars are awesome" would be transformed into a map between words and frequencies: {i : 1, love : 1, cars : 2, because : 1, are : 1, awesome : 1}. Like Schumaker et al. said, the approach offers a rich set of options to work with the data [14]. Some research utilized the representation to form TF-IDF matrices, which are built based on the frequency of a term in a document and the scarcity of documents containing the term [36], [37], [50]. On the other hand, other research like [14], [48], [53] transformed the representation into boolean values of whether a term exists in a document or not. However, the approach does come with some limitations. First, it ignores the order and meaning of words in a sentence [56]. For example, the bag of words approach would not be able to distinguish between "Ruth is taller than Babe" and "Babe is taller than Ruth," even though the two sentences are completely opposite in meaning. Another shortcoming is that sparse representations could increase the space complexity of a model drastically, making the model inefficient [56]. Because of this shortcoming, researchers often limit features in ways shown in the previous section.

4-b-ii. Other Textual Representations

Another way to represent textual data in Natural Language Processing is to use noun phrases. Schumaker et al. utilized a noun phrasing approach to represent textual information [14]. Noun phrasing is similar to the bag of words approach except that the parts of speech are tagged to each word and that nouns and surrounding words that form phrases, such as "the big black cat", are extracted [14], [57]. Therefore, unlike the bag of words approach, this approach captures meanings more holistically [57]. However, because it involves tagging parts of speech for each word in textual data and then identifying which adjacent words are part of the noun phrase, it is generally more complex than the bag of words approach [57]. For that reason, many researchers prefer to use the bag of words approach [14]. Schumaker and his team also introduced an approach called "named entity [14]." This approach is a one-step extension of the noun phrasing approach, where nouns are classified into categories that have semantic meanings [14]. For example, MUC-7 categorizes nouns into "date, location, money, organization, percentage, person and time [14]." In general, the approach tags nouns by matching the nouns to extraction patterns stored in "large lexicons of sample entities and/or

word patterns [14]." Whenever there is a match, the tagger will assign a corresponding entity tag to the noun [14]. In their research, they used McDonald et al.'s AzTeK system to tag nouns into 7 categories: date, money, organization, location, percentage, time and person [14]. Similarly, Vu et al. built their own Named Entity Recognition System to filter out unnecessary data from Twitter data [55]. In this case, they tagged nouns into 4 categories: persons, organizations, hardware, and software [55]. Subsequently, they removed tweets that do not "contain any named entities as listed on the company keyword list [55]."

Schumaker and Chen made a direct comparison among all three techniques above [14]. As he had expected, the bag of words showed the worst result with 0.04422 in closeness, 57% in directional accuracy, and 1.59% in simulated training results [14]. The poor performance can be attributed to the fact that the bag of words technique contains "too many noisy article terms" that can inappropriately influence classifications in a support vector machine model [14]. The authors originally expected the named entities to show the best performance due to anticipation that "a more abstract textual representation would perform better [14]." But, to the contrary, the noun phrases technique demonstrated the best result with 0.04887 in closeness, 58% in directional accuracy, and 2.57% in simulated training engine results, while the named entities technique showed 0.03407 in closeness, 55% in directional accuracy, and 2.02% in simulated trading engine results [14]. The authors introduced a fourth technique to verify their anticipation that abstraction helps [14]. Instead of using the named entity technique, they extracted proper nouns, which can be seen as "a superset of named entities but without the entity categories [14]." This technique performed with better closeness, directional accuracy and simulated trading engine results (0.04433, 58.2%, 2.84%) than the results of the noun phrases technique [14]. The improvement shows that abstraction in textual representation is beneficial to algorithmic performance [14]. The authors directly attributed the success of proper nouns to the fact that it is "freer of the noise plaguing noun phrases and free of the constraining categories used by named entities [14]."

