A Hybrid Approach for Stock Market Prediction Using Financial News and Stocktwits

Alaa Alhamzeh[1,2], Saptarshi Mukhopadhaya[2], Salim.Hafid[1,2], Alexandre.Bremard[1], Előd Egyed-Zsigmond[1], Harald Kosch[2], and Lionel Brunie[1]

[1] INSA de Lyon / LIRIS, 20 Avenue Albert Einstein, 69100 Villeurbanne, France
[2] Universität Passau, Innstraße 41, 94032 Passau, Germany
{alaa.alhamzeh,salim.hafid,alexandre.bremard,elod.egyed-zsigmond,lionel.brunie}@insa-lyon.fr,
{saptarshi.mukhopadhaya, Harald.Kosch}@uni-passau.de

Abstract. Stock market prediction is a difficult problem that has always attracted researchers from different domains. Recently, different studies using text mining and machine learning methods were proposed. However, the efficiency of these methods is still highly dependant on the retrieval of relevant information. In this paper, we investigate novel data sources (Stocktwits in combination with financial news) and we tackle the problem as a binary classification task (i.e., stock prices moving up or down). Furthermore, we use for that end a hybrid approach which consists of sentiment and event-based features. We find that the use of Stocktwits data systematically outperforms the sole use of price data to predict the close prices of 8 companies from the NASDAQ100. We conclude on what the limits of these novel data sources are and how they could be further investigated.

Keywords: Stock market · Sentiment analysis · Online news · Stocktwits · Classification.

1 Introduction

Stock market prediction has been always a challenging task as it depends on various factors and is positioned at the intersection of linguistics, machine learning and behavioral economics [1]. The prediction task can be addressed as a binary classification problem, i.e. whether a particular stock price will rise up or fall down, or as a regression problem where the goal is to predict the future stock price. Generally, two main approaches are considered [2]:

- Technical Stock Analysis: based on the historical numerical values of the stock such as the opening price, the closing price, the traded volume, etc.
- Qualitative Stock Analysis: based on external financial factors like the textual information contained in social media, financial news articles and company profiles.

In our work, we ran experiments using both types of analyses. The stocks on which these experiments were run correspond to 8 different companies from the NASDAQ100 stock exchange.

Our contribution essentially consists in a novel combination of several data sources for stock market prediction, namely Stocktwits in combination with online News, and in running different experiments to compare the quality of these data sources and the predictiveness of the textual features.

Furthermore, our work is mainly a sentiment-analysis-based approach performed on the textual data, though we do also run hybrid-based experiments that involve an event-based approach. The main challenges of sentiment-based approaches to stock market prediction are the finance-specific language and the lack of labeled data. General purpose sentiment-models are not effective enough. In this paper, we aim to run experiments that seek the answer of the following questions:

- Can the use of textual data systematically improve the performance of models based on numerical data?
- Is there an optimal observation period that a model should consider before giving a price movement prediction?
- How can one combine the information retrieved from different data sources?

The paper is organized as follows: in Section 2, we take a close look at the conceptual background of our work as well as the state-of-the-art studies considering stock market prediction. In Section 3, we come to our contribution details. We validate the results in Section 4. Finally, we discuss the overall research questions and future work in Section 5.

2 Related Work

Stock market prediction is not a new problem, therefore many approaches have been tested involving various techniques. In [3], Fung et al. have proposed a method based on the *efficient market hypothesis* [4] which states that the current market is the assimilation of all the information available. They first found the trend using a piece-wise linear segmentation algorithm based on a t-test. Using agglomerative hierarchical clustering they grouped the useful trends. They then used guided k-means clustering to align the useful news with the trends. A special weighting scheme was then proposed to give importance to the news which support only one type of trend. Finally, the news and trends were aligned and given to an SVM (Support Vector Machine)[5] predicition model.

In 2010 Kaya and Karsligil [6] proposed an approach where each news is labeled based on the change in the stock price for the considered company. They considered Noun-Verb combinations, instead of single words, as features. News articles were divided into samples, with each sample corresponding to a single day. Feature selection was then performed using the Chi-square method. An SVM model was finally used for classification.

