INVESTIGATING THE CAUSAL LINK BETWEEN THE STOCK MARKET AND TWITTER SENTIMENTS USING CAUSALITY MODELS AND DEEP LEARNING METHODS

by

Narjessadat Seyeditabari

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2018

Approved by:

Dr. Mirsad Hadzikadic

Dr. Wlodek Zadrozny

Dr. Samira Shaikh

Dr. Minwoo Lee

Dr. Jared Hansen

ABSTRACT

NARJESSADAT SEYEDITABARI. Investigating the causal link between the stock market and Twitter sentiments using causality models and deep learning methods. (Under the direction of DR. WLODEK ZADROZNY AND DR. MIRSAD HADZIKADIC)

An investment theory called the efficient market hypothesis (EMH) claims that it is impossible to outperform the market, and therefore, stocks always trade at a fair value. An important assumption of EMH is that all investors make decisions rationally, without any emotional bias. Despite its wide use, EMH struggles to explain why certain types of investments perform better than others, particularly in liquid financial markets (i.e., the stock market). Since the mid-1980s, some have proposed that this is because liquid financial markets are not always as orderly as is assumed by the efficient market advocates. The best explanation for this is the "noise trader" theory of Black [1] and Delong [2], which posits that if some investors trade on a "noisy" signal, asset prices will deviate from their intrinsic value. Examples of noise include investor behavior, news, and social media. Behavioral finance is a new field that specifically studies the cases where non-rational sources cause the classical financial theory to fail.

In this thesis, we investigate the relationship between Twitter and the stock market. To do this, we needed to create a novel training dataset of financial tweets with labeled sentiments. We first used Mechanical Turk to generate a small set of financial tweets, and then designed finance-specific models (using a combination of natural language processing and deep learning) that could accurately predict the sentiments for a much larger set of stock market tweets that span three years. Our final model has an accuracy of 92.7%, which is substantially better than other comparable models. To determine if there is a causal relationship between the sentiments expressed in tweets and the stock market, we applied Granger causality and Bayesian Probabilistic

causality models to our new dataset. We found that there is a significant causal relationship between tweets and a company's stock return at a lag of three hours and one day. Knowing this, it could help investors modify their investment strategies to take into account sentiments expressed in social media.

DEDICATION

To my dad, and his unwavering faith in me. Without it, this never would have happened.

To my mom, and her daring and determined spirit. With her as my guide, I have the courage to do anything.

And to my husband Chris, who never doubted.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AMT Amazon Mechanical Turk

CNN Convolutional Neural Network

DAGs Directed Acyclic Graphs

EMH Efficient Market Hypothesis

GC Granger Causality

LSTM Long Short Term Memory

ML Machine Learning

SA Sentiment Analysis

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

On April 23rd, 2013, the Associated Press' Twitter account was hacked and used to spread false news that there was an explosion in the White House Barak Obama was injured[1]. On the floor of the Chicago Mercantile Exchange, traders quickly reacted to the tweet, selling S&P futures and buying Treasury 10-year futures, which continued until the Associated Press tweeted that its account had been hacked. This is simple example how social media can quickly influence the stock market (among many others) is what motivates the goal of this thesis: to better understand and measure how much social media, and in particular Twitter, actually impacts the stock market.

Towards this end, the first step is to measure the sentiments towards specific stocks that are expressed in tweets. Several studies have used sentiment analysis on Twitter data in the context of the stock market, but most of them did not use a context-specific dataset, nor did they use a sufficiently complex model, which prevented them from predicting sentiments with a high degree of accuracy. For example, Kolchyna et al. [3] combined lexicon-based approaches and support vector machines to classify tweets, resulting in a final accuracy of only 71%. To address this, task 5 of the 2017 SemEval competition [4] challenged researchers to perform fine-grained sentiment analysis on stock market tweets. Jiang et al. [5] won first place in this task by applying an ensemble model consisting of a Random Forest model, a Support Vector Machine, and various regression models; they also combined multiple features, such as word embeddings and lexicons. Our entry for the same task [6] achieved an accuracy only

[1] https://www.usatoday.com/story/theoval/2013/04/23/obama-carney-associated-press-hack-white-house/2106757/

slightly lower than the winning model, but we used a simpler approach. Instead of an ensemble classifier, we used a simple Random Forest classifier, and a context-specific feature set that we engineered based on a financial lexicon from Loughran et al. [7]. In a recent paper, Sohangir et al. [8] found that certain deep learning models worked substantially better than regression models and data mining for sentiment analysis of financial tweets derived from StockTwits[2]. In particular, they found that their CNN performed very well, with an accuracy of 90.8%, while their LSTM did not perform nearly as well, achieving an accuracy of only 69.9%.

In our work, we used Amazon Mechanical Turk (AMT) to precisely label a set of financial tweets to use as our benchmark, and then thoroughly preprocessed this dataset using several techniques. We then needed to create a baseline model so that we could compare the performance of deep learning models to traditional machine learning models. To create the baseline, we built upon our SemEval work [6] by using an SVM instead of a Random Forest as our model, and using TF-IDF instead of a term document matrix for the feature set. Finally, we thoroughly compared different Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs), and found that: (1) the lowest error rate was achieved when using a balanced dataset of positive and negative tweets, and a custom, an involved, preprocessing technique, and a shallow CNN; and (2) the highest accuracy was achieved by a shallow LSTM model with a higher number of cells. This is a significant improvement on our baseline performance, and the performance of previous sentiment analysis work in the context of the stock market [8].

The ultimate goal of this thesis is not to predict the market using Twitter data, but rather, to show to that there is a causal relationship between the stock market and sentiments expressed in tweets and to also characterize the extent of this relationship. To do this, we applied two different causality models, Granger and Bayesian, to detect

[2]www.stocktwits.com

short and long term (hourly and daily) effects of Twitter on the stock market, and visa versa. With Granger causality, we detected a significant causal relationship at three hours and one day before the changes to a particular stock happened. With Bayesian causality, we did not detect a causal relationship at three hours, but we did detect a stronger magnitude of causation at one day compared to Granger. In line with previous work in this area [9, 10], Bayesian causality further showed that the sentiments in our dataset negatively influenced the stock market.

1.2 Sentiment Analysis

With the rise of social networks and micro-blogging, the amount of textual data on the Internet has grown rapidly, and the need to analyze it has increased along with it. Sentiment analysis has emerged as a useful and influential approach for analyzing this type of data to investigate people's emotions and understand human behavior in multiple domains. For example, Bollen et al. [9] used social-media sentiment analysis to predict the size of markets, while Antenucci et al. [11] used it to predict unemployment rates.

Historically, sentiment analysis has been used to analyze longer form documents such as reports, news stories, and blogs. However, the usage of micro-blogging applications, such as Twitter, has spiked. Twitter in particular is regularly used by celebrities, companies, and politicians, as well as students, employees, and costumers. With the proper analysis, this data can be leveraged to obtain a concise understanding of a single topic from differing viewpoints, which many companies and researchers have started to exploit.

While social media and blogging are excellent windows into people's opinions and viewpoints about diverse topics, it can be challenging to analyze their contents because people often develop a topic- or context-specific vocabulary. A word in one context can have an entirely different meaning, or sentiment, in another. For example, in a professional context, the word 'tax' can have a positive or neutral sentiment,

while it generally has a negative sentiment in casual conversations. Therefore, it is often recommended to use a domain-specific approach.

To effectively use sentiment analysis in the financial domain, several shortcomings need to be addressed. First, the datasets used for sentiment analysis are not finance-specific, which by itself contributes to low accuracy in sentiment analysis models [9, 12]. Second, the models used for sentiment analysis make poor predictions because they don't use features specific to finance, and also tend to be too simple. [9, 7, 13, 14].

To investigate the relationship between Twitter and the stock market, we addressed these shortcomings in two novel ways. First, we created a publicly available, annotated dataset of tweets that are specific to the stock market, which did not exist before. This by itself should improve research in this area. Second, mostly basic machine learning classifiers or lexicon-based models have been used to date for sentiment analysis in the context of the stock market. Our neural network model is one of only a few used in this context, and none of the others have better accuracy than ours [8].

1.3 Interaction of twitter sentiment with the stock market

It is now very popular to investigate financial problems using data from various social media platforms, and this is especially true for Twitter data, since it is also real time channel. In previous research, it was suggested (although not perfectly demonstrated) that if it is properly modeled, Twitter can be used to forecast useful information about the market. This can be seen in a study by Th'arsis et al., where they reduced their error rate by one to three percent when predicting the Expected Returns in different industries [12] by including features from a Twitter sentiment analysis from Kolchyna et al. [3] in their SVM. Another example is from Alanyali et al. [15], who found a positive correlation between the number of mentions of a company in the Financial Times and the volume of its stock. Before the emergence of Twitter, it was difficult to quickly mitigate the negative effects of false news or rumors on financial markets. With Twitter, it is now possible, but it is also a double-edged sword

- Twitter can just as easily be used spread false news and rumors and hurt financial markets faster than ever before. This substantial influence on financial markets is one of the most compelling reasons to study the relationship between tweets and the stock market. In this thesis, we investigate this interaction by combining sentiment analysis, deep learning, and causal analysis to address two questions: to what extent do sentiments expressed on Twitter influence the stock market, and conversely, how much influence does the stock market have on the emotions of people when they talk a certain stock?

1.4 Thesis summary

Chapter 2 is a literature review of sentiment analysis in general, and the financial domain in particular. We will also review current literature about the how sentiment analysis has been used to study the financial market and predict changes. In Chapter 3, we will cover the sentiment analysis models, and the financial Twitter dataset, that we created to study the relationship between Twitter and the stock market. In this chapter, we will introduce labeled three years twitter dataset that has been labeled using our model. The model that we created for labeling the tweets in this domain, has lower error rate and higher accuracy compared to the state-of-the-art sentiment analysis work in the context of the stock market [8]. In Chapter 4, we will describe the causality analysis we performed. We applied two different causality models on the Twitter dataset and stock market returns. We show that there is indeed a causal relationship between the stock market and sentiments expressed in tweets, and quantify its extent. We demonstrate this causal link in different intervals, and finally, concluded that Granger causality shows significant result on three hour and one day intervals. The Bayesian causality models, only demonstrated causal link at one day interval. In Chapter 5, we will summarize our findings and methods.

The contribution of the thesis is in three different ways. First, we provide a public Twitter dataset with labeled sentiments that is specific to finance [16]. Second, we

provide a highly accurate deep learning model for labeling new financial tweets. This model, can be used to label more tweets in this area. Finally, we show that there is indeed a causal relationship between social media (in this case, Twitter) and the stock market. By applying multiple causal models, we identify various characteristic of this causal relation such as, weight, interval of time-series, and finally the time it take for the causation to take an effect. Taken together, we expect that this research will open new avenues for further research in this area.

CHAPTER 2: LITERATURE REVIEW

Typically, sentiment analysis starts with data pre-processing and feature selection. In the case of machine learning approaches, all data will be labeled using these features. Pre-processing and feature selection often plays an important role in the result of the model, because text is very sensitive to noise. For example, spelling problems in a text can have substantial effect on sentiment analysis models. The next step in sentiment analysis is to analyze the labeled data with a particular sentiment analysis method, with lexicon-based and machine learning methods being the two most common types. After applying the method, the predicted sentiments will be evaluated in the final step. This chapter will review the literature on different methods on sentiment analysis, and specifically, different models on labeling text in the context of financial market. We will finish this chapter by a review of literature on causal link between social media (e.g. news, tweets, and etc.) and stock market.

2.1 Pre-processing for text

Before applying any method of sentiment analysis to text, it is crucial to apply pre-processing methods and feature selection methods, which have substantial effects on the results of sentiment analysis. Following is a description of the most typical techniques that are used for this purpose.

2.1.1 Part-of-Speech Tagging (POS)

Since adjectives and adverbs are the parts of speech that best demonstrate sentiment in any sentence, POS tagging is used to identify the phrases in a piece of text that contain them. The most common feature is to extract two or three consecutive words from texts Turney [17], and then use the frequency distribution of these parts

of speech to extract patterns. Part-of-speech tagging for correct sentiment analysis was introduced by Brill [18], and its importance was shown by Manning [19]. Pak and Paroubek [20] used POS tagging with Twitter messages, which showed that context-sensitive texts usually contain more pronouns. The effectiveness of POS for sentiment analysis is still an open question. For example, Go et al. [21] and Kouloumpis et al. [22] reported no improvements using POS for sentiment analysis, whereas Agarwal et al., Barbosa and Feng, and Pak and Paroubek showed at least small improvements [23, 24, 20].

2.1.2 N-grams

N-grams are used to create a bag of words with different length. A unigram is a single word used in the text, while bigrams and trigrams are the two and three length phrases used in the text, respectively. There is no consensus on the most effective length of the n-grams. Kouloumpis et al. [22] showed that using unigrams and bigrams is effective in improving sentiment analysis, with trigrams performing poorly. Pang et al. [25] reported that unigrams perform better compared to bigrams on sentiment classification of movie reviews, while Dave et al. [26] reported that bigrams and trigrams worked better than unigrams for polarity classification of product reviews. Pak and Paroubek [20], "is" and "have") and using negation (e.g., "not", "cannot") are usually a good idea before creating these tokens.

2.1.3 Stemming and lemmatization

In this approach, every word is converted to its root. For example, words like sadness, sadly and sadder would all be converted to sad. This approach is especially helpful when using lexicon based methods of sentiment analysis, as it reduces the length of your dictionary or bag of words, therefore searching in dictionary will be easier. However, there are risks to this process, since over stemming or under stemming the words might lead to wrong sentiment analysis.

2.1.4 Stop-words removal

This is one of the most common pre-processing steps in sentiment analysis. Stop-words are defined as any word that has a connecting function in sentences (e.g., "as", "is", "on" and "which"), and before applying any sentiment analysis method, they should be removed from the text because they occur frequently but have very little impact on the semantics of a sentence, because they can impact the overall sentiment of the text.

2.1.5 Negation Handling and But-clauses

Negation handling is needed when negation is used in a sentence; negation words include not, cant́, couldnt́, wont́, dont́, neither, nor, lacks, etc. Although there is not a specific list of negation words, they need to be considered in pre-processing of text. The usual process for handling these words is to negate the semantics of the sentence without that word. Similar to negation words, but-clauses usually have the same impact on the sentence. It is critical in a sentiment analysis to consider both of these two categories in pre-processing analysis.

2.1.6 Micro-blogging Characteristics

In addition to the general pre-processing and feature selection approaches previously discussed, there are some that are specific to micro-blogging. For example, using emoticons in short text to evaluate emotion Kouloumpis et al. [22] or using hashtag datasets in tweets are common types of feature extraction. Popular sources for emoticon data include Internet Lingo Dictionary Wasden [27], Wikipedia and the Emoticon dataset developed by Go et al. [21] for Stanford University.

2.1.7 Combining Methods

Generally, it is not an easy task to decide which pre-processing or feature selection method (or combination) to use because most of these methods produce different

results depending on the dataset. To address this, Kouloumpis et al. [22] evaluated and compared the performance of different combinations of methods, and showed that using n-grams, along with bag of words and emoticons, improved their sentiment analysis, while adding POS decreased performance. In contrast, Agarwal et al. [23] showed that adding POS tags, improved their lexicon based analysis.

2.2 Sentiment Analysis Definition and methods

The recent explosion of textual data presents an unprecedented opportunity for investigating people's emotions and opinions, and for understanding human behavior. Although there are several methods to do this, sentiment analysis is an especially effective method of text categorization that assigns emotions to text (positive, negative, neutral, etc.). Although methods for text categorization were introduced a long time ago Salton and McGill [28], sentiment analysis is a much more recent addition Das and Chen [29].

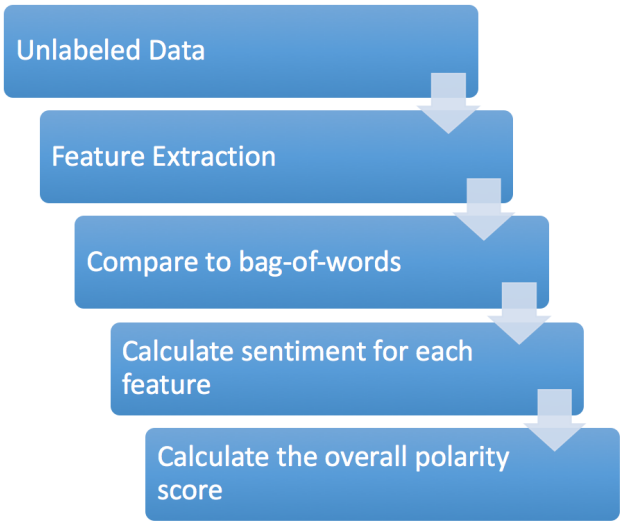
Sentiment analysis methods have been used widely on blogs, news, documents and microblogging platforms such as Twitter. It has also been used on customer reviews Turney [17] to give a "thumbs up" or "thumbs down", movie reviews Pang et al. [30], financial blogs O'Hare et al. [31], and a combination of financial news and Twitter Souza et al. [12]. The widespread use of Twitter by people from a variety of backgrounds and professions[1] , made it one of the most popular sources for performing sentiment analysis on text. Although the variety and volume of topics in Twitter datasets make them especially appealing for this type of analysis, these same features also make it extremely challenging.