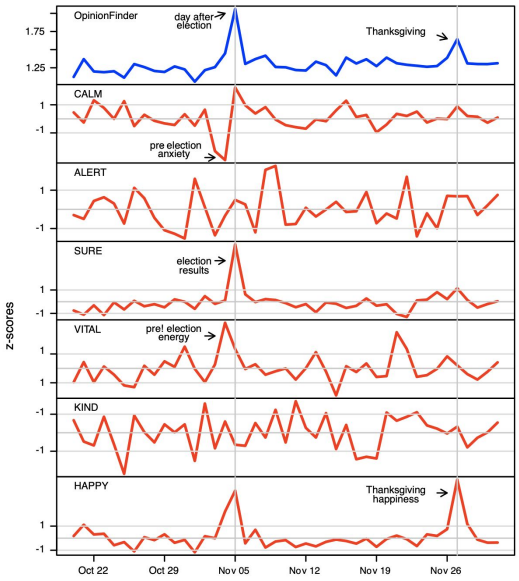
4-b-iii. Sentiment Score

Instead of extracting words and their frequencies directly from textual data, some researchers preferred to create sentiment scores from textual data first and then train their prediction models with the scores. One of the most commonly used tools for generating sentiment scores is OpinionFinder, which allows users to assess the sentiment polarity (positive or negative) at a sentence level [13]. For each day, the ratio of positive and negative tweets can be calculated, and a time series of this metric has demonstrated high correlation with Gallup's Consumer Confidence Index [13]. The OpinionFinder tags a sentiment polarity to each word by using its own lexicon, and the lexicon has been worked on by many research experiments like [58], [59], and [60]. In Bollen et al.'s research, the algorithm counted the number of positive or negative words based on OpinionFinder's lexicon on a given day to construct a time series of ratio between occurrences of positive and negative words [13]. However, they pointed out that OpinionFinder "adheres to a unidimensional model of mood" and lacks richness of human mood [13]. Therefore, they supplemented this tool with another tool called Google Profile of

Mood States (GPOMS) [13]. The tool allows users to classify the sentiments of textual data into 6 categories of calm, alert, sure, vital, kind, and happy [13]. For each tweet, GPOMS matches words to its lexicon [13]. And for each dimension of mood, the score is given based on "the weighted sum of the co-occurence weights of each tweet term that matched the GPOMS lexicon [13]." Both scores from OpinionFinder and GPOMS are normalized into z-scores according to the formula below [13]. You can see that the mean $\bar{x}$ and standard deviation σ are calculated on a sliding window basis of k days [13].

$$\mathbb{Z}_{X_t} = \frac{X_t - \bar{x}(X_{t\pm k})}{\sigma(X_{t\pm k})}$$ [13]

The effectiveness of these sentiment scores was tested in their research[13]. They tracked the sentiment polarity ratio from OpinionFinder and the six sentiment scores from GPOMS on a period between October 5th 2008 to December 5th 2008, which includes the U.S. presidential election and Thanksgiving [13]. According to the time series of each sentiment score shown on the right, all 7 metrics demonstrated visually significant changes around the two events [13]. For example, the polarity ratio rose drastically to the more positive side when the two events occurred and quickly reverted to the baseline afterwards [13]. They noted that scores from GPOMS were able to capture more nuanced sentiment trends [13]. For example, the Calm metric demonstrated a huge drop a day before the election, indicating "pre-election anxiety [13]." Simultaneously, the Vital score rose drastically too, which showed "pre-election energy [13]." On the other hand, the Happiness score showed a huge increase on the day of Thanksgiving as anticipated [13]. They also tried to identify any correlation between the score of OpinionFinder and the six scores of GPOMS by running a multivariate regression on them [13]. According to the result of the regression, the Sure, Vital, and Happy score showed a significant p-value of 4.25e-08, 0.004, and 1.30e-5 [13]. This indicated that the positivity score of OpinionFinder overlaps in certain extent to these particular scores [13]. However, other scores showed insignificant correlation, which indicates that GPOMS does indeed provide additional perspective on public mood [13].



[13]

Some researchers like Li et al. [15] came up with their own lexicons based on two assumptions - 1) "the semantic orientation of a word tends to correspond to the semantic orientation of its neighbors in the textual content," 2) "A firm-specific article with a positive

(negative) tone is typically in accordance with the upward (downward) price trend of a relevant stock [15]." Accordingly, they defined the probability of a word w being positive as followed.

$$\begin{aligned} P^{+}(w) &= P(w|E=+,T=\uparrow) \\ &\approx P(w|T=\uparrow)P(E=+|w,T=\uparrow) \\ &= P(w|T=\uparrow)\sum_{i=0}^{M}P(e_i=+|w,T=\uparrow) \end{aligned} \quad [15]$$