Dang and Doung [7] proposed an approach where they labeled the news using a price label (positive, negative and neutral). Furthermore, they created their own financial dictionary for Vietnamese language and tagged the words with parts of speech tags. Only adjectives and verbs were used, and the words in the dictionary were labeled with positive and negative scores based on their frequency in the positive and negative news. They used delta TF-IDF (Term Frequency Inverse Document Frequency) to give an importance degree to the words that are unevenly distributed between positive and negative classes. Term reduction was performed using the OCFS algorithm. This algorithm finds the centroid of the training corpus and scores each word accordingly. After all the processing, they used a SVM to classify the stock price movement.

Deep learning is another way of making stock market predictions. In [8], the authors predicted the stock price using news sentiment score and historical stock prices. Each news article was given a sentiment using python NLTK library. Neutral news were discarded, and for each of the other news the maximum polarity score between positive and negative was taken, then the average score of all the news for each particular day was calculated. The final model used the past prices and the sentiment scores as inputs for the prediction.

Although plenty of research work has been done on the problem of Stock Market Prediction, there is yet to be a single benchmark against which all experiments can be compared. That means that most published works have used different datasets and different evaluation approaches. For example some research works [8] considered stock prediction as a regression problem, while other papers [7] considered it as a classification problem. Some research [9] also focused more on evaluating the correlation between price change and sentiment change and did not even try to predict the price change. Due to these reasons it's very difficult to compare our work to the state of the art research.

3 Contribution

We present in this section the details of our model. First, we introduce the different data sources we consider in our experiments in Section 3.1. We go through their filtering and cleaning process in Sections 3.3 and 3.2. Later on, we describe our system architecture and the configuration of its different parameters in Section 3.4. Finally, we present individually the sentiment-based score and the event-based score for our proposed hybrid approach in Section 3.5.

3.1 Datasets Description

The data used in this paper can be divided into 3 separate categories : price data, stocktwits and news articles. This data has been collected through API channels and Python scraping scripts for a period of 19 months (from 01/02/2019 to 30/09/2020).

3.1.1 Price Data The price data used in this study is collected by the Alphavantage API[3], which has partnered with major institutions, exchange platforms and brokers around the world. As mentioned in the official API documentation, the historical data is derived from the Securities Information Processor (SIP) market-aggregated data, which contains the standard Open-High-Low-Close-Volume time series. The split and dividend events are taken into account using a split/dividend-adjustment in order to prevent misleading price change signals, thus to ensure that the data represents the true movements of the market which can then be used as an input for our technical analysis. The collected price data were the daily prices (Open, High, Low, Close and Volume) for a total of 8 companies from the US NASDAQ100:

AAL (American Airlines Group Inc), AAPL (Apple Inc), AMGN (Amgen Inc), AMZN (Amazon.com Inc), FB (Facebook Inc Common Stock), GOOG (Alphabet Inc Class C), GOOGL (Alphabet Inc Class A), MSFT (Microsoft Corporation), NFLX (Netflix Inc).

3.1.2 Stocktwits The stocktwits data has been collected directly from the official Stocktwits API[4] symbol stream endpoint. The original data contains the message body itself as well as some meta-data such as the timestamp, the likes, the author's information (username, name, followers, following, likes, etc.) in addition to a sentiment hashtag the author has given to his/her tweet. This sentiment can only be "Bullish" (the user is confident that the price will rise in the near future) or "Bearish" (the user is confident that the price will fall in the near future). This sentiment label is optional, so the user may or may not add it before sending his tweet.

The stocktwits have been collected for the same 8 companies as for the price data as detailed in Table 1.