Dealing with different types of datasets is in general not a trivial task, and the challenges when analyzing larger documents, such as blogs and news, are different from those involved with analyzing short texts. The pre-processing steps involving

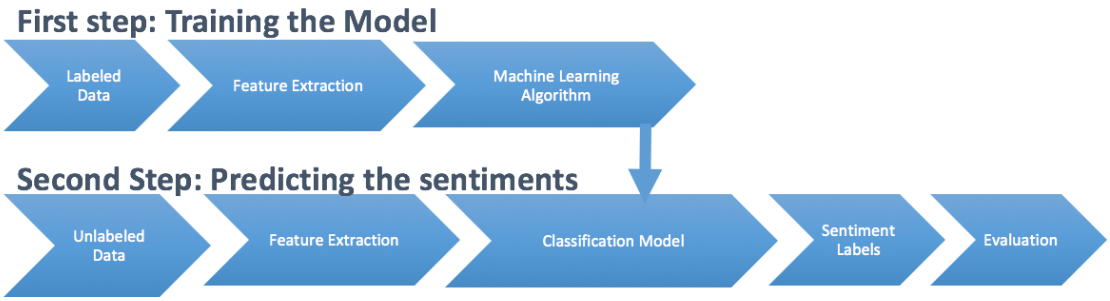
[1]The average number of tweets per day has grown from 5,000 in 2007 to 500 million in 2016. http://www.internetlivestats.com/twitter-statistics/

negation words, stop-words, and tokenization are considered to be general challenges for the analysis of any type of text, while pre-processing text from micro-blogging platforms, such as Twitter, have their own challenges, including the abbreviation and misspelling of words, incorrect grammar and syntax, and the use of URLs, pictures and emoticons. These platform-specific features make the pre-processing a critical part of sentiment analysis for short text.

The initial approach for text representation Salton and McGill [28] was using bag of the words method. In the studies after that, lexicon-based method and machine learning supervised method, the two main approaches to sentiment analysis, both somehow rely on the bag of words method. In the machine learning method, these bag of words are being used as a classifier, whereas in the lexicon-based approach they are being used in order to assign a polarity score to text. Then, the overall polarity score of the text will be calculated with various formulas from those polarity scores; yet the most common computation is a simple summation of all polarity scores. Figure 2.1(a) shows process of this approach. Another approach is using machine learning classifiers. In machine learning an already labeled dataset will be used to to identify the features and design a proper classifier model. Then the machine learning classifier will be applied on an unlabeled dataset to assign sentiments. Figure 2.1(b) shows the process of sentiment analysis in supervised machine learning approaches. In the recent years, deep learning techniques has had an important role in the progress of the sentiment analysis results. In the following sections, we will only focus on reviewing the literature on machine learning and deep learning approaches for sentiment analysis, since we are not exploring lexicon based methods.



(a) Process of sentiment analysis using lexicon approach. In this approach usually a bag of word or dictionary will be used on unlabeled data to detect the sentiments of words, and then using a formula an overall sentiment of sentiment will be calculated.



(b) Process of sentiment analysis using supervised machine learning method. This approach uses labeled datasets to train the model. Once the machine learning model is designed using appropriate features, the model will be applied on an unlabeled dataset to predict the desired sentiment labels. An evaluation method will evaluate the model in the end.

Figure (2.1) Process of Sentiment Analysis using Lexicon and Machine Learning Approach

2.2.1 Machine Learning

Machine learning in sentiment analysis is mostly a supervised classification method which uses a dataset that has previously been labeled with a polarity score (or sentiment orientation). They create and select features that are the input for a learned model than can differentiate between those labels (classes) in a never seen, unlabeled dataset. Machine learning techniques has been very popular in microblogging and Twitter platforms, with the most common methods being Naive Bayes, Maximum Entropy, Support Vector Machines, and Logistic Regression. The general process of applying these learning algorithms starts with pre-processing and feature selection on the data, which is a critical step. Redundant features reduce the speed of the algorithm that leads to a decrease in the quality of classification. Mitchell describes the basics of machine learning very well in his book Mitchell [32].

During the feature selection step, features such as bag of words and combination of words [21, 20], Part-Of-Speech tags [23, 24] (with or without words' prior sentiment), and the syntax features of tweets (e.g., hashtags, retweets, punctuations, etc.) [22] are widely used. Learning algorithms have demonstrated various results with different datasets and choice of features. Therefore, every feature selection or machine learning model could result very differently on the same dataset, and feature selection process and choosing the machine learning model that suits the data is crucial.

After feature selection, the dataset can then be used to train a learning algorithm, with Naive Bayes and Multinomial Naive Bayes being one of the most frequently used methods. Generally, with this classification method, a sentiment will be determined for a word if the probability of belonging that said word to the same sentiment class is higher. The one restriction of this method is that words in the training dataset must belong to all sentiment classes with different probabilities. Saif et al. [33] used set of semantic features. They examined the semantic concepts that serve extracted entities and incorporated these features to Naive Bayes classifier. These

features improved F-score accuracy 6.5% and 4.8% compared with unigrams and POS features, respectively. Another study used Multinomial Naive Bayes for a binary classification of blogs in interval of word, sentence and paragraph O'Hare et al. [31]. Using simple bag of word features to train the model, the study showed that this produced a substantial improvement over full document classification, and that word-based approaches perform better than sentence-based or paragraph-based approaches.

A Support Vector Machine (SVM) is another widely used supervise learning method for sentiment classification. The basic idea is to find a hyperplane that separates the text into classes so that the distance between the words in one sentiment class is maximized to the nearest word in another sentiment class. In a multi-class sentiment analysis, SVM performs pair-wise classifications between each of the labels. In general, SVMs has shown to do well in text classification. For example, one study Mohammad et al. [34] performed an SVM-based sentiment analysis on a tweet dataset from SemEval-2013 [35]. They showed that the combination of n-gram, POS, capital words, hashtags, lexicons, punctuations, emoticons, and negations feature selection techniques with an SVM performed better than unigrams, obtaining an F-score of 69.02% for the message-level analysis and 88.93% for the term-level task. By using a different set of features for a sentiment analysis on tweets, which included verb groups and adjectives from WordNet, senti-features from SentiWordNet, various dictionaries of emoticons, abbreviations, and slang Hamdan [36], an SVM classifier improved the F-score by 2% compared to the SVM used in Nakov et al., while a Naive Bayes classifier improved it by 4%.

In addition to these two commonly used methods, Decision Trees have been used in sentiment analysis; here, non-leaf nodes represent a conditional based on a feature, branches are the possible decisions, and leaves are the class labels. Although decision trees have several advantages, such as transparency and direct information about feature importance, they are prone to over-fitting (a significant disadvantage) since

the training dataset split with every decision. Using an ensemble method to combine classifiers is a promising approach can improve the accuracy of classification. In one study, Random Forest, SVM, MNB, and Logistic Regression were combined Da Silva et al. [37]. They implemented two different approaches for feature representation: bag-of-words and feature hashing. Moreover, bag-of-words and lexicons were more effective than hashing features.

Although using machine learning for sentiment analysis on Twitter data has been successful, its success is limited because Twitter data has a high rate of conversions and modifications, which requires frequent and time-consuming re-training of the model Liu [38]. Go et al. [21] used a distant supervision approach which generates an automatic training data using the emoticons used in the tweets. This approach increased the error rate of the analysis which may affect the performance of classifiers Speriosu et al. [39]. Another limitation of machine learning methods is that often a classifier trained in one context does not generalize well to a different context, which can drastically reduce the accuracy of its predictions [40].

2.2.2 Deep Learning and other approaches

Deep learning is a relatively new area in Machine Learning that models data with multiple levels of abstractions, and to learn these levels, it uses neural networks with multiple layers. Typically, all the deep learning models in text-related tasks start with word embedding from text collection, which is then used to create representations of the documents.

Interest in using deep learning for sentiment analysis has spiked recently, and can be seen by comparing the methods submitted for SemEval-2014 and SemEval-2015 to those submitted for SemEval-2016. In SemEval-2015 task 10, the majority of submitted methods for sentiment analysis on Twitter data used SVM, maximum entropy, CRF, or linear regression as the classifiers, while the most common feature types were bag of words, hash tags, handling of negations, word shape and punctuation.

None of the top-ranked teams in SemEval-2014 and SemEval-2015 used deep learning models for tasks involving sentiment analysis on Twitter data. In contrast, 5 of the 10 top-ranked teams in task 4 of SemEval-2016 [4] used deep neural networks, and 7 teams used word embedding features generated from word2vec or G1oVe.

To compute semantic information on large text datasets, Maas et al. presented a vector space model that learns word representations [41]. A deep convoluted neural network was applied to two different datasets: The Stanford Sentiment Tree-bank (SSTb), which contains sentences from movie reviews, and the Stanford Twitter Sentiment corpus (STS), which contains Twitter messages [42]. They included two convolutional layers in their model, allowing the model to handle words and sentences of any size. Their model for single sentence sentiment prediction achieved an accuracy of 85.7% for the SSTb corpus and an accuracy of 86.4% for the STS corpus.

In another study, sentiment specific word embedding (SSWE) was learned from tweets that were collected using distant supervision by Tang et al. [43]. To do this, they developed three neural networks that were then used as features for Twitter sentiment analysis. The methods were evaluated on the SemEval-2013 dataset, and the best result, which combined SSWE with sentiment lexicons and the same features used by Mohammad et al. [34], had an F-1 score of 86.58%. SSWE was combined with a number of features, including sentiment lexicons, emoticons, and emphatic lengthening, and when evaluated on the dataset SemEval-2014 [44], it obtained the second best ranking with an F-1 score of 87.61%.

Dong et al. [45] proposed an Adaptive Recursive Neural Network (AdaRNN) for entity-level Twitter sentiment analysis that used a dependency tree to find the sentiment words syntactically related to the target then propagate the sentiments associated with these words to the targets. The method was evaluated on a manually annotated dataset consisting of 6,248 training and 692 testing tweets and obtained an F-1 score of 65.9%. Using this same dataset, Vo and Zhang [46] developed a method

that used a rich set of automatically generated features that outperformed AdaRNN with an F-1 score of 69.9%. In their approach, a tweet is split into a left and right context in relation to a specific target. Word embedding are then used to model the interactions of the two contexts that were used to detect the sentiment towards the target. The authors explored a range of pooling functions for automatically extracting the rich features.

In another approach, Severyn and Moschitti used word2vec as the word embedding tool in the neural language model and was trained on large collection of tweets [47]. The resulting network model was composed of a single convolutional layer followed by a non-linearity, max pooling, and soft-max classification layer. After building the model, it was tested on a supervised corpus from Semeval-2015 and ranked in the top two positions for both the phrase-level subtask A (among 11 teams) and the message-level subtask B (among 40 teams).

2.3 Evaluation Techniques

Since in sentiment analysis we want to predict a sentiment (i.e., class or label) for each, with the sentiments typically being positive, negative, or neutral, it is a standard classification problem and therefore most of the evaluation techniques widely used for classification also apply to sentiment analysis. A confusion matrix, shown in Table 2.1, is the foundation for evaluating performance of a classification technique; in our case, we use this to evaluate how well a sentiment analysis classifier predicts whether a text is positive or negative.

TP: Cases that were predicted to have a positive sentiment and were actually positive.

TN: Cases that were predicted to have a negative sentiment and were actually negative.

FP: Cases that were predicted to have a positive sentiment but were actually negative.

FN: Cases that were predicted to have a negative but were actually positive.

Table (2.1) A confusion matrix used to evaluate classifiers.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FP
Actual Negative	FP	TN

The most popular evaluation metrics used in sentiment analysis are accuracy, precision, recall, and F-score and they are based on the confusion matrix. For binary sentiment analysis, all four of these metrics are used, but the F-score is generally the only one used for multi-class sentiments.

Accuracy: the proportion of correct classifications (measurements) to the total number of classifications. It is one of the most common methods used in sentiment analysis.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Also called positive predictive value, this is the proportion of "positive" predictions that are actually "positive".

$$Precision = \frac{TP}{TP + FP}$$

Recall: Also called sensitivity, this is the proportion of "positive" results that are correctly predicted. Recall can be viewed as a classifier's completeness, since a low recall indicates many false negatives.

$$Recall = \frac{TP}{TP + FN}$$

F-score (usually $F_1 score$): This measure of accuracy is a balance between precision and the recall. Since accuracy alone is not sufficient for evaluating a classifier, it is

often coupled with an $F_1 score$, which is the harmonic mean of recall and precession.

$$F_1 score = \frac{Precision \times Recall}{Precision + Recall}$$

One of the main problems in evaluating the models built for sentiment analysis is the small number of benchmark datasets. To create one, it is common to manually assign sentiments to a set of text, and a popular, although expensive, way to do this is to crowd source it using Amazon's Mechanical Turk platform, as several studies have done [21, 35, 48]. The main problem with manual assignment of sentiments to datasets, is the potential biases of the person performing the task.

2.4 Sentiment Analysis in Financial Context

Domain-specific sentiment analysis is being used to analyze or investigate various areas in finance, such as corporate finance and financial markets, investment and banking, asset and derivative pricing. Ultimately, the goal is to understand the impact of social media and news on financial markets and to predict the future value of stocks.

Loughran and McDonald [7] showed that using non-business word lists for sentiment analysis in a business context is inappropriate, because it will produce misclassification and misleading results. To illustrate this, they used the Harvard-IV-4 list on a business dataset and found that three-fourths (73.8%) of the negative word counts were attributable to words that were not actually negative in a financial context. For example, liability and depreciation often have a negative sentiment in a non-financial context, but they are neutral in a financial one, especially in financial 10-K reports. To improve the performance of sentiment analysis in the financial domain, they developed an alternative negative word list, along with five other word lists for other sentiments, that better reflect sentiment in financial texts.

To investigate the correlation between tweets and market movement, financial context sentiment analysis was implemented by applying SentiWordNet's word list [49].

In this study, the probability of an entire tweet being "happy" and "sad" was calculated the log-probability of all its tokens, and then using these probabilities, Chen et al. [49] calculated the sentiment percentage of all tweets per day by counting the frequency of "happy" or "sad" tweets in a given day.

O'Hare et al. [31] identified topic shifts within financial blogs using sentiment analysis. They first created a lexicon of financial blogs annotated with their sentiment polarity with respect to companies' names, and then it to show that their word-based approach performed better than sentence-based or paragraph-based approaches.

Devitt and Ahmad [50] combined POS tagging and WordNet with a lexical cohesion graph based method in order to calculate the sentiment intensity and polarity in financial news and then compared it to the polarity score of words in SentiWordNet. Their Basic Cohesion Metric marginally outperformed the baseline, indicating that there is some benefit to exploiting the graph structure.

2.5 Diagnostic relationship of Sentiment Analysis and Stock Market

Recently, studying the impact of sentiment analysis on business and economic problems has attracted the attention of researchers. The efficient market hypothesis (EMH) is an investment theory claiming that it is impossible to outperform the market, and therefore, stocks always trade at their fair value. EMH also assumes that all investors make decision rationally, without any bias from emotions. Despite its wide use, EMH cannot explain why certain types of shares tend to perform better than others from the point-of-view of investments. Since the mid-1980s, some have proposed that liquid financial markets are not always as orderly as is assumed by the efficient market advocates. For example, as the "noise trader" theories of Black [1] and De Long et al. [2] suggest, if some investors trade on a "noisy" signal that is unrelated to fundamentals, then asset prices will deviate from their intrinsic value. Since then, there have been various attempts to identify and study the effect of investors, news, and other behavioral noise on the financial scope. Behavioral finance is a new field

in finance that combines these sentiments with finance to understand some of the anomalies that classical financial theory fails to describe. The rest of this paper will focus on studies that investigate the impact of sentiment analysis on financial domain.

Baker and Wurgler [51] presented evidence that investor sentiment may have significant effects on the cross-section of stock prices. They created an index for investor sentiment that is based on the common variation in six underlying proxies for sentiment, such as returns of IPO, dividend, number of IPOs, and number of turnovers. Adding the sentiment index into various time series regression analyses, they showed that several firm characteristics that display no unconditional predictive power actually hide strong conditional patterns when adding the sentiment.

In the extraordinary study by Loughran et al., the reaction of the market is examined the at the time of a 10-K report filing [7]. Using multivariate models and multiple control variables in a regression analysis, they showed that on the date of filing, these reports have a negative impact on company returns. Importantly, these results suggest that textual analysis can help us understand the impact of information on stock returns, and even if tone does not directly cause returns, it might be an efficient way for analysts to capture other sources of information.

One of the earliest studies that used typical sentiment analysis to show that high negativity in news predicts lower returns up to 4 weeks after the release of the story. In particular, they used Harvard's word-list, applied regression modeling, and autoregressive (VAR) models to show that the pessimism (negative sentiment) generated by the Wall Street Journal's column [10]. reduced the volume and returns on the NYSE and DOW JONES, as well as Fama-French's Small-Minus-Big returns.

Davis et al. [52] investigated the effect of optimistic and pessimistic language in financial press releases on future firm performance. They used DICTION 5.0 to measure the usage of optimistic and pessimistic language in each of the quarterly earnings press releases in the news, and then applied this measure into a regression

model consisting of accounting and financial market variables and concluded two things. First, readers form expectations about the habitual bias of writers, and second, that they react more strongly to reports which violate these expectations. This strongly suggests that readers, and by extension the markets, form expectations about and react to the affective aspects of text, not just its content.