Here, T is the stock price trend, where an upward arrow indicates rising stock price and a downward arrow indicates declining stock price [15]. E represents the sentiment polarity of surrounding words, where + indicates positive sentiment and - indicates negative sentiment [15]. e_i represents "the sentiment word in the paradigm set S," while M is "the total number of positive words in the set [15]." The first clause of the equation above ("P(w | T=up)") can be estimated by dividing "the number of the documents tagged with upward stock trend containing word w in the training corpus" by "the total number of the documents containing word w in the training corpus [15]." The second clause ("$P(e_i = + \mid w, T = up)$") can be estimated by the following equation according to the authors [15].

$$P(e_i=+|w,T=\uparrow) \approx \lambda I^{+}(w,e_i) + (1-\lambda) \times S(w,e_i)$$

Here, λ is "the coefficient that adjusts the contribution of the semantic similarity [15]." $I^{+}(w,e_i)$ represents "the statistical association between word w and positive word e_i in articles with an upward trend which is measured by χ^2 [15]." $S(w, e_i)$ indicates "the semantic similarity in terms of semantic relationships in the WordNet [15]." By multiplying the two clauses, the authors were able to calculate the probability of a word being positive (or negative) [15]. As discussed in one of the previous sections, sentiment scores were calculated based on this sentiment analysis [15].

4-c. Processing Algorithms

4-c-i. Naive Bayesian

Old but very popular for natural language processing, the Naive Bayesian algorithm is a probabilistic classification algorithm [61]. This means that the classifier returns the class with the maximum "posterior probability" for a given document [61]. It is based on Bayes's Rule that dissects a conditional probability into multiplications of other probabilities [61]. Accordingly, the classification can be represented with the following equation, where c-hat is the outcome class and C represents the set of all classes and d is the input document [61].

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} \quad [61]$$

We can drop the P(d) on the bottom because the probability of document d stays the same inside the argmax [61]. P(d | c) can then be converted to $P(f_1, f_2, f_3, ..., f_n \mid c)$, where f's are features from a bag-of-words representation of the document d [61]. And in order to simplify the equation above for algorithmic calculations, the Naive Bayesian algorithm makes two important

assumptions [61]. First, it assumes that the position of a word does not influence the classification [61]. Second and more importantly, it assumes that each $P(f_i \mid c)$ is independent of each other and thus $P(f_1, f_2, f_3, ..., f_n \mid c) = P(f_1 \mid c) * P(f_2 \mid c) * \ldots * P(f_n \mid c)$ [61]. The second assumption is the reason why this approach is called "naive" because given the grammatical and semantic structure of a sentence, such assumption would normally not make sense [61]. The final equation can be calculated as follows [61].

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} \quad [61]$$

Most notably, Antweiler and Frank used a naive bayesian algorithm in their research [48]. They admitted that assumptions behind naive bayesian algorithms are "highly unrealistic," but they noted that it "performs rather well in practice [48]." Specifically for their implementation of naive bayes, they used a software package called the Rainbow package, which is publicly available for academic purposes on the internet [48]. Similarly, another use case of Naive Bayes is the research by Li et al. [53], for which the author used the Naive Bayes module in Perl.

4-c-ii. Support Vector Machine and Regression

Support Vector Machine ("SVM") is one of the most popular options for NLP based financial forecasting. It is "a non-probabilistic binary linear classifier used for supervised learning [28]." Because it is a binary classifier, it is often found in research like [36], [37], [62] that have categorical results, such as upward or downward stock price movement. Basically, the algorithm "finds a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points [63]." Often, there may be multiple hyperplanes that fit the classification rule, so the algorithm's objective "is to find a plane that has the maximum margin" between two classes [63]. The reason why this algorithm is called SVM is that it relies on support vectors to identify the best hyperplane [63]. A support vector is an N-dimensional vector input that defines the location and form of the hyperplane [63]. Like other machine learning algorithms, the algorithm has a loss function, and a gradient descent is used to optimize the function, which in turn also "maximizes the margin of the classifier [63]." Sometimes, a support vector machine may need to be supplemented by a kernel function [64]. This function allows a machine learning algorithm to solve nonlinear problems using a linear classifier like SVM [64]. It is normally defined by $K(x,y) = \langle f(f_1), f(f_2)...,f(f_n) \rangle$, where f_i represent an i-th dimension feature and K represents the kernel function that transforms features into different forms using a function f [64]. There are multiple kernel functions available for machine learning algorithms [65]. One of the most popular kernels in this domain is the radial basis function, and research like [19], [37] used it to train SVMs.