Table 1. Stocktwits distribution by company

Company	Count
AAPL	508,940
AMZN	335,327
FB	183,048
MSFT	172,258
NFLX	165,000
AAL	114,182
GOOGL	41,522
AMGN	7,565

3.1.3 News The news were collected from various resources and stored into an ElasticSearch index. Some examples of the news' sources are The Wall Street

[3] https://www.alphavantage.co/documentation/
[4] https://api.stocktwits.com/developers/docs/api

Journal, The Washington Post, USATODAY, and CNN. There are around 800K news articles. Each article has a publication date, a title, a message and a full-text, collected using RSS feeds and a full text scrapper. The message part represents a short description snippet that is contained in the RSS feed for the articles. Therefore, we have chosen them as the textual data input for our prediction model since the titles may not provide enough information and the full-texts are very long to process. Moreover, full-texts are not proven to systematically perform better than the messages.
However, this news data is not labeled. In other words, we do not know which events or which sentiment an article contains. Furthermore, companies records very different frequencies in terms of news articles. The distribution of the news data per company is detailed in **Fig.1**.

3.2 Data Pre-processing

The collected news data and the stocktwits have to be cleaned as they contain noise such as HTML tags and extra spaces (**Fig.2. step 1**). We use Vader[10] as the Sentiment Analyzer for the sentiment-based approach. Therefore, we do not need to remove punctuation as Vader is capable of handling sentences as they are. Vader is also capable of handling emojis contained in Stocktwits, so those were also not removed. The details about Vader will be discussed in 3.5.1. However, we did have to run a preprocessing pipeline for the event-based feature. The pipeline involved tokenizing, lemmatizing, removing stopwords, removing non-alphabetical characters, and removing time-related words (daytime, months, days of the week). For the news articles, we replaced all the company names with company symbols, for example - "American Airlines" and "American Airlines INC" would both be replaced with "AAL". And for the news articles, we removed company names once the data was filtered. The reason for that was to avoid introducing a frequency bias, since the articles are filtered in such a way that they systematically contain the company name (see Section 3.3), and having the same words systematically present would bias the word embedding (details on the usage of the word embedding will be explained in Section 3.5.2)

3.3 Data Filtering

As our news are taken from various resources, they hold no guarantee of being firm-specific. Therefore, to make sure the news articles are relevant to the 8 companies whose stocks we're predicting, we filtered the news using the company name(**Fig.2. step 2**).
For Stocktwits in order to reduce the noise we filtered the twits and only kept ones which have at least 10 followers and 100 likes and that contain at least 20 words.

3.4 System Architecture and Parameters

There are two important parameters to our experiments. The first one is the Press Observation Period (pop), which defines the number of previous days of

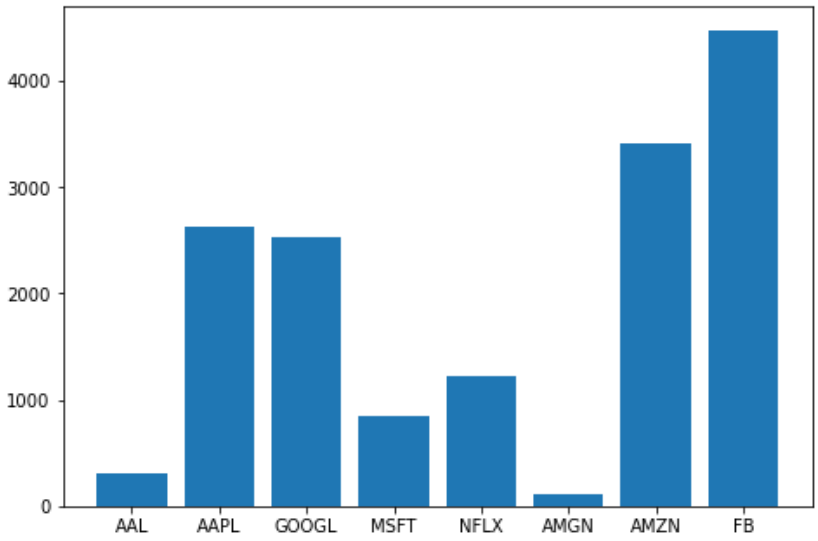

Fig. 1. News distribution based on company name

news articles/stocktwits fed into the model in order to make a prediction. The second one is the Price Change Period (pcp), which determines the day of the close price prediction with regards to the day the prediction is done. For example, if we have (**pop = 3**, **pcp = 1**) and the current date is the day d then we are feeding the model the news/stocktwits of days **d-2,d-1,d** till the closing time of the stock market on day d in order to predict the price movement of **d+1**.