Tharsis et al. used an SVM and lexicon-based approach based on Kolchyna et al. [3] for their sentiment analysis on tweets relating to certain retail industries [12]. In particular, they determined the impact of sentiment scores on their corresponding stock returns and volume using Granger causality.

Alanyali et al. found a positive correlation between the number of mentions of a company in the Financial Times and the volume of its stock [15].

Lillo et al. [14] applied General Inquirer to measure the absolute or relative difference between positive and negative words in the sentiment of news arriving into the market for a specific company. The main analysis of the paper is a linear regression and partial correlation analysis which showed that the sentiment time series is correlated with the company return, and additionally, both the return and the sentiment of news can effectively explain trading polarization dynamics.

Yu et al. [53] found a causality between sentiment and the volatility and liquidity of FTSE100 and DJIA30 sectors using the Thomson Reuters News Analytics (TRNA) sentiment engine and ARCH/GARCH predictive models.

Sprenger et al. [54] used Naive Bayes sentiment analysis to classify stocks based on tweets. They then describe a methodology to analyze market reactions to different combinations of types of news events, which suggests which type of news is more important from the perspective of an investor.

Ranco et al. [13], by applying a manually labeled dictionary of financial words and a SVM method, found that there was low Pearson correlation and Granger causality between tweet volume and sentiment and the DJIA index of various companies over

time. However, they found a statistically significant dependency between abnormal returns and the corresponding sentiments in high volume tweets several days after the event.

Shen et al. [55] used emoticons, stop-words, stemming, N-gram, and Twitter-specific features for preprocessing the data. They performed sentiment analysis on financial tweets relating to multiple companies over time using SVM. Then, they used Granger causality to show that this time-series of sentiments (i.e., movements of tweets) could predict a company's stock returns.

Bollen and Pepe [9] applied OpinionFinder and Google-Profile of Mood States (GPOMS) on tweets to assign multiple moods (Calm, Alert, Sure, Vital, Kind, and Happy), and then using Granger causality analysis and a Self-Organizing Fuzzy Neural Network, show that public moods can predict changes in the closing values of the DJIA. When using their method to predict the daily closing value of the DJIA, they found that by including a sentiment score, they reduced the Mean Percentage Error of accuracy by more than 6% (accuracy = 87.7%) compared to not including it.

2.6 Chapter summary

Reviewing the sentiment analysis domain, made it clear that there are some aspects in this field that needs to be improved. First, the datasets used for sentiment analysis are not finance-specific, which by itself contributes to low accuracy in sentiment analysis models [9, 12]. Second, the models used for sentiment analysis make poor predictions because they don not use features specific to finance [9, 7, 13, 14]. Therefore, it is necessary to improve on previous models, and better yet, create models that can predict sentiments for the financial text more accurately, with incorporating features from the context.

In the literature review of causal impact between social media and finance, the models have mostly been a simple Granger causality. Most of causality models have used either regression models [7, 51], some variation of ARIMA-GARCH models [10,

53], or Granger causality models [9, 55, 12]. What has been missing in these analyses was first to look at the Granger link in different intervals of data (i.e. hours, minutes, seconds.). Second, lack of applying other causality models is very obvious.

In the next chapter, we attempt to address the shortcomings of sentiment analysis, by introducing a model that can predict sentiment of a tweet with high accuracy. In chapter 4, we first applied Granger causality in different intervals of the stock return. We find that causal link exists in 3 hour and 1 day interval of data with lags of 2 days. Then we apply a Bayesian causality model for time-series on our datasets. With Bayesian causality, while detecting a higher causal weight compared to Granger models, we only detected causal link in one day interval.

CHAPTER 3: MODELS OF TEXT

This part, we first will describe three different approaches for labeling tweets. After that, in the next chapter, we investigate our causality models identifying the relationship between sentiment analysis of tweets and stock market returns.

In first part of this chapter, we first analyze a relatively small dataset that was introduced by SemEval 2017, Task 5 [6]. We examine various machine learning models, such as Random Forest, SVM, Linear Regression, and Naive Bayes, on this dataset. These models worked really well for stock market data introduced by SemEval 2017, and helped us to better understand the different aspects and challenges of analyzing text with a specific context. But, there was a few reasons we could not use the dataset for the causality models. The shortcomings were that first, it was a really small dataset (only about 1800 tweets), and second this dataset did not have dates assign to its tweets. Therefore, we only used this model, as a good instance for our baseline model. In the second part of this chapter, we first labeled a larger benchmark dataset using Amazon Mechanical Turk (AMT), and then created a deep learning model that can predict the stock market tweets.

To the best of our knowledge there was no already labeled tweets on stock market, that is large enough for more sophisticated approaches. This is why, a lot of advanced machine learning and deep learning models has never been tried on tweets targeting stocks. For part two of analysis, we labeled a larger dataset in duration of three months using the Amazon Mechanical Turk. Then, we used machine learning classifiers in order to re-produce the same sentiments produced from AMT as our baseline model. We improved our baseline accuracy, using Convolutional Neural Network and Long Short Term Memory network. Ultimately, we created a highly accurate

dataset in duration of three years, that will be used in chapter 4, in investigating the relationship between stock market and sentiment scores.

3.1 Part 1: An application of Sentiment Analysis on Stock Market Tweets

3.1.1 Data pre-processing and feature selection

SemEval task 5, subtask 1 provided a training dataset with 1800 tweets. Every tweet had a sentiment score between -1 to 1 and it showed its sentiment toward the stock symbol that was assigned to that tweet. Table 3.1 describes variables in the training dataset we used for analyzing the tweets.

Table (3.1) Attributes used to create the sentiment classification model.

Label	Description
ID	Each tweet was assigned a unique ID
Span	Part of tweet that carries the sentiment of the stock.
Sentiment	Score provided to us with numbers between -1 to 1.
Cashtag	Stock symbol that was the target of each tweet, e.g. $GE.

To prepare the dataset for classification, we first converted the sentiment scores to -1, 0 and 1. Tweets with sentiments between -0.01 and 0.01 were labeled as zero, positive sentiments labeled as 1 and negative tweets were labeled as -1. We then disregarded the tweets with neutral sentiment, which left us 1560 tweets to train our model. Some tweets had multiple Spans, describing the sentiment toward the Cashtag. To keep things simple, we concatenated the spans of each tweet with each other. Then using the Python NLTK library we deleted the punctuations, tokenized the spans, and deleted the stop words. Since certain stop words in financial context can have impact on the sentiment of the tweets, we excluded them from the stop word list. Words like "up", and "down" were not removed from tweets. We also removed the negations from the stop word lists, as we later handle the negations on our own

when creating the features.

To add features to our training dataset, we used the Loughran et al. wordlist [7]. This is a list of positive and negative words for financial 10-K reports containing the summary of the company's performance. We calculated number of positive or negative words in each Span, using the Loughran et al wordlist in the added features. There were some words, such as "short" which was not in any wordlist as a negative word, yet shorting a stock expresses a negative sentiment toward that stock. For this reason, we manually added positive or negative words to each list that to our best knowledge carry those sentiments. Table 3.2 shows some of the words were added to Loughran et al wordlist.

Table (3.2) Example of the words added to Loughran et al wordlist.

Word	Sentiment
Profit	Positive
Long	Positive
Short	Negative
Decay	Negative

Adding these words to the wordlist improved our results. Then we realized in context of finance, co-occurrence of some words with each other in one tweet changes the sentiment of the tweet completely. For example, "short" and "sell" are both negative words in context of finance, but selling a short contains a positive sentiment in stock market context. Another example would be the co-occurrence of "go" and "down", or "pull" and "back" in our tweets. In a similar fashion we also we handled the negations. Once we found these patterns, we normalized our data, i.e. we replaced the combinations of words in the tweet with a single positive or negative label, which we treated just as another positive or negative word. We then re-counted the number of positive or negative words in the tweet and updated our feature vectors. Table 3.3

shows examples of patterns we found in the tweet to have changed the sentiment of the word. The normalization had a benefit of increasing the counts of rarely occurring examples.

Table (3.3) Example of the word couples and their replacements used to normalize the data (tweets). Replacing word couples with one positive and one negative word will reduce dimensionality and will normalize the data for better accuracy in prediction.

Word 1	Word 2	Replaced with
Go	Up	OKAY
Go	Down	NOTOKAY
Sell	Short	OKAY
Pull	Back	NOTOKAY

3.1.2 Comparing different Machine Learning methods and results

After pre-processing our data and creating all our features (Tweet, Positive-Count, Negative-Count), we used WEKA to classify our tweets. Our feature vectors were the combination of document vectors generated by Weka's StringToWordVector filter which is a term document matrix, followed by the features extracted from the data as explained above. Among all the classification methods that we used, Random Forest did give us the best result with accuracy of 91.2%. Table 3.4 shows results from various classifiers using our training data. The random forest model in WEKA provided both a class prediction and class probability for each tweet in the training and test set.

Table (3.4) Results of different Weka classifiers using 10-fold cross validation and default settings

Classifier	Accuracy	F-score	Precision	Recall
Random Forest	**91.26%**	86.50%	91.30%	82.20%
SVM	90.43%	85.40%	88.90%	82.20%
Logistic Regression	84.69%	79%	74.30%	84.30%
Naive Bayes	83.73%	73.30%	83.30%	65.40%

Since the final float score needed to be between -1 and 1, for tweets classified as negative we made the sentiment score the negative of the class probability; for positive classifications, the sentiment score was simply the class probability. It is interesting as another research idea to see what made Random Forest to work better that other machine learning algorithms with this dataset. The winner of this competition used, Jiang et al., linguistic features, sentiment lexicon features, domain-specific features and word embedding feature and then employed these features to construct models by using ensemble regression algorithms [5].

SemEval-2017 training dataset was a relatively small dataset, which would prevent us from implementing any neural network models for prediction. Therefore, we think a step to create a better model is to increase the size of training dataset. Next section is the process of expanding this dataset to a larger domain specific one. We believe this dataset will be extremely helpful to other researches that would like to investigate the sentiment analysis on stock market tweets.

3.2 Part 2: Labeling new datasets

3.2.1 Labeling using Amazon Mechanical Turk

The data was submitted to Amazon Mechanical Turk, was asked to be labeled by 4 different workers. Snow et al. [56] suggested that 4 workers is enough number to make

Table (3.5) Summary of tweets labeled by Amazon Mechanical Turk. Most of the tweets were labeled as Neutral, but has been removed from the dataset as we are predicting in binary values. The positive labels are four times more than the negative tweets.

Range	Label assigned to tweets	Count
[-2, -0.5]	Negative	2082
[-0.5, 0.5]	Neutral	9008
[0.5, 2]	Positive	8386

sure that enough people have submitted their opinion on each tweet and so the results would be reliable. We assigned only AMT masters as our workers, meaning they have the highest performance in performing wide range of HITs (Human Intelligence Tasks). We also asked the workers to assign sentiments based on the question: "Is the tweet beneficial to the stock mentioned in tweet or not?". It was important that tweet is not labeled based on perspective of how beneficial it would be for the investor; rather how beneficial it would be to the company itself. Each worker assigned numbers from -2 (very negative) to +2 (very positive) to each tweet. The inter-rater percentage agreement between sentiments assigned to each tweets by the four different workers had the lowest value of 81.9 and highest of 84.5. We considered labels 'very positive' and 'positive' as positive when calculating the inter-agreement percentage.

At the end, the average of the four sentiment was assigned to each tweet as the final sentiment. Out of 20013 tweet records submitted to AMT, we assigned neutral sentiment to a tweet if it had average score between [-0.5, +0.5]. We picked the sentiment positive/negative if at least half of workers labeled them positive/negative. Table 3.5 is a summary of the number of tweets in each category of sentiment.

One downside of this dataset was that the number of positive and negative tweets are not balanced. In order to overcome this issue, we tried many things. At the end balancing the train set by oversampling our negative tweets led to the best result. We also have tried under-sampling positive train set, but it performed worse in accuracy.

3.3 Method and Models

3.3.1 Preprocessing

Twitter messages due to its nature of being informal text, requires a thorough preprocessing step in order to improve classifier's prediction. Twitter messages generally contain a lot of misspelled words, grammatical errors, words that does not exist, or has been written in a non-conventional way. Therefore, in our preprocessing step, we attempted to address all these issues in order to retrieve the most information possible from each tweet.

3.3.1.1 Text substitution

We applied two different text substitution. In our first attempt, we substitute every word that contains both number and an alphabet with <alphanum> tag, and all the numbers with the tag <num>. For instance, '12:30' would be replace with <num>:<num>, 'ftse100' will be replace by <alphanum>, and '500' with <num>.

This way, all hours, measures will be treated the same way, hoping to reduce the amount of non-frequent words in our vocabulary. For example, every time will be replaced by <num>:<num>, and every price will be replaced by $<num>.

3.3.1.2 Spelling correction

In order to address the issue of misspelled words and try to retrieve as many words possible so that it can be recognizable by Word2Vec. [1] For example, we removed '-' or '.' in every word and checked if now they will be recognizable by Word2Vec. Similar attempts was applied on these types of words:

- Remove 'ś'
- Change word in 'Word1-word2' format to 'word1 word2'

[1] We applied Google's Word2Vec pre-trained model with 300 dimensions to get word embeddings from each word.

- Delete consecutive duplicate letters.
- Delete '-' or '.' between every letter of word

3.3.2 Word Embeddings

Word embeddings has been the most effective and popular feature in Natural Language Processing. The two most popular word embedding are GloVe [57] and Google's Word2Vec [58]. We used 300-dimensional pre-trained Word2Vec vectors whenever we could find a word available and otherwise we assigned random initializations. From roughly 10,000 tokens in our vocabulary, around 600 of them was randomly initialized. It was essential for us to use pre-trained embeddings since we used to create a vocabulary in order to see if a particular word exists or not.

As future work, it would be interesting to train a new embedding for stock market context and see if that would increase the accuracy of our model.

3.3.3 Baseline Model

We used Amazon Mechanical Turk to manually label our stock market tweets. In order to create a baseline for our analysis, we applied the same preprocessing technique explained before, and the same machine learning classification method and feature set we designed for [6] on the current dataset. We modified Loughran's lexicon of positive and negative words [7] to be suited for stock market context and used it to calculate number of positive or negative words in each tweet as feature. For example, 'sell' has a negative sentiment in stock market context, that has been added to Loughran's lexicon. We ultimately added around 120 new words to his list. Also, we replaced couple of words that come together in a tweet, but has different sentiment in stock market context with one word, to be able to retrieve its actual sentiment. For example, 'Go down' and 'Pull back' both contain negative sentiment in stock's perceptive. Around 90 word-couples was defined specifically for this context. 3.6 shows the baseline for different machine learning classifiers.

Table (3.6) Baseline accuracy for 11,000 tweet dataset. The best accuracy was when using the SVM with TF-IDF, and only the pos/neg count as feature. Adding the word-couple as a feature, improved the accuracy in Random Forest model, and slightly decreased the accuracy in the SVM.

Classifier	Feature Set	Accuracy
Random Forest	[TF-IDF]	78.6%
Random Forest	[TF-IDF, pos/neg count]	78.9%
Random Forest	[TF-IDF, pos/neg count, Wrod-couple]	79.4%
SVM	[TF-IDF]	77.9%
SVM	[TF-IDF, pos/neg count]	**79.9%**
SVM	[TF-IDF, pos/neg count, Word-couple]	79.5%

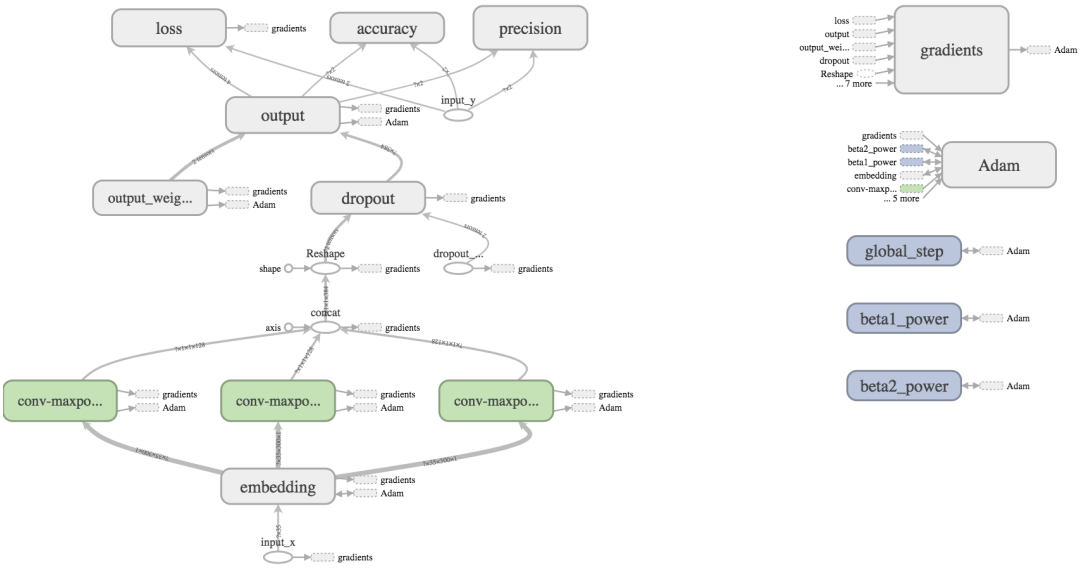


Figure (3.1) Architecture of our CNN model, produced by Tensorboard.