A slightly modified variation of SVM is the Support Vector Regression ("SVR") model [28]. According to Awad and Khanna [66], "the regression problem is a generalization of the classification problem" in that regressions' outputs are continuous values while those of classifications can only be chosen from a predetermined set of classes. Likewise, SVR is useful

for cases when a model is trying to output a numerical prediction [66]. In this domain, SVR is mostly used to predict returns [28]. This generalization process is "accomplished by introducing an ε-sensitive region around the function, called the ε-tube [66]." The exact algorithmic steps won't be discussed in this paper, but [66] provides a good overview behind the process. A typical example of SVR is the model by Schumaker and Chen [14]. In this case, they used a specialized SVR called "Sequential Minimal Optimization" that is capable of "discrete number analysis [14]." Another example is that of Hagenau et al. [57]. Also in their research, a SVR model was used to predict returns based on financial news [57].

4-c-iii. Decision Rules

Some research like [46], [50], [55] utilized a rule-based algorithm called the Decision Rules to classify stocks into upward, downward, and no price movement. A decision rule is basically a set of conditional statements that are made using features as conditions of the statements [67]. It follows a simple If-Then structure that is intuitive to understand and interpret [67]. Normally, conditions that have some predictive power are extracted from textual data based on training dataset, and the trained decision rules are applied to testing dataset to see if it has any predictive power [67]. For example, Rachlin et al. trained a Decision Tree Induction algorithm called C4.5 that "does not assume attribute independence [28], [46]. Its classifying rules were based on sentiment scores of the company of interest and the general public mood [46]. As another example, Peramunetilleke and Wong generated decision rules of three conditions (UP, DOWN, STEADY) based on occurrences of predetermined keywords and closing values [50]. Finally, Vu et al. also relied on C4.5 because the algorithm "eliminates the need for word independence assumption [55]."

4-c-iv. Other Algorithms

There are other algorithms that researchers have used to forecast stock performance. Most notably, Bollen et al. tried two regression models to identify any linear and non-linear relationship among the features and future stock price [13]. For identifying linear relationships, they utilized a linear regression technique called Granger Causality Analysis, which correlates a variable, like in this case DJIA index price, with lagged values of features [13]. DJIA index price was regressed with lagged values of all 7 sentiment scores and DJIA historical prices [13]. And as a result of this experiment, it was found that the calm score that was lagged by 3 days exhibited the most significant correlation to the DJIA index price [13]. On the other hand, the authors used a Self-organizing Fuzzy Neural Network (SOFNN) model to identify any non-linear relationships [13]. Based on the finding above that the calm score has the best correlation to the index price, the authors utilized the model to find out whether a pair of the calm score and one of the other 6 scores would improve the prediction accuracy [13]. From the analysis, they found out that a mix of calm and OpinionFinder sentiment had "no effect on prediction accuracy [13]." Some research like [45] combined multiple classifiers into a stack. In Das and Chen's research [45], they combined "Naive Classifier, Vector Distance Classifier, Discriminant-Based Classifier, Adjective-Adverb Phrase Classifier, Bayesian Classifier [28]".

However, the accuracy level of the stack didn't show much improvement over the traditional Bayes Classifier [45].

4-d. Performance Evaluation

Researchers in this domain have used multiple metrics to evaluate their performance, but evaluations can be broken down into largely three main metrics, namely closeness, directional accuracy, and return from a simulated environment. In Schumaker and Chen's experiments, the closeness metric measured the mean squared error ("MSE") between the actual stock price and the predicted value and thus indicated the proximity of the prediction to the actual result [14]. This metric mainly applies to research that predicts discrete values of stock return. For example, Hagenau et al.'s discrete stock return predictions were evaluated using R^2 (similar to MSE) to see how close the predictions were to the actual values [57]. Next, the directional accuracy, as the name implies, is whether the predicted direction matches the actual directional movement of the stock price [14]. This is the most frequently used metric for binary or ternary classifications, such as [14], [46], [52], [53], [55], [57]. Normally, an accuracy greater than 50% is regarded as a good result because it is better than random chance [14]. The simulated trading engine is a simulated setting where a trading algorithm trades a fund of a certain amount on each trade based on predetermined rules [14]. In Schumaker and Chen's research, the rules were modified from Mittermayer's rule [36], which optimizes near term profit [14]. The engine took an article and evaluated the stock price after 20 minutes of the article's release and invested if the expected stock price was at least 1% greater than the stock price when the article was released [14]. The research assumed zero transaction cost [14]. Zhai et al. also used a market simulation to assess the profitability of their research [37]. In this case, the simulator assumed a transaction cost of $20 [37].