3.5 Text based features and models

3.5.1 Sentiment-based score: Sentiment analysis plays a significant role in extracting the essence of textual data. When it comes to stock movement prediction, it is interesting to study how the stock price movement changes based on the sentiment tone of the news articles and the stocktwits. However, there are many approaches to find the sentiment of a text:
1. Rule based approach: the sentiment is predicted using lexicons and grammatical rules.
2. Machine learning based approach: the sentiment is predicted using a model which has learned through examples.

Machine-learning-based approaches require data labeled by domain experts. On the other hand, rule-based sentiment analyzers[5] use patterns and lexicons and therefore do not need any labeled data. Since our news dataset is unlabeled, we will be using a rule-based sentiment analyzer to get the sentiment score.

[5] Vader(https://pypi.org/project/vaderSentiment/),
Textblob(https://textblob.readthedocs.io/en/dev/)

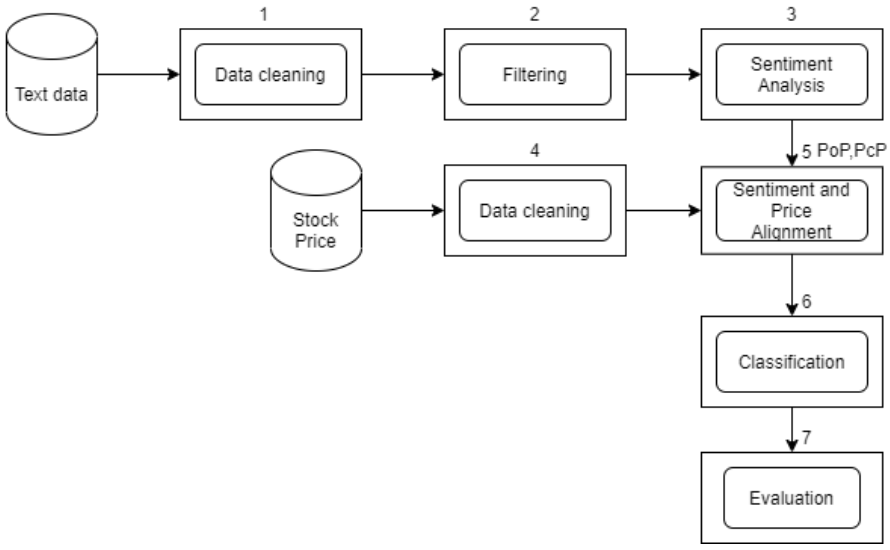

Fig. 2. Overall system architecture

Vader is a very well known sentiment analyzer. It has the capacity to handle grammatical rules such as negation, conjunction and punctuation along with a gold standard sentiment lexicon.

The lexicons used by Vader have a score for each word in the vocabulary between -4 to +4. In [10] it has been shown that Vader has outperformed other rule-based sentiment analyzers and machine-learning-based sentiment classification models. Although it was initially created for classifying twitter sentiment, in [11] the author has used Vader for financial sentiment analysis and has shown that Vader outperforms the machine-learning-based approaches. In [9] the author has shown good correlation between the sentiment change and the stock price change using Vader. As we are dealing with the financial domain, we updated Vader's lexicon with a financial lexicon called SentiBignomics[12],[13],[14]

Inspired by [9], we used Vader in our experiment to calculate sentiment scores(**Fig.2. step 3**). In our experiments we added temporal weights to the sentiment scores, that are linearly decreasing with time. Indeed, the further an article is published from the day we want to predict the price change, the less it counts.