3.4 Neural Network Models

3.4.1 Convolutional neural networks

Convolutional Neural networks (CNNs) have been shown to be useful in variety of applications specially in image processing. Although they have been designed originally for image processing and classification, they found their way into natural language processing and models created using CNNs led to state of the art result in text classification [59, 60] and specifically in classifying tweets [42, 47]. The network

design for CNN can be seen in Figure 3.1. Our CNN model[2], contains an input layer, in which after pre-processing we will reshape each tweet to a matrix. Then we will have Convolutional layer, with specific filters, and finally a max-pooling layer. Specification of each layer is described as follows:

3.4.1.1 Input layer

One problem in using CNNs for tweet classification is the difference in size (i.e. number of words) in tweets. CNNs originally were introduced for image classification, and by design have a fixed size input layer. To overcome this problem, we chose to make all tweets the same size by adding padding to shorter tweets and cutting off the longer ones to make all our tweets the same length. We set the length of out tweets to 35, and among all the tweets in our data, we had only 63 tweets that had to be shortened. This way we could represent each tweet in our dataset by a 35 x 300 dimensional matrix; 35 being the number of terms in each tweets, and 300 is the dimension of the representative vector in our pre-trained embeddings described in section 3.2.

3.4.1.2 Convolutional layer

Having our input matrix, the Convolutional layer, consisting of multiple sliding window functions, will move through the whole matrix embedding vector (word), and these convolutions slide through the matrix to generate an output each time. For example, a filter of length 5 would go through all 35 embedding vectors (words), 5 rows at a time for 30 steps, generating 31 outputs. In our experiment we used convolutions covering three, four and five words at a time, and the output is passed to a ReLU activation function.

[2]Our model, was built and modified based on a Convolutional network available at https://github.com/bernhard2202/twitter-sentiment-analysis.

3.4.1.3 Max-pooling and soft-max

Then we create a 384 dimensional vector with max-pooling on the outputs of our convolutions for each tweet (in example above each convolution creates 31 outputs for each tweet, we select the maximum and disregard all others, so we get one output for each of 384 convolutions). This output vector then will be passed to a soft-max layer to generate a normalized probability score for classification.

3.4.1.4 Training and regularization

Stochastic optimization on cross-entropy-loss was used to train the CNN using Adam optimizer [61]. The data was divided 90% to 10% as train and development sets. After every 1000 training step the performance of the CNN on development data was evaluated and the training was stopped after eight epochs (i.e. 70k training steps) with learning rate of 1e-4. We used this learning rate, because it is low enough to make the neural network more reliable. Although, this will make the optimization process slow, it was not our concern because of our relatively small dataset. A dropout layer for convolutions was used to avoid over-fitting during training. This layer disables each neuron with the probability of 0.5, resulting in a network which uses on average half the neurons in the network in each training step.

3.4.2 Recurrent Neural Networks

Recurrent neural networks, has been shown to be a powerful tool in many NLP tasks such as sentiment analysis [62], machine translation [63], and speech recognition [64]. In RNNs the input will be fed to the network sequentially as opposed to CNNs, in which you have to feed the same size input into the network simultaneously. This makes RNNs a preferred candidate for sequential data with various size inputs, like text. They are constructed with inter-unit connections which creates a directed graph, and their internal state can be considered as a memory which keeps track of previous states.

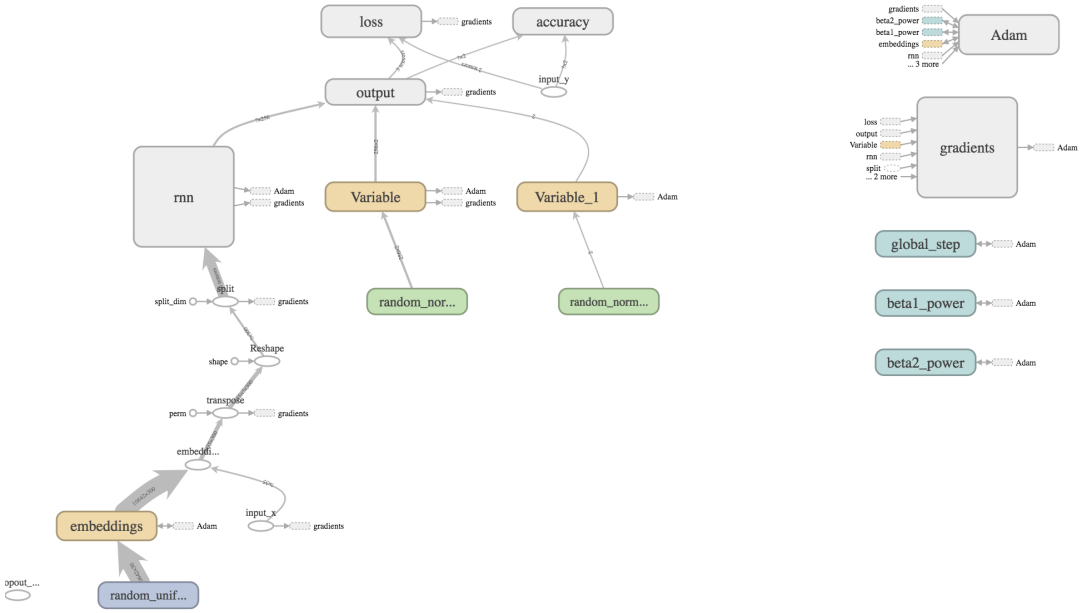


Figure (3.2) Architecture of our LSTM model, produced by Tensorboard.

An issue that arises from this design is that it cannot handle long-term dependencies reliably during back propagation, resulting in vanishing or exploding gradients. This happens because the error should propagate over a long distance in the network. Long Short-Term Memory (LSTM) tries to overcome this issue by adding an explicit memory component to the network's architecture that prevents the gradients to decay very fast, and clipping large gradients will prevent the exploding radiant problem. This is why we decided to try a LSTM network.

In this task, we used a network consisting an embedding layer, one layer of 128 LSTM units and a softmax layer to normalize the output. We also tried variations of this architecture once with 256 LSTM cells, and once with two layer of 128 LSTM cells. You can see the performances for each of these architectures (along with other models) in tables 4.2 and 3.8. The network design for LSTM model can be seen in Figure 3.2.

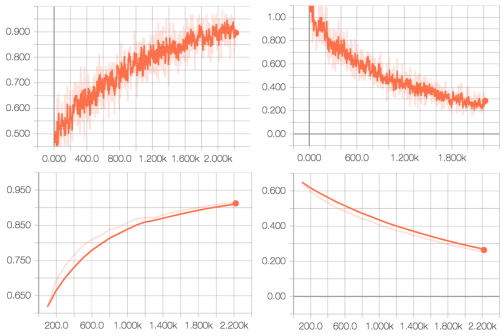


Figure (3.3) Plots of accuracy and loss for each step in train and test set for best loss in CNN, from Tensorboard. Top-left is the accuracy and top-right is the loss for train set. Bottom-left shows the accuracy and bottom-right shows the loss for each run in test set.

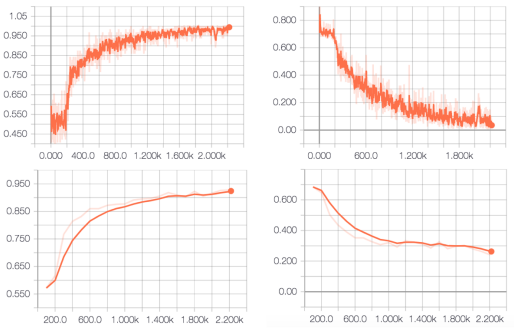


Figure (3.4) Plots of accuracy and loss for each step in train and test set for best accuracy in LSTM, from Tensorboard. Top-left is the accuracy and top-right is the loss for train set. Bottom-left shows the accuracy and bottom-right shows the loss for each run in test set.

3.5 Results

As explained in pre-processing, additional challenge of our dataset was the unbalanced nature of sentiments. In one attempt, we used an unbalanced test set as well as unbalanced train dataset. The result really jumped in accuracy when we used balanced train and test dataset. We re-sampled the negative tweets to create the same number of negative tweets as the positive ones. By doing that, our test set accuracy increased by 8% in CNN and 10% with LSTM.

Changes in preprocessing, improved our accuracy drastically. We tried out two different preprocessing alterations. First attempt was examining the effect of removing or keeping '#' and '$' in the dataset. In all of our runs, we let these two characters to remain in our dataset. With the idea, that each hashtag would differentiate the word with or without these character and result in better capturing the context. But ultimately, removing them increased the accuracy. We believe due to the fact that our vocabulary was relatively small (at most 10643 words), removing these characters helped with eliminating non-frequent words and reducing number of features. The

effect of removing these characters can be seen in the lowest loss of 0.25 in our CNN model. Figure 3.3 shows the accuracy, and loss for this model for both train and test set in each step.

Second, we replaced all of our tags that has been explained in section 3.1 with just one tag <num> with the same justification for removing characters. But, for both LSTM, and CNN we had slight decrease in accuracy and increase in loss.

LSTM in general, trained faster than CNN, and the best accuracy was achieved when we used higher number of LSTM cells (256) with only one layer. Highest accuracy was 92.7% in this model, which was a significant jump from baseline. We removed both '#' and '$' from our dataset, for this model. The 2-layer LSTM did not perform well in accuracy and loss. Because this increase in the complexity of model, would require more data for training. Figure 3.4 shows the accuracy, and loss for this model.

3.6 Chapter summary

In the first part of this chapter, we first analyzed a small dataset that was introduced by SemEval 2017 challenge, Task 5 [6]. We used two different features that was specific to the stock market in our feature set, and using a Random Forest model, and achieved the 7th place in the competition. The dataset that was introduced by SemEval, could not be used for our causality analysis for two different reasons. First, the dataset was only about 1800 tweets. And secon, the dataset was not a time-series of tweets. Therefore, this model with minor changes, has been used as our baseline model in part two of this chapter.

In the second part of this chapter, we first introduced a Twitter dataset that has been labeled by positive or negative sentiments using Amazon Mechanical Turk. Then, we thoroughly compared various deep learning models, and finally introduced our LSTM model with 256 cells, which outperformed all the other models, with accuracy of 92.7%. The most recent and to the best of our knowledge, the only deep

learning paper we have seen in sentiment analysis of financial Tweets by Sohangir et al., [8] used CNN and LSTM models, and they achieved the highest accuracy of 90.8%.

While this model was the best accuracy achieved in sentiment analysis of stock market tweets, there are still places for improvement in this area. We suggest some other steps to be added to the pre-processing analysis. For example, it would be interesting to analyze the hashtang-ed words and figure out if they are a real indicator of a subject or not, using the frequency of hashtag being mentioned in dataset. If not, they can be separated and considered as words. Also, having more tweet dataset, will help us to try out other types of deep learning models, specifically, by trying out a deeper networks. Another attempt in this area could be to create word embeddings, specifically in context of finance.

Table (3.7) Result of various LSTM and CNN Accuracy. The LSTM model with 256 cells outperformed all the other models with accuracy of 92.7% in accuracy.

NN	Specification	Train	Test
CNN	Unbalanced Train/Test	91.5%	80.6%
CNN	Balanced Train/Test	89.7%	88.7%
CNN	Remove '#' and '$'	89.7%	91.6%
CNN	Unique Tag	95.9%	90.4%
LSTM	Unbalanced Train/Test	98.3%	81.6%
LSTM	Balanced Train/Test	97.9%	91.6%
LSTM	Remove '#' and '$'	91.8%	91.8%
LSTM	Unique Tag	98.4%	91.1%
LSTM	2 layer + 128 cell	83.6%	86.6%
LSTM	1 layer + 256 cell	98.4%	**92.7%**

Table (3.8) Result of various LSTM and CNN Loss. Our CNN model when '#' and '$' has been removed in the pre-processing step showed the least error rate. The LSTM model with 256 cells had a very close error rate to the CNN model.

NN	Specification	Train	Test
CNN	Unbalanced Train/Test	0.25	0.40
CNN	Balanced Train/Test	0.26	0.30
CNN	Remove '#' and '$'	0.31	**0.253**
CNN	Unique Tag	0.20	0.27
LSTM	Unbalanced Train/Test	0.07	0.68
LSTM	Balanced Train/Test	012	0.31
LSTM	Remove '#' and '$'	0.28	0.27
LSTM	Unique Tag	0.03	0.34
LSTM	2 layer + 128 cell	0.39	0.31
LSTM	1 layer + 256 cell	0.04%	0.259

CHAPTER 4: MODELS OF CAUSALITY

In Chapter 3, we used Amazon Mechanical Turk to manually label the sentiments for a dataset of 11,000 stock market tweets. Using this labeled dataset, we then thoroughly compared various neural network models against different baselines. Our deep learning method achieved the highest accuracy of 92.7%, compared to the baseline accuracy of 79.9% using an SVM. This is a substantial improvement of the state-of-the-art for sentiment analysis of stock market tweets. Next, we used this deep learning classifier to assign positive and negative sentiments for three years of stock market tweets from 2015 to 2017. Finally, by summing across different time intervals, we calculated the sentiments associated with each tweet at fifteen and thirty minutes, one and three hours, and at one day. In this chapter, we used this aggregated dataset to investigate the causal link between Twitter sentiments and stock market returns.

In our initial causality analysis, our dataset was only for three months of data, on one day intervals. We used that dataset with a wide range of stock companies. In the process, we captured the strong evidence of relationship between many stock returns and the sentiments in both directions. We also found out that most of the stocks that showed any causality were in technology section, where there is more appearance of tweets on Twitter. This evidence of tweets is explained in part 4.5. In the next step, we will identify the causal link in various intervals, in three years of data, for APPLE, FACEBOOK, and AMAZON. We calculate the sentiment and stock return on 15min, 30min, 1hour, 3hours, and 1day intervals. Then Granger causality, and a Bayesian causality will be applied on the dataset.

4.1 Stock market returns

To begin, we downloaded the closing prices for the 100 stock ticker symbols mentioned in our labeled dataset of tweets.[1] Then, we calculated the relative daily return for each company, which is an asset's return relative to a benchmark per day. This is the preferred measure of performance for an active portfolio [2] because it is normalized and because it a stationary time-series, a feature that is essential for most time-series analysis (and specifically, Granger causality). Stationary time-series means that they have a time-invariant mean and variance

We used the following formula to calculate relative stock return:

$$\begin{aligned} Stockreturn &= \tfrac{(p_1 - p_0)}{p_0} \\ p_0 &= Initialstockprice \\ p_1 &= Endingstockprice \end{aligned} \tag{4.1}$$

4.2 Granger Causality Models

Granger causality (GC) is a probabilistic theory of causality[65] that determines if the information in one variable can explain another. According to Suppes [66], an event A causes an event B if two conditions hold: (1) the conditional probability of B given A is greater than the probability of B alone, and (2) A occurs before B. This is a common approach in econometrics, which Clive Granger expanded on [67].

In Granger causality, a variable A causes B if the probability of B, conditional on its own history and that of A, does not equal the probability of B conditional on its history alone. The advantage of this model is that it is both operational and easy

[1] Of the 100 companies mentioned, we replaced the stock symbols of companies that were owned by another with the symbol of the parent company. Specifically, we replaced $LNKD (LinkdIn) with $MSFT (Microsoft) and replaced $SCTY (Solar City) with $TSLA (Tesla). We also excluded the following companies from the list of 100 companies: VXX, GLD, SPY, GDX , SPX , WFM , EMC, APP, BRCM, and GMCR. These companies were either not currently trading, their trading data could not be found, or they were a specific index (e.g., S&P 500.

[2] https://www.investopedia.com

to implement, but it is criticized for not actually being a model of causality (rather, it's a model of increased predictability). Critics have pointed out that even when A has been shown to Granger cause B, it does not necessarily follow that controlling A will directly influence B. Further, nor does it tell us the magnitude of the effect on B. Granger Causality is primarily used for causal notions of policy control, explanation and understanding of time-series, and in some cases, for prediction.

Correlation is not causation It is important to understand that correlation is different than causation. When two variables A and B (e.g., time series) are correlated, this means that there is a statistical association (or dependence) between them. However, it does not necessarily mean that the relationship is causal (i.e., A causes B, or B causes A). Conversely, if there is a causal relationship between two variables, it does not necessarily follow that they are correlated. Further, correlation is a symmetric relationship (i.e., a measure of statistical linear dependence) while causation if asymmetric. For example, the activity of a windmill is correlated with wind velocity - the faster the wind, the greater the rotation of the windmill's blades. Someone might conclude that the rotation of the windmill's blades causes the wind, therefore wind is caused by the rotation of windmills. In this example, the speed that correlation (simultaneity) between windmill activity and wind velocity does not imply that wind is caused by windmills [3].