5. Limitations of NLP based Financial Forecasting

As Loughran and McDonald [42] showed through their research, around 75% of terms defined as negative in "a general emotion word dictionary" are actually not negative in financial textual data. And the opposite can sometimes be true; "an emotionless word can be a sentiment in the realm of finance to some degree [15]." Despite these findings, most sentiment scores calculated in papers mentioned here relied on non-financial sentiment lexicons [13], [19], [54]. In some cases, researchers like [35], [45] utilized expert-generated or hand-picked keywords that can be very subjective. This shows the lack of domain-specific resources in this area, which can be attributed to mainly two reasons. First, research in this domain is relatively young, as one of the earliest reviews in this domain was published in 2014 according to Xing et al. [20]. Given the short history, the number of research in this domain is visibly smaller than those in other domains. Next, most of the groundbreaking research led by quantitative investment firms are not disclosed to the public [68]. For example, Renaissance Technologies has generated an annual return of 66% from 1988 to 2018, a return unmatched by fundamental investment experts like Warren Buffet and Peter Lynch [69]. How the fund made a tremendous return remains to be unknown, as Bloomberg reported that "the Medallion Fund, an employee-only offering for the quants at Renaissance Technologies, is the blackest box in all of finance [68]."

Another problem stemming from its short history is lack of standardized research procedures. Hagenau et al. noted that "most related research in this area suffers from the fact that each researcher uses his proprietary method and evaluates those methods on the ground of a proprietary data set [57]." He then claimed that this lack of standardization makes research in this domain "vaguely comparable [57]." Xing et al. also claimed that "a main current difficulty to summarize and compare the existing studies is various and incomplete measurements used by researchers [20]." This problem is easily noticeable from the fact that almost no research utilized all three metrics discussed in Section 4-d, which made comparisons across multiple research very difficult.

6. Conclusion

The enormous return the Medallion Fund has been making for almost 30 years seems to indicate that the market is not as perfectly efficient as Eugene Fama claimed [1], [2]. It is not the only evidence, as the research on NLP-based financial forecasting showed significant performance in predicting the stock market (at least better than 50%). It is true that most of them were tested only for a short period of time and the actual trading viability remains to be unknown. However, the notion that market psychology that is reflected in textual data can be used to predict the market is intuitive qualitatively and is proven to some degree quantitatively through research on this domain. Improvement in the ability to forecast stock price returns is needed not only because of the money it brings but also because of the sophistication it brings to the market. Major boom-and-busts in financial instruments can be attributed to blinded greed that has led investors to ignore signals showing unhealthy conditions of underlying assets. Information extracted through NLP techniques discussed in this paper can be useful in preventing ignorant investments that could potentially cause a crisis like the one in 2008. Having said that, there is still a long way to go since this domain is still in its early stage.

References

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

[2] Fama, E. F. (1991). Efficient capital markets: II. The journal of finance, 46(5), 1575-1617.

[3] Samuelson, P. A. (1973). Proof that properly discounted present values of assets vibrate randomly. The Bell Journal of Economics and Management Science, 369-374.

[4] Malkiel, B. G. (1999). A random walk down Wall Street: including a life-cycle guide to personal investing. WW Norton & Company.

[5] Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 1-19.

[6] Lo, A. W., & MacKinlay, A. C. (2011). A non-random walk down Wall Street. Princeton University Press.

[7] Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. Applied Intelligence, 26(1), 25-33.

[8] Gallagher, L. A., & Taylor, M. P. (2002). The stock return–inflation puzzle revisited. Economics Letters, 75(2), 147-156.

[9] Butler, K. C., & Malaikah, S. J. (1992). Efficiency and inefficiency in thinly traded stock markets: Kuwait and Saudi Arabia. Journal of Banking & Finance, 16(1), 197-210.

[10] Gruhl, D., Guha, R., Kumar, R., Novak, J., & Tomkins, A. (2005, August). The predictive power of online chatter. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (pp. 78-87).