3.5.2 Event-based Score With the advances in the field of event-extraction, many published studies have used event-based features to solve the problem of stock market prediction. Feldman et al.[15] proposed a hybrid approach to stock market prediction using both sentiment analysis and event extraction, the events were extracted using a predicate-level semantic business event extractor

designed by a team of linguistic engineers and financial experts. Han et al.[16] implemented an event-extraction approach for online Chinese news based on an event-trigger dictionary combined with word embedding and deep learning. The current state-of-the-art on event-extraction works at the sentence-level, that is, detecting not only the event (what happened) through a rule-based pattern or a trained model, but also the relevant entities (to whom the event happened). While this kind of approach is the one that currently yields the state-of-the-art results, it usually requires lots of resources (access to rich extensive labeled data sources). In this paper, we will approach event-extraction in a more simple way as we will work at the word-level. The idea is to see how much improvement we can get out of combining a relatively simplistic event-based feature with sentiment-based features.

It has been proven that the events with most affect on stock market prices are the events related to firm-fundamentals. Shao et al.[17] demonstrated that news related to firm-fundamentals explained, on average, 39% of annual returns in the early 2010s, and Kogan et al.[18] found that fundamental firm-level information present in public news accounts for 20-40% of stock price volatility. Therefore, we started by defining a list of seven events closely related to firm fundamentals:

- Product launch.
- Product recall.
- Merge or acquisition.
- Price change.
- Legal related event.
- Bankruptcy related event.
- Financial related event.

We assign a score to each event according to its impact on the stock market. The scores were initially assigned using online finance literature on the impact of corporate events on the stock market. We then performed a series of tests on the training datasets to fine-tune the scores for each event.

Furthermore, inspired by Peng et al. [19], we define an initial list of 10 seed words for each event-category. For example, the seed words for the event-category product-launch were: *product, launch, publish, release, unveil, announce, reveal, introduce, unseal, relaunch.* Once we had a list of seed words for each event-category, we extended those lists using both a financial ontology[6] and a Word2Vec word embedding trained with our finance-specific dataset. The closest words to the initial seed words were generated based on both the financial ontology and the word embedding using cosine similarity. We then kept the top 50 most relevant words for each event-category. Finally, we calculated a vector V = (logN1, ..., logNm) where Nm is the number of words in the news article/stocktwit that belong to the event-category m. If the word count is equal to zero we replace it with a large negative number (e.g. -100).

In order to determine to which event-category, if any, a news article/stocktwit belongs to, we compute the V vector and then pick the event that corresponds to

[6] FIBO: The Financial Industry Business Ontology

its maximum value. However, if two events have the same word count, we pick the event whose weight is the highest. We also defined a minimum threshold (initially set at a high value, e.g. 10, then fine-tuned) for the word count.

3.5.3 Combined Score While we know which events can impact the stock market, for some events it is not obvious to know if the impact will be positive or negative (e.g price-change event). To avoid introducing a bias by giving the event-weight a positive/negative value, we give all the events positive values and we let the sentiment analyzer decide on the score's polarity. To calculate the final score feature, we multiply the sentiment analyzer score by the detected-event's weight.

$$combinedScore = detectedEventScore * sentimentScore \quad (1)$$

If no event is detected, the *detectedEventScore* is equal to 1 and the *combinedScore* is equal to the *sentimentScore*

4 Evaluation

In this section we report the results of seven experiments which evaluate the ability of the system to predict the price change direction (i.e price moving up or down) of the next day (pcp=1) for 8 different companies from the US NASDAQ100 based on various price observation periods (pop). The goal of the experiments is to define the most efficient prediction pipeline (i.e to find, given our datasets and our proposed features, the most efficient combination of data input, model features, price observation period and prediction model to optimize the close prices prediction).

Different models were tested (Linear Regression, SVM, Ensemble models, LSTM) before settling on a Random Forest (RF) model optimized using a Grid search algorithm. The hyperparameters that were optimized using the Grid search algorithm are the number of trees in the RF, the number of features considered before splitting at each leaf node, the maximum depth of the RF and whether or not bootstrapping is used for sampling data points. The RF prediction model is common to all experiments. The experiments differ however in the input datasets and in the features used to train the prediction model.