Formal Definition of Granger Causality: A time-series Y can be written as an autoregressive process [4], which means that the past values of Y can , in part, explain

[3] `https://en.wikipedia.org/wiki/Correlation_does_not_imply_causation`

[4] An autoregressive (AR) model is a representation of a type of random process; as such, it is used to describe certain time-varying processes in nature, economics, etc. The autoregressive model specifies that the output variable depends linearly on its own previous values and on a stochastic term (an imperfectly predictable term); thus the model is in the form of a stochastic difference equation. `http://paulbourke.net/miscellaneous/ar/`

the current value of Y. Formally, an autoregressive model is defined as follows:

$$Y_t = \alpha + \sum_{i=1}^{k} \beta_j Y_{t-i} + \epsilon_t. \tag{4.2}$$

To define his version of causality, Granger introduced another variable X to the autoregressive model, which also has past values like Y.

$$Y_t = \alpha + \sum_{i=1}^{k} \beta_j Y_{t-i} + \sum_{j}^{k} \lambda_j X_{t-j} + \epsilon_t. \tag{4.3}$$

If adding X improves the prediction of current values of Y, when compared to the predictions from the autoregressive model alone, then X is said to "Granger cause" Y . Technically, Granger causality is an F-test, where the null hypothesis is that all of the λ are equal to zero for all j. Note that you can also test the reverse case; that is, test whether Y "Granger causes" X. Both causal directions, or none, are possible. Tests for Granger causality should only be performed on stationary variables, which means that they have a time-invariant mean and variance. Specifically, this means that the variables must be I(0) [5] and that they can be adequately represented by a linear AR(p) process [6] .

4.2.1 Three month comparison of social media sentiment analysis and stock market returns

Before using Granger causality, we first use KPSS [7] to test if a time-series is stationary. The null hypothesis for this test is that the data is stationary. We applied

[5]In statistics, the order of integration, denoted I(d), of a time series is a summary statistic, which reports the minimum number of differences required to obtain a covariance-stationary series. `https://en.wikipedia.org/wiki/Order_of_integration`

[6]The autocorrelation function of an AR(p) process is a sum of decaying exponentials.The simplest AR process is AR(0), which has no dependence between the terms. Only the error/innovation/noise term contributes to the output of the process, so in the figure, AR(0) corresponds to white noise. `http://paulbourke.net/miscellaneous/ar/`

[7]Kwiatkowski-Phillips-Schmidt-Shin: http://debis.deu.edu.tr/userweb /onder.hanedar/dosyalar/kpss.pdf

this test on our time-series of stock market returns, and also on our sentiment time-series at different intervals. If the p-value was greater than 0.05, the test did not reject the null hypothesis that the data was stationary. When the null hypothesis was rejected (i.e., the dataset was non-stationary), we determined an appropriate lag that would make the dataset stationary. After applying all of the appropriate lags to make the datasets stationary, we used Granger causality determine if there was a causal relationship between our stock market returns and the sentiments manually labeled by Amazon Mechanical Turk (AMT), and then if there was one between the returns and the sentiments predicted by the baseline SVM classifier (bSVM) we built in part 3.3. To test for causality in both directions, we built two models.

$$\textbf{Model (1):} \quad RV \sim Lags(RV, LAG) + Lags(SSC, LAG) \tag{4.4}$$

$$\textbf{Model (2):} \quad SSC \sim Lags(SSC, LAG) + Lags(RV, LAG) \tag{4.5}$$

Model one determines if sentiment scores have a causal effect on stock return values, while model two determines if sentiment scores affect stock return values. In both models, the lag (LAG) is the number of days the cause precedes the effect, the return value (RV) is the calculated daily return for 83 different stocks, and the sentiment scores (SSC) are from either (1) the sentiments manually labeled by Amazon Mechanical Turk or (2) assigned by our baseline SVM classifier. We ran the models twice, one for each sentiment dataset, and tested lags between one and ten. Running these models answered two questions: (1) is there a causal relationship between sentiment scores in tweets and stock return values (in either direction, or both), and (2) how far in the past was the cause (i.e., the lag)? The P- and F-values of all statistically significant permutations of the Granger causality models are in Appendix B.

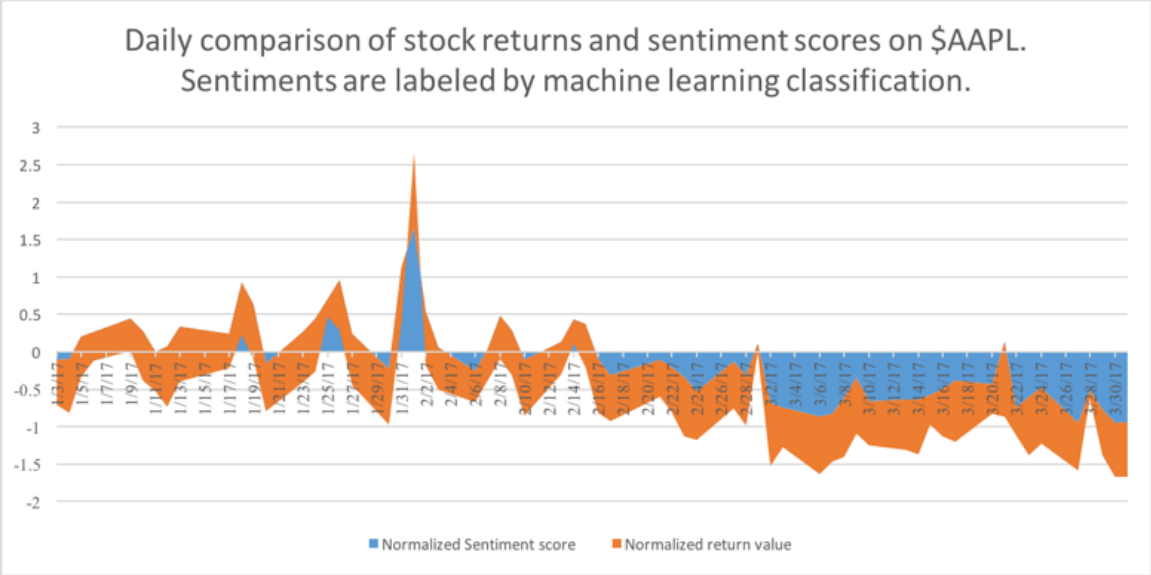


Figure (4.1) Daily comparison of stock returns and sentiment scores on $APPL. Sentiments are labeled by AMT. This shows that there is a general trend between stock return and the sentiments labeled by AMT.

Figure 4.1 shows the relationship between the daily sentiments assigned by AMT and the return values for $AAPL. Similarly, Figure 4.2 shows the relationship between the daily sentiments assigned by bSVM and the return values for $AAPL. **[Is this a general trend across all stocks? What is the correlation?]** Taken together, these two figures strongly suggest that there is a correlation between stock return values and sentiment scores, and that the sentiments assigned by AMT appear to be better correlated with return values that the sentiments assigned our bSVM. The most likely cause of this poorer correlation is high error rate of the bSVM, and is a strong motivator to instead use sentiments assigned by LSTM classifier that we developed, with its 20% in error rate. We expect that by improving the accuracy of sentiments assigned to tweets, the causal analysis will likewise improve.

For nineteen (235) stocks, Figure 4.3 shows how many days before a return value (the lag) that Granger causality detected a significant causal effect from sentiment scores (Model 1; $SC \rightarrow RV$). For each stock, we show two bars: one for the AMT labeled tweets (blue) and one for the bSVM labeled tweets (orange). If a bar is

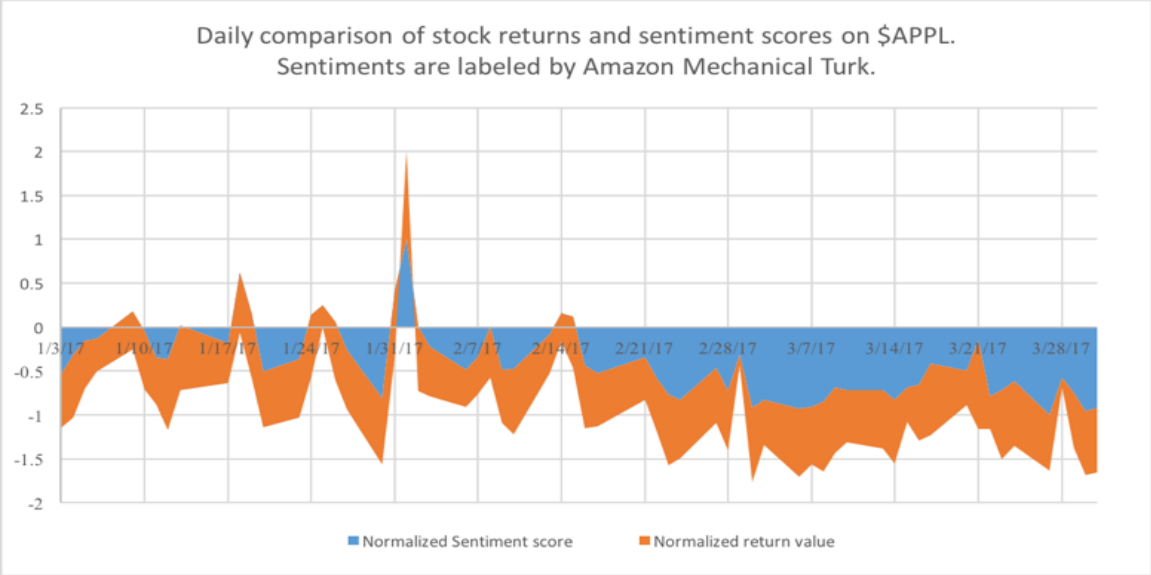


Figure (4.2) Daily comparison of stock returns and sentiment scores on $APPL. Sentiments are predicted by ML model. This shows that there is a general trend between stock return and the sentiments labeled by our machine learning model. Although the trend is not as obvious as the one with AMT, but it still exists. This is a visual representation of that 20% error rate is damaging the trend to some extend.

missing, it means that there was no significant causal relationship for any lag tested. Figure 4.4 is identical to Figure 4.3, except that it shows the reverse causal direction (Model 2; $RV \rightarrow SC$). For the remaining 64 stocks, we also see statistically significant causal relationships between returns and sentiment scores, but we chose not to show them because there is less consistency between the two sentiment datasets (which is expected, given the low accuracy of the bSVM labels).

4.2.2 Three year comparison of social media sentiment analysis and stock market returns

In the previous causality analysis, we used three months of daily stock return values from a variety of companies and showed that there is a causal relationship between many of their stock return values and the sentiments associated with tweets. This occurred in both directions (SC $\rightarrow$ RV and $RV \rightarrow SC$) and at different lags, depending on the stock. Further, we found that most of the stocks involved in a causal

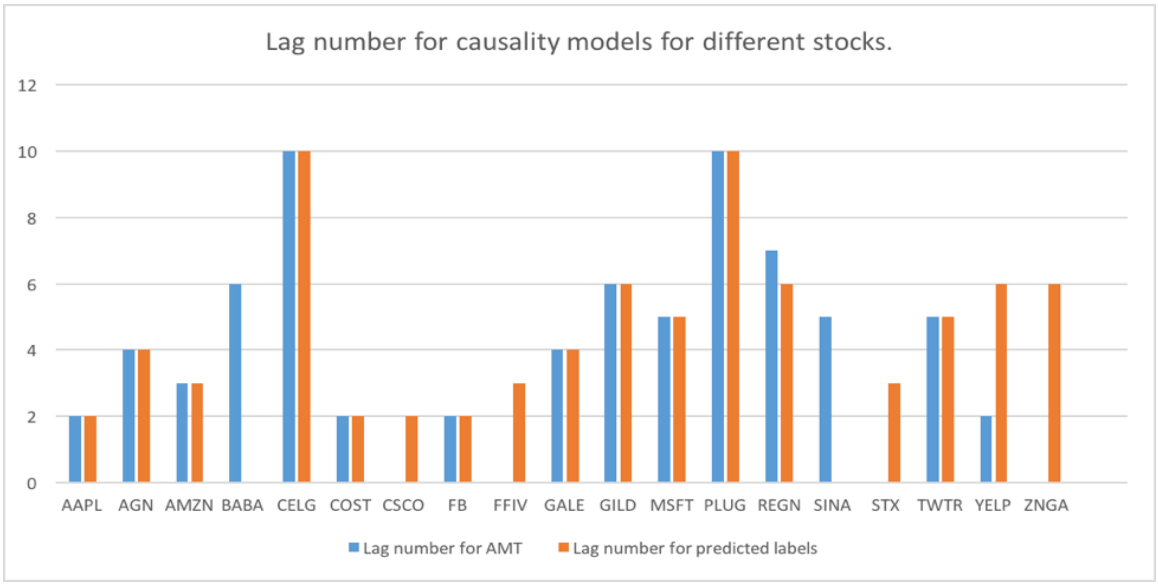


Figure (4.3) Lag number for GC for various stocks in model two. Lag is the number of days before current day that sentiment score had causal effect on stock market return.

relationship were from the technology sector, where there is a greater frequency of tweets than is seen with non-technological stocks (explained further in part 4.5).

In this section, we performed an in-depth causal analysis for the three stocks most commonly referred to in social media – Apple, Facebook, and Amazon – over a period of three years from 2015 - 2017. We used our LSTM model 3.5 to assign sentiments to an expanded Twitter dataset, which had 386,251 tweets and covered the same three year period as the stock return values. We then applied the two GC models described in 4.2.1 to find causal relationships between the sentiments and return values at five different intervals: fifteen and thirty minutes, one and three hours, and one day. For a particular interval, all of the sentiments in that interval were summed to get an aggregate score. We found causal relationship between tweet sentiments and return values for Amazon and Facebook (in both directions) at fifteen minutes, three hours, and one day. No causal relationships were found for Apple.

Looking more closely at the results of the causality analysis, we see in Figures 4.5 and 4.6 that before three hours, the value of the lag fluctuates, but at three hours

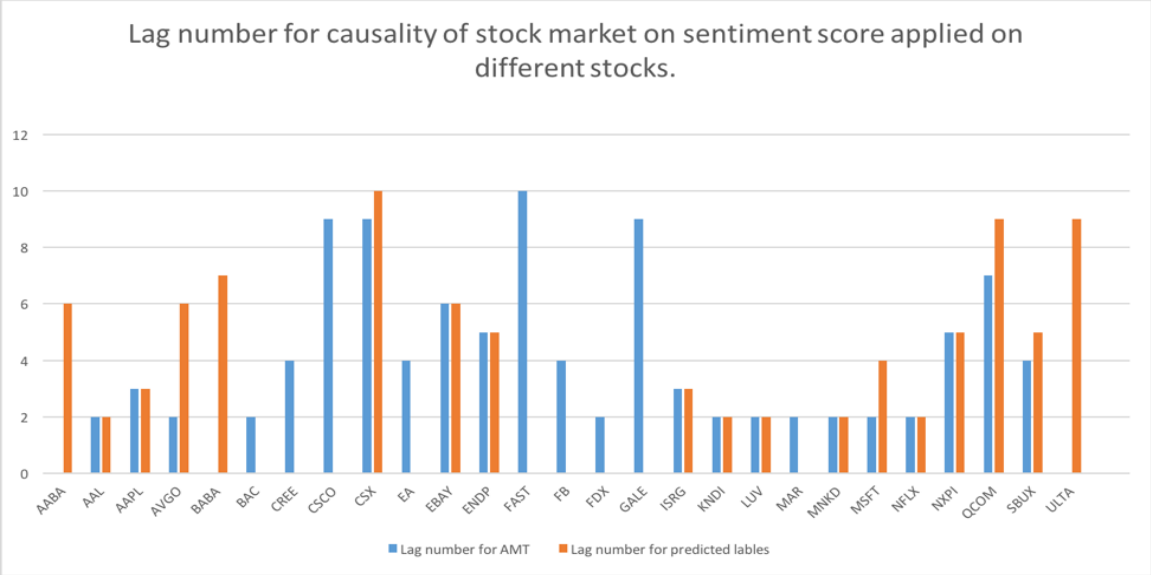
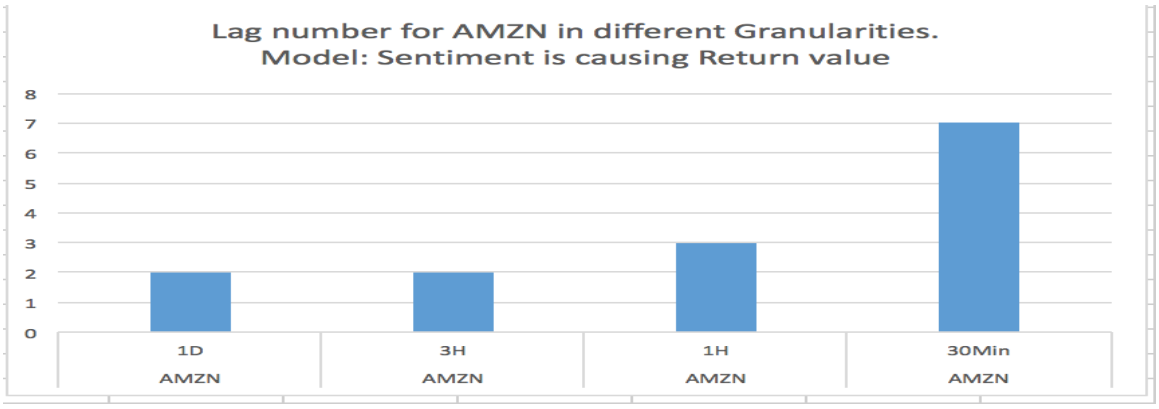


Figure (4.4) Lag number for GC for various stocks in model one. Lag is the number of days before current day that stock market return had causal effect on sentiment scores.