[11] Mishne, G., & Glance, N. S. (2006, March). Predicting movie sales from blogger sentiment. In AAAI spring symposium: computational approaches to analyzing weblogs (pp. 155-158).

[12] Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. Economic record, 88, 2-9.

[13] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of computational science, 2(1), 1-8.

[14] Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 1-19.

[15] Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. Information Sciences, 278, 826-840.

[16] Gidofalvi, G., & Elkan, C. (2001). Using news articles to predict stock price movements. Department of Computer Science and Engineering, University of California, San Diego.

[17] Segal, T. (2020, March 31). Fundamental Analysis. Retrieved from https://www.investopedia.com/terms/f/fundamentalanalysis.asp

[18] Hayes, A. (2020, March 16). Technical Analysis Definition. Retrieved from https://www.investopedia.com/terms/t/technicalanalysis.asp#citation-4

[19] Zhang, W., Li, C., Ye, Y., Li, W., & Ngai, E. W. (2015). Dynamic business network analysis for correlated stock price movement prediction. IEEE Intelligent Systems, 30(2), 26-33.

[20] Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: a survey. Artificial Intelligence Review, 50(1), 49-73.

[21] Anton, M., & Polk, C. (2014). Connected stocks. The Journal of Finance, 69(3), 1099-1127.

[22] (n.d.). Retrieved from https://www.spss-tutorials.com/pearson-correlation-coefficient/

[23] Laloux, L., Cizeau, P., Bouchaud, J. P., & Potters, M. (1999). Noise dressing of financial correlation matrices. Physical review letters, 83(7), 1467.

[24] Plerou, V., Gopikrishnan, P., Rosenow, B., Amaral, L. A. N., & Stanley, H. E. (1999). Universal and nonuniversal properties of cross correlations in financial time series. Physical review letters, 83(7), 1471.

[25] Kwon, Y. K., Choi, S. S., & Moon, B. R. (2005, June). Stock prediction based on financial correlation. In Proceedings of the 7th annual conference on Genetic and evolutionary computation (pp. 2061-2066).

[26] Kwon, Y. K., & Moon, B. R. (2003, July). Daily stock prediction using neuro-genetic hybrids. In Genetic and Evolutionary Computation Conference (pp. 2203-2214). Springer, Berlin, Heidelberg.

[27] Kwon, Y. K., & Moon, B. R. (2004, June). Evolutionary ensemble for stock prediction. In Genetic and Evolutionary Computation Conference (pp. 1102-1113). Springer, Berlin, Heidelberg.

[28] Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. C. L. (2014). Text mining for market prediction: A systematic review. Expert Systems with Applications, 41(16), 7653-7670.

[29] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (pp. 106-112). IEEE.

[30] Liu, C., Hoi, S. C., Zhao, P., & Sun, J. (2016, February). Online arima algorithms for time series prediction. In Thirtieth AAAI conference on artificial intelligence.

[31] Hamzaçebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. Expert Systems with Applications, 36(2), 3839-3844.

[32] Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. Expert Systems with Applications, 40(14), 5501-5506.

[33] Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial. Neurocomputing, 10, 215-236.

[34] Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN and GA for stock market forecasting. Expert systems with Applications, 33(1), 171-180.

[35] Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., & Zhang, J. (1998, October). 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, pp. 2720-2725). IEEE.

[36] Mittermayer, M. A. (2004, January). Forecasting intraday stock price trends with text mining techniques. In 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the (pp. 10-pp). IEEE.

[37] Zhai, Y., Hsu, A., & Halgamuge, S. K. (2007, June). Combining news and technical indicators in daily stock price trends prediction. In International symposium on neural networks (pp. 1087-1096). Springer, Berlin, Heidelberg.

[38] Dolan, R. J. (2002). Emotion, cognition, and behavior. science, 298(5596), 1191-1194.

[39] Damasio, A. R. (1994). Descartes' error: Emotion, rationality and the human brain.

[40] Kahneman, D., & Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In Handbook of the fundamentals of financial decision making: Part I (pp. 99-127).

[41] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation.

[42] Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. The Journal of Finance, 66(1), 35-65.

[43] O'Connor, B., Balasubramanyan, R., Routledge, B. R., & Smith, N. A. (2010, May). From tweets to polls: Linking text sentiment to public opinion time series. In Fourth international AAAI conference on weblogs and social media.