The first experiment uses only the price data (i.e historical daily close prices for the 8 companies) as a feature for the prediction, and serves as a benchmark to see how the model's performance improves when stacking up more complex features. The second experiment uses only sentiment scoring based on the news data. The third experiment uses only sentiment scoring based on the stocktwits data.The fourth experiment uses a combination of price data and sentiment scoring using only the news data for sentiment analysis. The fifth experiment uses the same combination of price data and sentiment scoring but uses only the stocktwits data for sentiment analysis. The sixth experiment uses a combination of price data, sentiment scoring and event weights based only on the news data.

The seventh and last experiment uses a combination of price data, sentiment scoring and event weights based only on the stocktwits data.

Table 2. F1-scores for American Airlines (AAL) for different pop

pop	3	8	13	18	23	28	33	38	43	48
Stocktwits count	6	14	22	30	38	46	54	61	68	75
News count	2	5	7	10	12	15	17	19	22	24
Price-based only	0.33	0.33	0.32	0.34	0.34	0.39	0.32	0.32	0.27	0.26
Sentiment-only (news)	0.44	0.54	0.46	0.43	0.45	0.48	0.51	0.45	0.46	0.52
Sentiment-only (stocktwits)	0.52	0.54	0.47	0.52	0.52	0.50	**0.56**	0.49	0.48	0.46
Price+Sentiment (news)	0.51	0.47	0.54	0.47	0.55	0.54	0.51	0.51	0.5	0.44
Price+Sentiment(stocktwits)	0.48	0.47	0.51	0.50	0.54	0.54	0.55	0.56	0.53	0.52
Price+Sentiment+Events(news)	0.51	0.47	0.51	0.44	0.52	0.53	0.47	0.53	0.50	0.42
Price+Sentiment+Events(stocktwits)	0.53	0.51	0.53	0.52	0.51	0.51	**0.57**	0.52	**0.57**	0.49

The evaluation metric used to evaluate the results of the experiments is the F1-score, which is the harmonic mean of precision and recall. The benefits of using the F1-score instead of a mere accuracy score is that the F1-score is less sensitive to class imbalance and is more sensitive to False Negatives and False Positives, both of which are important in a real trading scenario. The results of the experiments are presented in Table 2. As the results show consistent F1-scores (in terms of min, max, average and standard deviation) across all 8 companies we only show the results for American Airlines (AAL).

To train the model, we assign to each sample (corresponding to the day d) a price-change label of {-1,0,1} to indicate whether the price will have moved down, not moved, or moved up on the day d+pcp (e.g d+1 with pcp=1 in our case).

The prediction and evaluation pipeline goes as the following: for example, for American Airlines, when predicting the price change movement for a pcp=1, we obtain the highest F1-Score of 0.57 with a pop equal to 32 using the price-based, sentiment-based and event-based features (see results in Table 2). For example, to obtain the price change movement prediction for October the 1st 2020, we used a pop of 32 days, i.e from August 29th 2020 to September 30th 2020. In this time period, we first retrieved all of AAL's daily closing prices to make our price-based feature. Then, we retrieved our textual data (news articles or stocktwits, in this example only stocktwits were considered) and made it go through our preprocessing and filtering pipelines (see Sections 3.2 and 3.3). Once the textual data was preprocessed and filtered, we grouped them based on the publication day and the company name they refer to to form the samples. Each sample contains all of the textual data published on a given day and that refers to a single company. To extract the sentiment-based feature, we ran the Vader sentiment analyzer on each sample and got a score that is the average sentiment score across all textual data contained in a single sample. To extract the event-based feature, we construct the event vector V (see Section 3.5.2) based on all

the textual data contained in a sample. If the maximum value of the vector is superior to the minimum detection threshold, we assign to the sample the event-weight that corresponds to the event who had that maximum value. If no event was detected (i.e no value of the event vector was superior to the minimum detection threshold) then no event-weight is assigned to the sample. Finally, for each sample, we calculated the compound score (see Section 3.5.3). This compound score will be the sentiment-based + event-based feature. For the AAL example, the F1-Score of 0.57 was then obtained by calculating the harmonic mean of the precision and recall when predicting whether the stock price of AAL will go up, down, or won't change from day d to day (d+1) for all the days within the testing period (which accounts for 30% of the whole dataset timespan of 19 months, see Section 3.1), using the price-based, sentiment-based and event-based features.