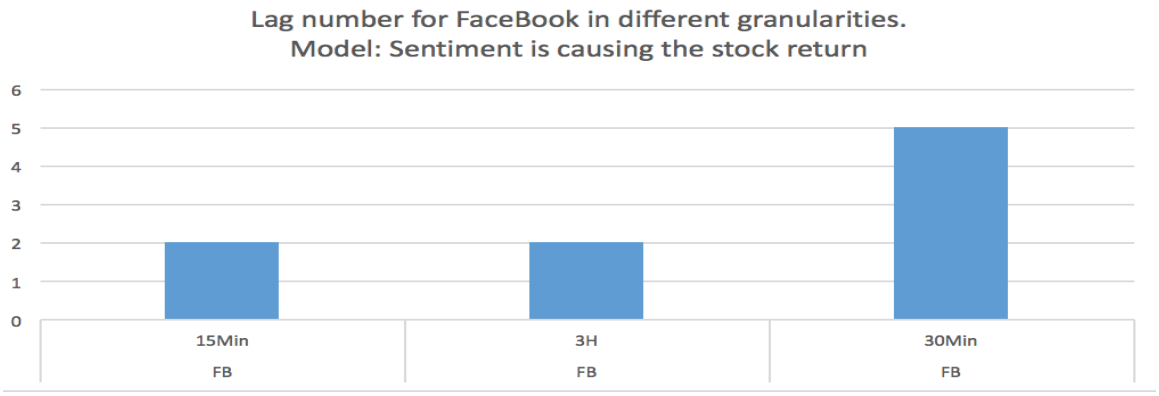
and one day, it stabilizes at a lag of two. We also calculated the causality weight as suggested by Geweke [68], who proved that the linear dependence of a causal model (i.e., the causality weight) can be captured by the F-measure. For both Amazon and Facebook, we found the greatest causality weight at three hours (Figures 4.7 and 4.8). This result, along with the stabilization of the lag at three hours, suggests that we should select an interval of three hours for further analysis.

4.3 Bayesian Causality Networks

The probabilistic models based on directed acyclic graphs (DAGs) was first introduced in 1921 by Sewall Wright [69]. These models have long been used in many different fields. In particular, in Artificial Intelligence field it is called Bayesian networks. The nodes in a Bayesian network represent propositional variables of interest and the links represent informational or causal dependencies among the variables. The dependencies are quantified by conditional probabilities for each node given its parents in the network. An example of Bayesian networks was introduced [70] that we believe will describe the model in a simple yet sufficient to describe the Bayesian



(a) Amazon shows significant results on 30MIN, 1HOUR, 3HOUR and 1DAY intervals.

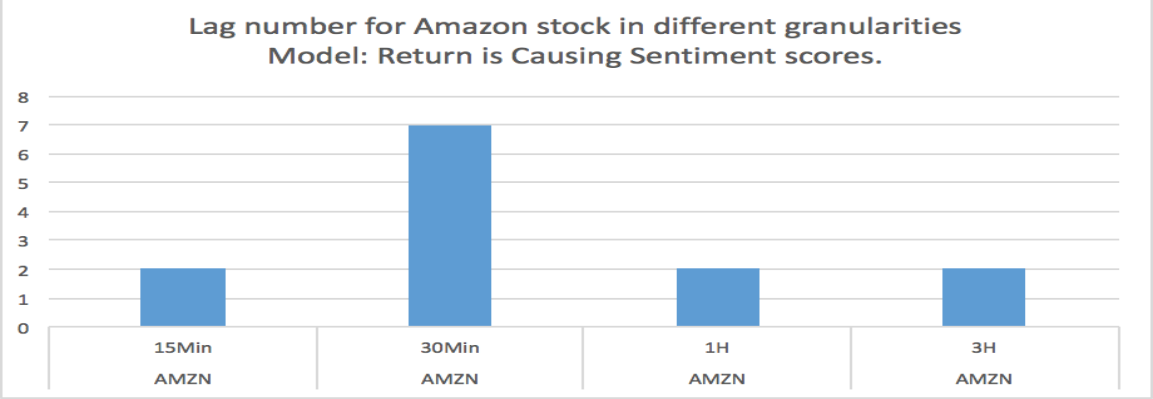


(b) Facebook shows significant results on 15MIN, 3HOUR and 1DAY intervals.

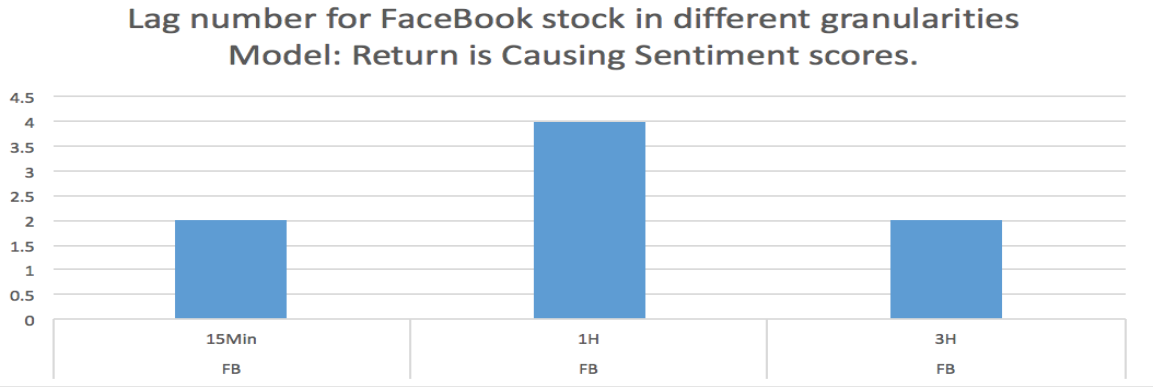
Figure (4.5) Statistically significant Lag numbers for Model 1: Sentiment causes the stock return. In this model, both Amazon and Facebook showed statistically significant causal link in different lags. The common lag between these two stock, was 30Min, and 3Hours lags.

networks:

A simple yet typical Bayesian network is shown in figure 4.9. It describes the causal relationships among the season of the year (X1), whether it's raining (X2), whether the sprinkler is on (X3), whether the pavement is wet (X4), and whether the pavement is slippery (X5). For example, absence of a link between X1 and X5 shows that there is no direct influence of season on slipperiness. Ultimately, the most important aspect of a Bayesian network is that they are direct representations of the knowledge and understanding of world as we know it, not of reasoning processes.



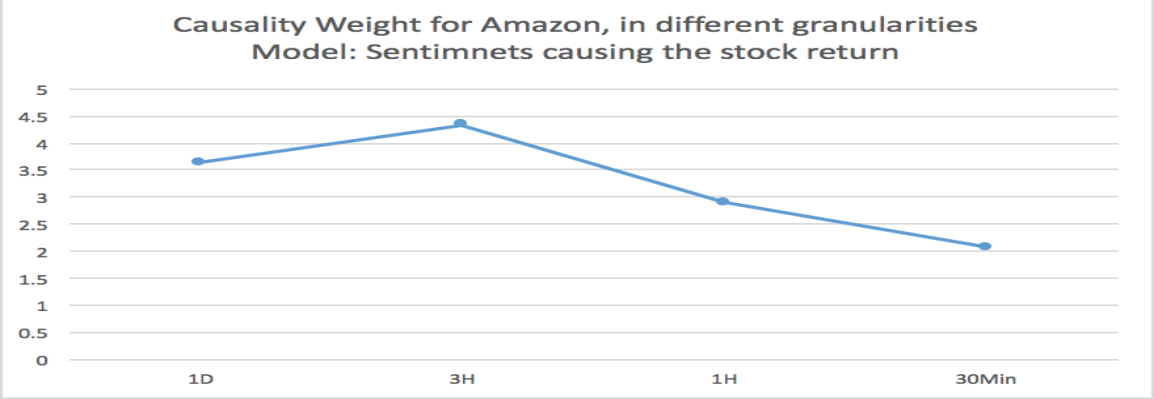
(a) Amazon shows significant results on 15MIN,, 30MIN, 1HOUR and 3HOURs intervals.



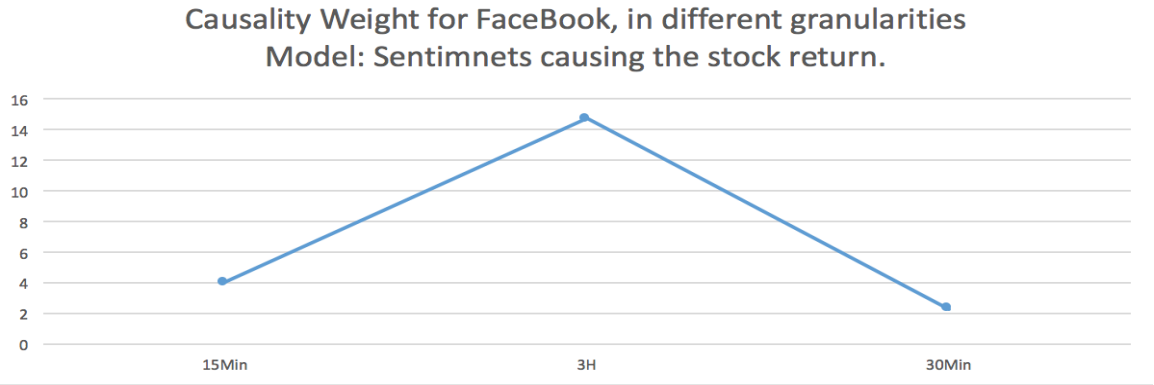
(b) Facebook shows significant results on 15MIN, 1HOUR and 3HOURS intervals.

Figure (4.6) Statistically significant Lag numbers for Model 2: Stock return causes the sentiments. In this model, both Amazon and Facebook showed statistically significant causal link in different lags. The common lag between these two stock, was 15Min, 1Hour, and 3Hours lags.

Probabilistic semantics. A full probabilistic model needs to represent the join distribution of every possible event, considering the values that has been denied by the values of all the variables. Bayesian networks can take into account the joint distribution of all the events. For instance, if X_i denotes the value of the variable X_i and pa_i denotes some set of values for X_i's parents, then $P(X_i|pa_i)$ demonstrates this conditional distribution. In the rain model example, $P(X_4|X_2; X_3)$ is the probability of wetness given the values of sprinkler and rain. The global semantics of Bayesian networks species that the full joint distribution is given by the product is demonstrated like this:



(a) Amazon shows significant causal weight on 30MIN, 1HOUR, 3HOUR and 1DAY intervals.



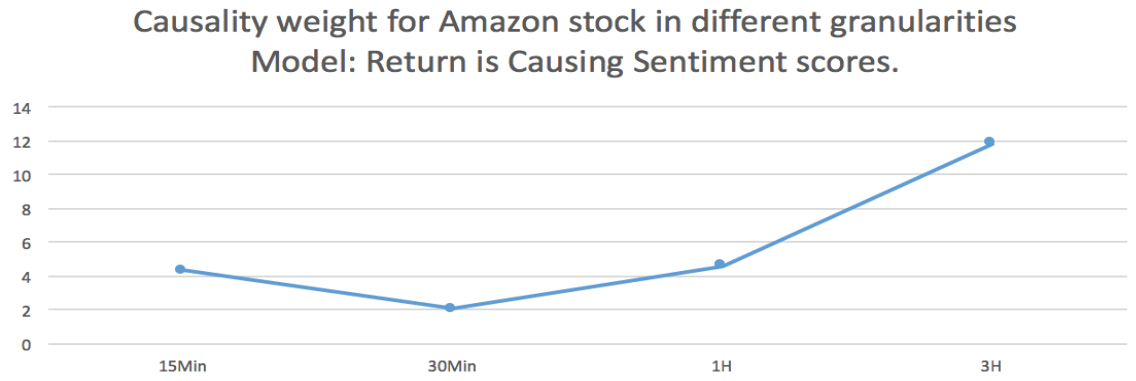
(b) Facebook shows significant causal weight on 15MIN, 3HOUR and 1DAY intervals.

Figure (4.7) Statistically significant Weights for Model 1: Sentiment causes the stock return. For both stocks, the causality weight was strongest at the 3Hour time. The lowest causal weight occurred at 30Min interval.

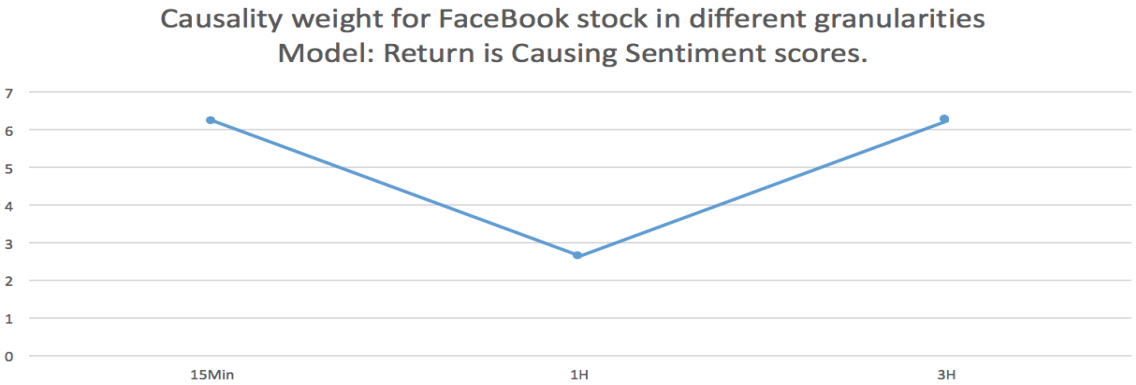
$$P(X_1, ..., X_n) = \Pi_i P(X_i|pa_i). \tag{4.6}$$

For instance, in our example network, we have:

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2; X_3)P(X_5|X_4) \tag{4.7}$$



(a) Amazon shows significant causal weight on 15MIN, 30MIN, 1HOUR, 3HOUR intervals.



(b) Facebook shows significant causal weight on 15MIN, 1HOUR and 3HOUR intervals.

Figure (4.8) Statistically significant Weights for Model 2: Stock return causes the sentiments. For both stocks, the causality weight was strongest at the 3Hour time. The lowest causal weight occurred at 30Min interval for Amazon and 1H for Facebook.

Learning in Bayesian networks. The gradient-based or Expectation-Maximization[8] methods [71, 72] can be used to update these conditional probabilities, very much like the way the weights in neural networks get updates. Then the structure of the network can be learned using network complexity and the degree of fit to the data. [73]

Causal networks. Most probabilistic models, such as Bayesian networks, focus on a distribution over possible observed events. Though, a causal network in this area, is basically a Bayesian network with an added condition on the parent of a node. For

[8] "In statistics, an expectation-maximization (EM) algorithm is an iterative method to find maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables." *https* : *//mathscinet.ams.org/mathscinet* − *getitem*?*mr* = 0501537

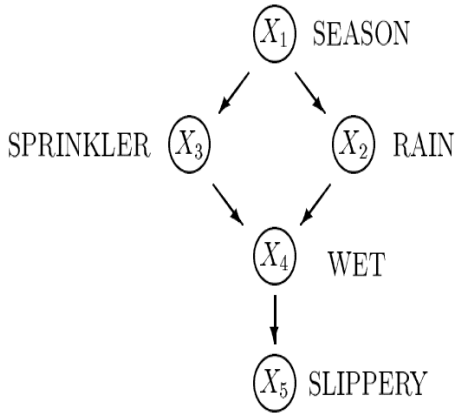


Figure (4.9) Illustration of a simple Bayesian network

instance, what if I turn the sprinkler on? What effect does that have on the season, or on the connection between wetness and slipperiness? In a causal network, the result of an intervention is obvious: the sprinkler node is set to $X_3 = on$ and the causal link between the season X_1 and the sprinkler X_3 is removed. All other causal links and conditional probabilities remain intact, so the new model is:

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_4|X_2; X_3 = on)P(X_5|X_4) \qquad (4.8)$$

4.3.1 Pearl's Bayesian Causality

Leland Neuberg has described Pearl's causality model in his book [74]. He describes that what Pearl mean by probability is the degree of belief. If V is a set of variables, with join probability of $P(V)$, assuming the variables are pre-ordered[9] with the cause-effect relations. Then Pearl defines a minimal set of predecessors of any X_i in V as a set that is independent of all of its other predecessors, the Markov parents, or immediate causes (PA_j) of X_i. That is, if $Q_i(\supseteq PA_j)$ is the set of predecessors of X_i then $P(X_j|Pa_i) = P(X_j|q_i)$ and the equality fails to hold if any proper subset of

[9]Pre-ordered variable is defined when the cause and effect variables have been detected in a dataset. The cause variable must come before the effect variable.

PA_j replaces PA_j. Then, the functional causal model that is based on V, would be $X_i = f_i(pa_j, u_j), i = 1, ..., n$ where the U_j represent errors and the f_i are functions.

Pearl uses mathematical graph theory, to show causal relations. He describes points as nodes of the graph. Lines are the edges which can be directed or an indirected edge. A path in graph theory, is explained a sequence of edges. When a graph is not directed, a path with at least two edges that ends at the node it began create a cycle. In a directed acyclic graph (DAGs) is used in Pearl's model, which uses the nodes of the graphs for the pre-ordering of V. The directed edges in DAGs when they enter a node, shows the immediate cause from the parent node. In summary his model consists of a graph representing a set of variable that are pre-ordered by hypothesized cause-effect relations, and join probability distributions between the variables.

Pearl explains [70], that his model does not include the order of the functional causal model as "a nonlinear, nonparametric generalization of the linear structural equation models (SEMs)" [75]. His model resembles the path analytic diagrams[10] that the Wrights introduced that was introduced in 1920s [75]. Pearl stresses that in "linear models, PA_j corresponds to those variables on the r.h.s. that have nonzero coefficients [70] ".