[44] Hayes-Roth, A. (2015, December 24). A former Topsy employee has an interesting theory on why Apple shut down this $200 million acquisition. Retrieved from https://www.businessinsider.com/apple-shuts-down-topsy-the-200-million-mystery-laid-to-rest-2015-12

[45] Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. Management science, 53(9), 1375-1388.

[46] Rachlin, G., Last, M., Alberg, D., & Kandel, A. (2007, March). ADMIRAL: A data mining based financial trading system. In 2007 ieee symposium on computational intelligence and data mining (pp. 720-725). IEEE.

[47] Chatrath, A., Miao, H., Ramchander, S., & Villupuram, S. (2014). Currency jumps, cojumps and the role of macro news. Journal of International Money and Finance, 40, 42-62.

[48] Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. The Journal of finance, 59(3), 1259-1294.

[49] Pestov, V. (2013). Is the k-NN classifier in high dimensions affected by the curse of dimensionality?. Computers & Mathematics with Applications, 65(10), 1427-1437.

[50] Peramunetilleke, D., & Wong, R. K. (2002, January). Currency exchange rate forecasting from news headlines. In Australian Computer Science Communications (Vol. 24, No. 2, pp. 131-139). Australian Computer Society, Inc..

[51] Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., & Visa, A. (2004). Combining data and text mining techniques for analysing financial reports. Intelligent Systems in Accounting, Finance & Management: International Journal, 12(1), 29-41.

[52] Butler, M., & Kešelj, V. (2009, May). Financial forecasting using character n-gram analysis and readability scores of annual reports. In Canadian Conference on Artificial Intelligence (pp. 39-51). Springer, Berlin, Heidelberg.

[53] Li, F. (2010). The information content of forward-looking statements in corporate filings—A naïve Bayesian machine learning approach. Journal of Accounting Research, 48(5), 1049-1102.

[54] Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. The Journal of Finance, 63(3), 1437-1467.

[55] Vu, T. T., Chang, S., Ha, Q. T., & Collier, N. (2012, December). An experiment in integrating sentiment features for tech stock prediction in twitter. In Proceedings of the workshop on information extraction and entity analytics on social media data (pp. 23-38).

[56] Brownlee, J. (2019, August 7). A Gentle Introduction to the Bag-of-Words Model. Retrieved from https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[57] Hagenau, M., Liebmann, M., & Neumann, D. (2013). Automated news reading: Stock price prediction based on financial news using context-capturing features. Decision Support Systems, 55(3), 685-697.

[58] Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In Proceedings of the 2003 conference on Empirical methods in natural language processing (pp. 105-112).

[59] Riloff, E., Wiebe, J., & Wilson, T. (2003, May). Learning subjective nouns using extraction pattern bootstrapping. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 25-32). Association for Computational Linguistics.

[60] Wilson, T., Wiebe, J., & Hoffmann, P. (2005, October). Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of human language technology conference and conference on empirical methods in natural language processing (pp. 347-354).

[61] Jurafsky, D., & Martin, J. H. (2014). Speech and language processing. Harlow: Pearson.

[62] Fung, G. P. C., Yu, J. X., & Lam, W. (2003, March). Stock prediction: Integrating text mining approach using real-time news. In 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings. (pp. 395-402). IEEE.

[63] Gandhi, R. (2018, July 5). Support Vector Machine - Introduction to Machine Learning Algorithms. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[64] Afonja, T. (2018, July 13). Kernel Functions. Retrieved from https://towardsdatascience.com/kernel-function-6f1d2be6091

[65] Dataflair Team. (2018, November 16). Kernel Functions-Introduction to SVM Kernel & Examples. Retrieved from https://data-flair.training/blogs/svm-kernel-functions/

[66] Awad, M., & Khanna, R. (2015). Support vector regression. In Efficient Learning Machines (pp. 67-80). Apress, Berkeley, CA.

[67] Molnar, C. (2020, April 27). Interpretable Machine Learning. Retrieved from https://christophm.github.io/interpretable-ml-book/rules.html

[68] (n.d.). Retrieved from https://www.bloomberg.com/news/articles/2016-11-21/how-renaissance-s-medallion-fund-became-finance-s-blackest-box

[69] Zuckerman, G. (2019, November 6). The history of blunders and missteps that led to the quant trading revolution. Retrieved from https://qz.com/1741907/renaissance-technologies-jim-simons-and-the-birth-of-quant-trading/