5 Discussion and Conclusion:

Different conclusions can be drawn from the results of our experiments. First, using sentiment score as a feature of the prediction model systematically increases the performance of the model (both when using filtered news or stocktwits) in comparison with its performance using solely price-based features, which means that we can affirm that the use of textual data can systematically improve the performance of a price-based model. Second, there is no clear conclusion as to which dataset (news articles or stocktwits) generates the most predictive sentiment scores. Third, using event-based features does not systematically improve the model's performance. However, that could be because the extracted event-features are not sophisticated enough. Further experiments ought to be done using state-of-the-art event-extraction methods to conclude whether or not event-based features systematically improve the model's performance. Fourth, our experiments didn't prove the existence of any optimal price observation period (which is defined as the number of previous days of news articles/stocktwits fed into the model as input in order to make a prediction output). It is possible, though not proven, that such an optimal value does not exist, given that the amount of impact that the news articles/stocktwits have on the stock market and the delay for that impact to take place both depend highly on the individual context of the situation and the the content of the news articles/stocktwits around.

References

1. Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, 2014.
2. Jung Hur, Manoj Raj, and Yohanes E Riyanto. Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade. *World Development*, 34(10):1728–1741, 2006.

3. Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Wai Lam. News sensitive stock trend prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 481–493. Springer, 2002.
4. Eugene F Fama. Efficient market hypothesis. *Diss. PhD Thesis, Ph. D. dissertation*, 1960.
5. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
6. MI Yasef Kaya and M Elif Karsligil. Stock price prediction using financial news articles. In *2010 2nd IEEE International Conference on Information and Financial Engineering*, pages 478–482. IEEE, 2010.
7. Minh Dang and Duc Duong. Improvement methods for stock market prediction using financial news articles. In *2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pages 125–129. IEEE, 2016.
8. Saloni Mohan, Sahitya Mullapudi, Sudheer Sammeta, Parag Vijayvergia, and David C Anastasiu. Stock price prediction using news sentiment analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 205–208. IEEE, 2019.
9. Arul Agarwal. Sentiment analysis of financial news. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 312–315. IEEE, 2020.
10. Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.
11. Sahar Sohangir, Nicholas Petty, and Dingding Wang. Financial sentiment lexicon analysis. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 286–289. IEEE, 2018.
12. Sergio Consoli, Luca Barbaglia, and Sebastiano Manzan. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Aspect-Based Sentiment Analysis on Economic and Financial Lexicon (January 14, 2021)*, 2021.
13. L Barbagliaa, S Consolia, and S Manzanb. Forecasting with economic news. *Available at SSRN*, 2020.
14. Luca Barbaglia, Sergio Consoli, and Sebastiano Manzan. Monitoring the business cycle with fine-grained, aspect-based sentiment extraction from news. In *Workshop on Mining Data for Financial Applications*, pages 101–106. Springer, 2019.
15. Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. The stock sonar—sentiment analysis of stocks based on a hybrid approach. In *Twenty-third IAAI conference*, 2011.
16. Songqiao Han, Xiaoling Hao, and Hailiang Huang. An event-extraction approach for business analysis from online chinese news. *Electronic Commerce Research and Applications*, 28:244–260, 2018.
17. Shuai Shao, Robert Stoumbos, and X Frank Zhang. The power of firm fundamental information in explaining stock returns. *Review of Accounting Studies*, pages 1–41, 2021.
18. Jacob Boudoukh, Ronen Feldman, Shimon Kogan, and Matthew Richardson. Information, trading, and volatility: Evidence from firm-specific news. *The Review of Financial Studies*, 32(3):992–1033, 2019.
19. Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. *arXiv preprint arXiv:1506.07220*, 2015.