4.3.1.1 Importance of Pearl's model

Stephen Morgan, in his book of "Counterfactuals and Causal Inference" [75] explained the importance and influence of Pearl's book on causality in 2000 [76] very well. Judea Pearl demonstrates a very robust graphical theory of causality. There are differences between the traditional path models and how he has used the directed acyclic graphs (DAGs) in his models. He has provided an extensive and powerful

[10]Sewall Wright introduced a method of estimating causal path coefficients by decomposing the correlations among a set of variables. He defined a set of rules for creating a path diagram which would allow for this mathematical decomposition. The basic idea is that the correlation of any two variables in a path diagram is the summation of the coefficients that connect the the two variables. The path analytic diagram needs to follow three main rules. First, no loops are allowed. Second, once you traveled a route forward, you cannot travel backwards. And finally, you are only allowed to have one curved arrow from first to the last variable in any route.

framework for thinking about causality.

Pearl has shown that graphs provide a very robust way of thinking about causal systems, and are extremely powerful in detecting the causal strategies and links to estimate the effects withing variables. His model has its limitations, specially compared to the potential outcome framework [11], the generality of these networks are being suppressed. Instead, Pearl has shown that the graphs provide direct powerful way of thinking about causal systems of variables and the identification strategies that can be used to estimate the effects within them. And therefore, Pearl's model is the confirmation for the importance of the graphical models, in spite of some of its limitations.

The importance of Pearl's work can be three different reasons. First, since the framework is nonparametric, it is usually not necessary to identify the functional dependency of outcome Y, on the a variable like X that causes it. For instance, $X->Y$ does not specify that the effect of X on Y is linear, quadratic, or any other highly nonlinear functionality. It simply just identifies that X is causing Y. This is specially helpful, since it does not provide any assumptions about the functional form. This is a very distinguish difference of his model with traditional path models, which has become Achiles' heel for these models [75]. Second, his model shows the importance of Collider variables, which is when the causality is influenced by two or more variables. These variables are special kind of endogenous variables [12] and need to be treated with caution. Finally, Pearl establishes methods of causal effect

[11] "The Neyman potential outcomes framework is based on the idea of potential outcomes and the assignment mechanism: every unit has different potential outcomes depending on their "assignment" to a condition. Potential outcomes are expressed in the form of counterfactual conditional statements of the case conditional on a prior event occurring. For example, a person would have a particular income at age 40 if they had attended a private college, whereas they would have a different income at age 40 had they attended a public college. To measure the causal effect of going to a public versus a private college, the investigator should look at the outcome for the same individual in both alternative futures. Since it is impossible to see both potential outcomes at once, one of the potential outcomes is always missing. This observation is described as the fundamental problem of causal inference." `http://sekhon.berkeley.edu/papers/SekhonOxfordHandbook.pdf`

[12] Endogenous variables have values that are determined by other variables in the system. These variables mainly are used in econometrics and sometimes in linear regression.

in three different ways: conditioning on variables that block all back door paths [13], conditioning for variables that allow for estimation by mechanism, and estimating causal effect by estimating by an instrumental variable that is an exogenous shock [14] to the cause.

4.3.1.2 Stock-Sentiment Bayesian model

The Bayesian graphical model that we investigate is demonstrated in figure . The graph on the left describes the causal relationships among the sentiment scores ($X1$), whether it's affecting the stock market return ($X2$), and in the other direction for graph on the right in figure 4.9. Full probabilistic models need to represent the join distribution of every event. As a result, since we only have one variable that we are trying to model, our simple network is described in equation 4.9. In the next step, we can use re-tweet count of each tweet, as a cause for how each tweet is valuable. The higher the re-tweet count, the higher would be the sentiment score of a tweet. Portfolio return can be added to this model to see if an stock in a portfolio would have causal affect on the portfolio. The figure 4.11 can be the graph of what this model would be like.

$$P(X_1, X_2) = P(X_1)P(X_2|X_1) \tag{4.9}$$

4.3.2 Google's Bayesian causality network for time-series data

This method that was introduced by [77] from Google, generalizes the widely used difference-in differences approach to the time-series setting by explicitly modeling the counterfactual [15] of a time series observed both before and after the intervention. In the paper, they explain the model and the advantages of the model. They describe

[13] A back door path is a non-causal path from node A to node Y . It is a path that would remain if any arrows pointing out of A was removed.

[14] An exogenous shock is an event from outside of the system that affects the system.

[15] A counterfactual conditional, is a conditional containing an if-clause which is contrary to fact. https://philpapers.org/rec/GOOTPO-2

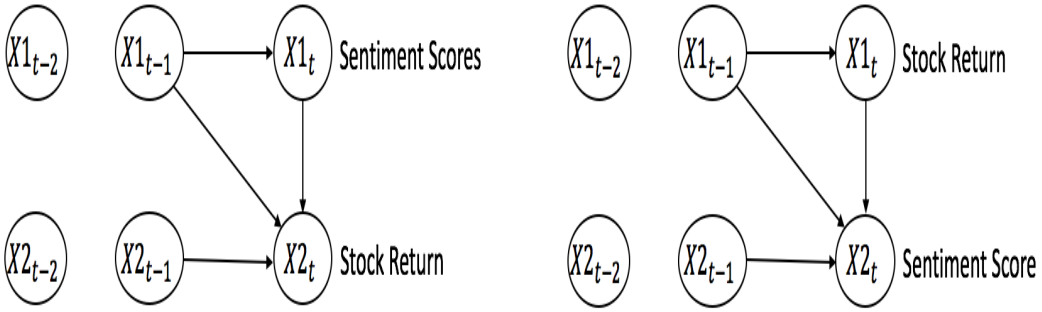


Figure (4.10) Bayesian network illustration of our model. The model on the left explains if the sentiment scores is showing causal affect on the stock market return. The graph on the right expresses the model if the stock return is causing the sentiments expressed in tweets.

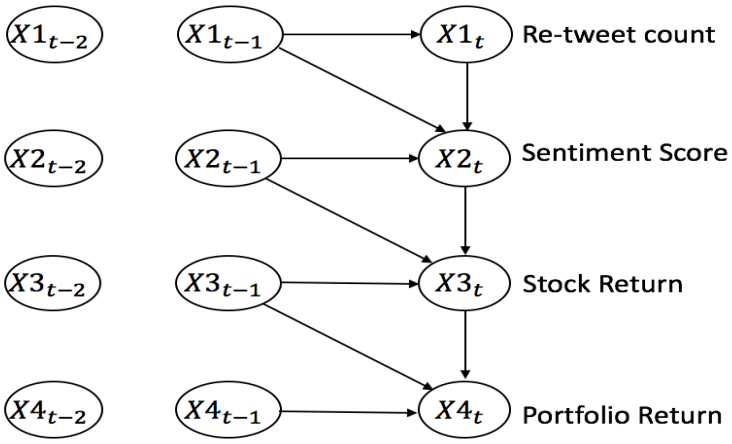


Figure (4.11) Future Bayesian network illustration of our model. For the future models, it would be interesting to understand the effect of re-tweet counts on the sentiments. An stock return also can have an effect on portfolio return.

that the model improves on current methods in two different ways. First, it provides a fully Bayesian time-series estimate for the effect. Second, it uses model averaging to construct the most appropriate synthetic control[16] for modeling the counterfactual.

[16] "The synthetic control method is a statistical method used to evaluate the effect of an intervention in comparative case studies. It involves the construction of a weighted combination of groups used as controls, to which the treatment group is compared. This comparison is used to estimate what would have happened to the treatment group if it had not received the treatment. Unlike difference in differences approaches, this method can account for the effects of confounder changing over time, by weighting the control group to better match the treatment group before the intervention. Another advantage of the synthetic control method is that it allows researchers to systematically select comparison groups." https://onlinelibrary.wiley.com/doi/epdf/10.1002/hec.3258

This approach focuses on measuring the impact of a discrete marketing event, such as the release of a new product, the introduction of a new feature, or the beginning or end of an advertising campaign, with the aim of measuring the event's impact on a response metric of interest (e.g., sales).

This method calculates the causal impact of a treatment as the difference between the observed value of the response and the (unobserved) value that would have been obtained under the alternative treatment. In this setting the response variable is a time series, so the causal effect is the difference between the observed series and the series that would have been observed had the intervention not taken place. In the model if the stock market causes the sentiment scores, the intervention is the sentiment scores, and the observed series is the stock market returns.

This approach has three common characteristics with the state-space paradigm [17]. "First, it allows to flexibly accommodate different kinds of assumptions about the latent state and emission processes underlying the observed data, including local trends and seasonality. Second, it uses a fully Bayesian approach to inferring the temporal evolution of counterfactual activity and incremental impact. One advantage of this, is the flexibility with which posterior inferences can be summarized. Third, it uses a regression component that precludes a rigid commitment to a particular set of controls by integrating out posterior uncertainty about the influence of each predictor as well as our uncertainty about which predictors to include in the first place, which avoids over-fitting." [77]

[17] "State space model (SSM) refers to a class of probabilistic graphical model that describes the probabilistic dependence between the latent state variable and the observed measurement. The state or the measurement can be either continuous or discrete. The term "state space" originated in 1960s in the area of control engineering (Kalman, 1960). SSM provides a general framework for analyzing deterministic and stochastic dynamical systems that are measured or observed through a stochastic process. The SSM framework has been successfully applied in engineering, statistics, computer science and economics to solve a broad range of dynamical systems problems." `https://onlinelibrary.wiley.com/doi/epdf/10.1002/hec.3258`

4.3.3 3 years comparison of social media sentiment analysis and stock market returns using Google's Bayesian Causality approach

For this phase we set the first 100 days as the pre-period, and the rest of three years as our post-period for the daily sentiments and stock market returns. This approach, first models the stock based on the pre-period, then predicts the post-period stock return once with including the sentiments and again without them. Then, this model calculates the effect of including the sentiment in the model. It uses a Bayesian probability approach to understand the probability of that effect being sporadic. On first attempt, it predicts the values in the time-series, and once more uses the sum of post-period and pre-period. At the end, it uses the one-sided P-value of a Bayesian probability to detect if detected effect was significant[18]. The response variable in our models was the stock market return and the intervention was the sentiments. Out of the two models that was created, the Stock causing the Sentiment showed causality in all three stocks. Comparatively, in analysis of causation of sentiments on stock only AMAZON showed causality in one day interval. As it is shown in Table 4.1, all sentiments showed negative effect on the stock market with average weight of **-25.46** and average decrease of **21%** in the stock market return prediction result. This means that by including the sentiment in the prediction of the stock return, on average, stock return was decreased by 21%. Adding the sentiment does not produce noise, but will affect the stock market in a negative way. This model did not show any significant result in other intervals. In the next three parts, we will describe the result of each stock in more details.

[18]The Bayesian interpretation of the one-sided P-value is that it is a test for direction, as the logit of the one-sided P value equals the log of the Bayes factor. From a Bayesian perspective, the one-sided P-value is not a test that involves the null hypothesis at all-instead, it is a test for the direction of an effect. $https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5965556/$

Table (4.1) Result of Bayesian Causality Model. All sentiment scores shows negative effect on the stock market with average weight of **-25.46** and average decrease of **21%**. In the Table, Prob of CE is Probability of Causal Effect, CW is Causation Weight, and PE is Percentage of Effect.

Stock	CW	PE	Bayes Prob.	Prob. of CE
Amazon	-31.76	Decrease of -20%	0.011	98.69%
Facebook	-21.78	Decrease of -21%	0.027	97.3%
Apple	-22.84	Decrease of -22%	0.02	97.9%

4.3.3.1 Amazon

Stock Cause Sentiment. The response variable (Sentiments) had an average value of approximately 82.02. By contrast, in the absence of an intervention (stock return), we would have expected an average response of 113.78. The 95% interval of this counterfactual prediction is [87.46, 140.95]. Subtracting this prediction from the observed response yields an estimate of the causal effect the intervention had on the response variable. This effect is **-31.76** with a 95% interval of [-58.93, -5.43].

Summing up the individual data points during the post-intervention period (which can only sometimes be meaningfully interpreted), the response variable had an overall value of 46.92K (46,920). By contrast, had the intervention not taken place, we would have expected a sum of 65.08K. The 95% interval of this prediction is [50.03K, 80.62K].

The above results are given in terms of absolute numbers. In relative terms, the response variable showed a decrease of-28%. The 95% interval of this percentage is [-52%, -5%].This means that the negative effect observed during the intervention period is statistically significant. The probability of obtaining this effect by chance is very small (Bayesian one-sided[19] tail-area probability p = 0.011). This means the causal effect can be considered statistically significant.

[19]The Bayesian interpretation of the one-sided P-value is that it is a test for direction, as the logit of the one-sided P value equals the log of the Bayes factor. From a Bayesian perspective, the one-sided P-value is not a test that involves the null hypothesis at all-instead, it is a test for the direction of an effect. *https* : *//www.ncbi.nlm.nih.gov/pmc/articles/PMC*5965556/

Sentiment causes the stock. During the post-intervention period, the response variable (stock return) had an average value of approx. 0.0018. By contrast, in the absence of an intervention (Sentiment), we would have expected an average response of -0.034. The 95% interval of this counterfactual prediction is [-0.079, 0.013]. Subtracting this prediction from the observed response yields an estimate of the causal effect the intervention had on the response variable. This effect is 0.035 with a 95% interval of [-0.011, 0.081].

Summing up the individual data points during the post-intervention period (which can only sometimes be meaningfully interpreted), the response variable had an overall value of 1.09. By contrast, had the intervention not taken place, we would have expected a sum of -20.93. The 95% interval of this prediction is [-49.44, 7.89].

The above results are given in terms of absolute numbers. In relative terms, the response variable showed a decrease of-105%. The 95% interval of this percentage is [+32%, -241%].

This means that the negative effect observed during the intervention period is statistically significant. If the experimenter had expected a positive effect, it is recommended to double-check whether anomalies in the control variables may have caused an overly optimistic expectation of what should have happened in the response variable in the absence of the intervention.

The probability of obtaining this effect by chance is p = 0.073. This means the effect may be spurious and would generally not be considered statistically significant.

4.3.3.2 Facebook

During the post-intervention period, the response variable had an average value of approx. 82.72. In the absence of an intervention, we would have expected an average response of 104.50. The 95% interval of this counterfactual prediction is [82.62, 125.34]. Subtracting this prediction from the observed response yields an estimate of the causal effect the intervention had on the response variable. This effect is -21.78

with a 95% interval of [-42.62, 0.10]. Summing up the individual data points during the post-intervention period (which can only sometimes be meaningfully interpreted), the response variable had an overall value of 19.19K. Had the intervention not taken place, we would have expected a sum of 24.24K (24,240). The 95% interval of this prediction is [19.17K, 29.08K].

The above results are given in terms of absolute numbers. In relative terms, the response variable showed a decrease of-21%. The 95% interval of this percentage is [-41%, +0%].

This means that, although it may look as though the intervention has exerted a negative effect on the response variable when considering the intervention period as a whole, this effect is not statistically significant, and so cannot be meaningfully interpreted. The apparent effect could be the result of random fluctuations that are unrelated to the intervention. This is often the case when the intervention period is very long and includes much of the time when the effect has already worn off. It can also be the case when the intervention period is too short to distinguish the signal from the noise. Finally, failing to find a significant effect can happen when there are not enough control variables or when these variables do not correlate well with the response variable during the learning period.

The probability of obtaining this effect by chance is very small (Bayesian one-sided tail-area probability p = 0.027). This means the causal effect can be considered statistically significant.

4.3.3.3 Apple

During the post-intervention period, the response variable had an average value of approx. 79.62. By contrast, in the absence of an intervention, we would have expected an average response of 102.46. The 95% interval of this counterfactual prediction is [81.98, 123.88]. Subtracting this prediction from the observed response yields an estimate of the causal effect the intervention had on the response variable.

This effect is -22.84 with a 95% interval of [-44.25, -2.36]. For a discussion of the significance of this effect, see below.

Summing up the individual data points during the post-intervention period (which can only sometimes be meaningfully interpreted), the response variable had an overall value of 18.47K (18,470). By contrast, had the intervention not taken place, we would have expected a sum of 23.77K. The 95% interval of this prediction is [19.02K, 28.74K].

The above results are given in terms of absolute numbers. In relative terms, the response variable showed a decrease of-22%. The 95% interval of this percentage is [-43%, -2%].

This means that the negative effect observed during the intervention period is statistically significant. If the experimenter had expected a positive effect, it is recommended to double-check whether anomalies in the control variables may have caused an overly optimistic expectation of what should have happened in the response variable in the absence of the intervention.

The probability of obtaining this effect by chance is very small (Bayesian one-sided tail-area probability $p = 0.02$). This means the causal effect can be considered statistically significant.

4.4 Comparison of Bayesian network and Granger Causality results

In both Bayesian and Granger Causality approach we have seen significant causality that has been identified. In case of GC, causality weight is smaller, but instead, we can detect that in general causality shows more stability in higher intervals, (i.e 3HOUR). Meaning, we have high LAG for lower intervals in general, but as it gets to 3HOUR and 1Day, usually it stabilizes in 2-3 lags. Bayesian network analysis, can not provide any information about LAGs, but we can see that with higher weight there is a constant negative effect from sentiments on stock market returns and visa versa.

There is not a proper approach to compare the causality of these two methods

with each other, other than the overall result. Bayesian Causality network, calculates the causality by first including the sentiment and then in the absence of it. Then it calculates the estimate of causal effect which in case of Amazon, stock market variable showed a decrease of -28% when including the sentiments. It ultimately, uses the Bayesian probability to understand if that effect has been obtained by chance or not (in this case it has not). On the other hand, Granger causality is a purely statistical approach with its own parameters. We can understand if there was an auto-regressive correlation between sentiment and stock market returns, and if there is, how long does it take for the effect to take place. Therefore, what we can show is that both has shown causality within their own methodology and both are indicating the same results, with different approach and specifications.

4.5 Evaluation

Although as long as the f-test in Granger causality is statistically significant, then the causality test is proven and done, but in order to understand this causality relationship better, we attempted to investigate certain dates in different stocks and understand the news that affected company stock on certain dates and how did it affected the Twitter which created our causality results. In the next parts, for different stocks that actually showed causality with presented analysis, we focus on specific dates. While focusing on the news that actually affected the stock, we show that there was a significant trend of that news on Twitter specially focusing the news.

4.5.1 Apple Inc.

According to our Granger causality model, Apple shows a lag of two days on impact of social media on stock market return. On *February first*, Apple Inc ($APPL) released [20] its profitable first quarter report which was above expectations and the stock went up by $4. On January 31st, Apple also reported record holiday quarter,

[20] www.marketwatch.com

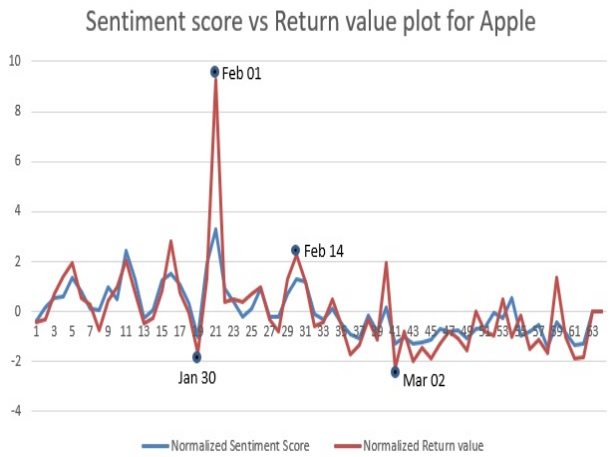


Figure (4.12) This plot, shows normalized tweeter sentiments calculated by Amazon Mechanical Turk and the Apple stock returns. We can see a similar growth trend for the sentiment score value and the return value from January 30th to February 1st.

stating iPhone7 sales boosted earnings after 3 consecutive quarters of low sales.

As it is shown in figure 4.12, we see a similar growth trend for the sentiment score value and the return value from January 30th to February 1st. On January 31st, Apple was set to post its numbers after the stock market closes, which created a trend of tweets regarding people suggesting to buy Apple stock on that day. There was a total of 354 tweets were sent by verified accounts on this topic, in these two dates. Table 4.2 shows a sample of tweets were mentioned in that two day period regarding APPL.

4.5.2 Facebook Inc.

Similar to Apple, the Granger causality model, shows a lag of two days on impact of social media on Facebook stock market return on figure 4.13. On *February first*, Facebook Inc ($FB) reaches record territory after earnings show huge growth. [21] There was a total of 200 tweets were sent by verified accounts on this topic, in these two days. Table 4.3 shows a sample of tweets were mentioned in that two day period regarding FB.

[21] www.marketwatch.com

Table (4.2) Example of Tweets targeting the Apple stock in January 31st and February 1st. There was a total of 354 tweets were sent by verified accounts on this topic, in these two dates.

Date	**Tweet**
'stockalert stocks watch today wallstreet aapl ua'	02/01/2017
'rt igtv chinas growing faster aapl results rise copper prices theres turn around sentiment'	02/01/2017
'apple iphone sales road record quarter aapl'	02/01/2017
'apple report first numbers slew new products selling including new macbook pro iphone 7 aapl'	1/31/2017
rt optionsaction 3 stocks could account 60 billion market cap swing week aapl fb amzn'	1/31/2017

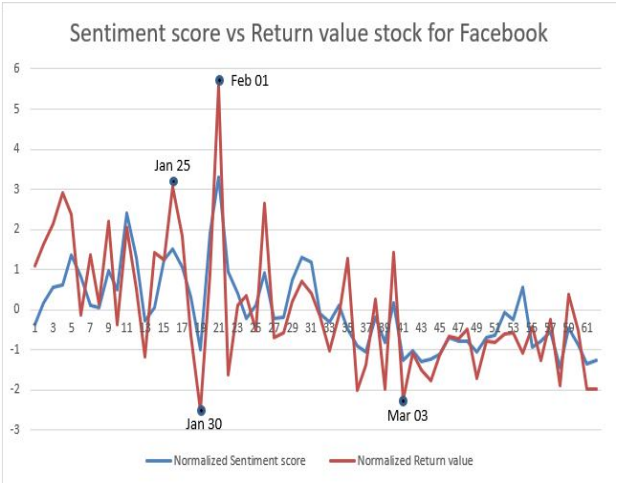


Figure (4.13) This plot shows normalized tweeter sentiments calculated by Amazon Mechanical Turk and the FaceBook stock returns. We can see a similar growth trend for the sentiment score value and the return value on multiple dates, such as Jan 25th, and February 1st.

Table (4.3) Example of Tweets targeting Facebook stock in January 31st and February 1st. There was a total of 200 tweets were sent by verified accounts on this topic, in these two days.

Date	**Tweet**
02/01/2017	'facebook earnings bell wow like apple also much trump bad tech check aapl fb amzn nflx amp nasdaq ytd'
02/01/2017	'facebook rallying close hope big number think probably see good number fb earnings'
1/31/2017	'facebook deliver another record set numbers fb'
1/31/2017	'fb winning option trading facebook take via cnnmoney' '

4.6 Chapter summary

In the first part of this chapter, we looked for causal relationships between stock return values and Twitter sentiments at one day intervals during a period of three months. The stock return values were from 83 different companies and represented diverse business sectors. Using Granger causality, we found strong evidence for bi-directional causality between many stock returns and the sentiments about the company expressed in tweets. Investigating further, we also observed that most of the stocks with a causal relationship were in the technology sector, which tend to have a higher frequency of tweets about them than companies in other sectors.

In the second part, we used an expanded dataset of stock return values that spanned a period of three years, from 2015 to 2017. Because the granularity of the return values was finer in this dataset (per minute), we partitioned both our return values and sentiment scores into five intervals: fifteen and thirty minutes, one and three hours, and one day. For each interval, we then used Granger and Bayesian causality to identify causal relationships between return values and sentiments for three companies:

Apple, Facebook, and Amazon.

Using Granger causality analysis at the different intervals for Amazon, Facebook, and Apple, we identified significant (although weak) causal links, at a lag of three hours and one day, for Amazon and Facebook. The strongest causal weight for these two stocks occurred at a three hour lag. Importantly, the causal link existed in both directions: tweets influenced future stock market returns, and stock market returns influenced future tweets. Then we used Google's time-series Bayesian causality model on the same stocks and intervals and found a significant, and strong, causal weight at an interval of one day. In addition, the Bayesian model also showed that the fluctuation of stock market returns had a negative impact on subsequent tweets, with an average weight of -25.46 and average decrease of 21%. This negative impact is supported by previous results [9, 10]. The Bayesian model did not show any significant result on other intervals. From the results of the different causal models, we conclude that there statistically significant causal link exists between the stock market and sentiments from stock market tweets.

We expect that this research will improve other research that is focused on the relationship between social media and various aspects of finance, including stock market prices, perceived trust in companies, and the assessment of brand value. For example, Calefato et al. [78] hypothesize that traditional websites and social media could show different effect on building trust in the customer-supplier relationships, based on the first impression provided to potential customers. In more details, they showed that the social media provide companies with tools to communicate to potential customers and, therefore, encourage affective commitment for the customers.

CHAPTER 5: Summary

In this thesis, we investigated the the relationship between social media (specifically, Twitter) and the stock market. Sentiment analysis of Twitter messages is a challenging task because they contain limited contextual information. Despite the popularity and significance of this task for financial institutions [9, 11], current models do not have a high accuracy, and almost all of them are not built specifically for stock market data. Therefore, there was an obvious need for a highly accurate sentiment classifier that is specifically tuned and trained for stock market data.

To understand causal relationship we need data about sentiments of stock markets. Given the lack of a publicly available Twitter dataset that is labeled with positive and negative sentiments, we first introduced a dataset of 11,000 stock market tweets that was labeled manually using Amazon Mechanical Turk. To understand if our methods for sentiment classification work, we needed to build classifiers. Deep Learning, is currently the most accurate approach to many tasks in NLP, including classification. Therefore, we reported a thorough comparison of various neural network models against different baselines. We found that when using a balanced dataset of positive and negative tweets, and a unique pre-processing technique, a shallow CNN achieved the best error rate, while a shallow LSTM, with a higher number of cells, achieved the highest accuracy of 92.7%, compared to the baseline accuracy of 79.9% using an SVM. This is a substantial improvement of the state-of-the-art for sentiment analysis of stock market tweets[8]. We expect that new models will emerge that will build on ours, and that we will see similar improvement in any research that investigates the relationship between social media and various aspects of finance, such as stock market prices, perceived trust in companies, and the assessment of brand value. The

labeled dataset of financial tweets and software are publicly available.

While our model achieved the best accuracy for the sentiment analysis of stock market tweets, there is still room for improvement. First, additional pre-processing steps could be added before predicting sentiments. For example, if a hashtag occurs frequently in a dataset, it is probably representative of a subject and should remain in the dataset in its original form. Otherwise, the hashtag should be separated and considered as individual words. Another improvement would be to increase the size of the Twitter dataset, we would allow us to try deeper neural network models. Finally, instead of using pre-trained word embeddings, we could create custom embeddings that are specific to finance.

Since ultimately we want to understand causal relationships, now that we can extract the sentiment relatively reliably, we can switch our attention to causality and their models. Although Granger causality has long been criticized for only working with two variable, and for not providing any information about why the Granger causality exists, it is simple, and is the only causality model provides information about lags in causality. On the other hand, while more complicated, Pearl's Bayesian causality model is very robust and can be used for problems with many variables. Using this framework would allow us to build more sophisticated models, such as one that incorporates a portfolio of stocks, or a volume of tweets, or a trading volume. The downside of this model is that it can be very complicated to use for time-series data, which is why we used Google's Bayesian model for the time-series analysis. While it only works on two variables, Google's model still takes advantage of the robustness of Bayesian networks.

We showed with two different causal models that there is a statistically significant causal link between the stock market and sentiments expressed in financial tweets. In particular, when Granger causality was applied to stock returns from Amazon, Facebook, and Apple at different intervals, we found that a significant (although

weak) causal link at a three hour and one day lag from the when tweets occurred. This causal link existed in both direction: tweets influenced future stock market returns, and stock market returns influenced future tweets. The strongest causal weight for all of the stocks occurred at a three hour lag. Then we used Google's time-series Bayesian causality model on our dataset and found a significant, and strong, causal weight at an interval of one day. In addition, the Bayesian model also showed that the fluctuation of stock market returns had a negative impact on subsequent tweets, with an average weight of -25.46 and average decrease of 21%. This negative impact is supported by previous results [9, 10]. While we cannot directly compare the Granger and Bayesian models, they both show that there are causal links between financial tweets and stock market returns.

The famous investment theory, the efficient market hypothesis (EMH), claims that it is impossible to outperform the market, and therefore, stocks always trade at a fair value. An important assumption of EMH is that all investors make decisions rationally, without any emotional bias. Despite its wide use, EMH struggles to explain why certain types of investments perform better than others, particularly in liquid financial markets (i.e., the stock market). This study has backed that idea, and provided evidence that there are other aspects when it comes to investment.

We expect that this research will improve other research that is focused on the relationship between social media and various aspects of finance, including stock market prices, perceived trust in companies, and the assessment of brand value. For instance, by having a model like ours that can predict sentiments scores in a financial context with high accuracy, this will improve causality analyses between social media and the stock market, and improve the prediction of stock prices from social media text [9, 7, 13, 14].

In conclusion, we have contributed to the community in three different ways. First, by providing a Twitter dataset with labeled sentiments that is specific to finance. [6]

Second, by providing a highly accurate deep learning model for labeling new financial tweets. Finally, by showing that there is indeed a causal relationship between social media (in this case, Twitter) and the stock market [16]. Taken together, we expect that this research will open new avenues for further research into the relationship between social media and finance.

REFERENCES

[1] F. Black, "Noise," 1986.

[2] B. De Long, A. Shleifer, L. Summers, and R. Waldmann, "Noise Trader Risk in Financial Markets," *Journal of Political Economy*, vol. 98, no. 4, pp. 703–738, 1990.

[3] O. Kolchyna, T. T. P. Souza, P. Treleaven, and T. Aste, "Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination," p. 32, 2015.

[4] S. Du and Z. Xi, "SemEval17.pdf," no. 39, pp. 120–125, 2016.

[5] M. Jiang, M. Lan, and Y. Wu, "ECNU at SemEval-2017 Task 5: An Ensemble of Regression Algorithms with Effective Features for Fine-Grained Sentiment Analysis in Financial Domain," *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 885–890, 2017.

[6] N. Tabari, A. Seyeditabari, and W. Zadrozny, "SentiHeros at SemEval-2017 Task 5 : An application of Sentiment Analysis on Financial Tweets," pp. 857–860, 2017.

[7] T. I. M. Loughran and B. Mcdonald, "When is a Liability not a Liability ? Textual Analysis , Dictionaries , and 10-Ks Journal of Finance , forthcoming," 2010.

[8] S. Sohangir, D. Wang, A. Pomeranets, and T. M. Khoshgoftaar, "Big Data: Deep Learning for financial sentiment analysis," *Journal of Big Data*, vol. 5, no. 1, 2018.

[9] J. Bollen and A. Pepe, "Modeling Public Mood and Emotion : Twitter Sentiment and Socio-Economic Phenomena," pp. 450–453, 2011.

[10] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *Journal of Finance*, vol. 62, no. 3, pp. 1139–1168, 2007.

[11] D. Antenucci, M. Cafarella, M. C. Levenstein, C. Rei, and M. D. Shapiro, "Using Social Media to Measure Labor Market Flows Dolan Antenucci," *Nber*, 2014.

[12] T. T. P. Souza, O. Kolchyna, and T. Aste, "Twitter Sentiment Analysis Applied to Finance: A Case Study in the Retail Industry," no. i, p. 19, 2015.

[13] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grar, and I. Mozeti, "The effects of twitter sentiment on stock price returns," *PLoS ONE*, vol. 10, no. 9, pp. 1–21, 2015.

[14] F. Lillo, S. Miccich, M. Tumminello, and J. Piilo, "How news affect the trading behavior of different categories of investors in a financial market," *Papers.Ssrn.Com*, no. April, p. 30, 2012.

[15] M. Alanyali, H. S. Moat, and T. Preis, "Quantifying the relationship between financial news and the stock market.," *Scientific reports*, vol. 3, p. 3578, 2013.

[16] N. Tabari, U. N. C. Charlotte, U. N. C. Charlotte, U. N. C. Charlotte, U. N. C. Charlotte, and U. N. C. Charlotte, "Causality Analysis of Twitter Sentiments and Stock Market Returns," pp. 11–19, 2018.

[17] P. D. Turney, "Thumbs up or thumbs down? Semantic Orientation applied to Unsupervised Classification of Reviews," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, no. July, pp. 417–424, 2002.

[18] E. Brill, "Some advances in rule-based part of speech tagging," *Proceedings of the Twelfth Annual Conference on Artificial Intelligence*, pp. 722–727, 1994.

[19] C. D. Manning, "Foundations of Statistical Natural Language Processing,"

[20] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," *In Proceedings of the Seventh Conference on International Language Resources and Evaluation*, pp. 1320–1326, 2010.

[21] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Processing*, vol. 150, no. 12, pp. 1–6, 2009.

[22] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 11)*, pp. 538–541, 2011.

[23] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," *Association for Computational Linguistics*, pp. 30–38, 2011.

[24] L. Barbosa and J. Feng, "Robust Sentiment Detection on Twitter from Biased and Noisy Data," *Coling*, no. August, pp. 36–44, 2010.

[25] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends® in InformatioPang, B., & Lee, L. (2006). Opinion Mining and Sentiment Analysis. Foundations and Trends® in Information Retrieval, 1(2), 91-231. doi:10.1561/1500000001n Retrieval*, vol. 2, no. 1-2, pp. 91–231, 2008.

[26] K. Dave, K. Dave, S. Lawrence, S. Lawrence, D. Pennock, and D. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, 2003.

[27] L. Wasden, "20 Internet Acronyms Every Parent Should Know," no. February, p. 44 pages, 2010.

[28] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[29] S. R. Das and M. Y. Chen, "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[30] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *Empirical Methods in Natural Language Processing (EMNLP)*, vol. 10, no. July, pp. 79–86, 2002.

[31] N. O'Hare, M. Davy, A. Bermingham, P. Ferguson, P. P. Sheridan, C. Gurrin, A. F. Smeaton, and N. OHare, "Topic-Dependent Sentiment Analysis of Financial Blogs," *International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, pp. 9–16, 2009.

[32] T. M. Mitchell, *Machine Learning*. 1997.

[33] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7649 LNCS, no. PART 1, pp. 508–524, 2012.

[34] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets," *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, vol. 2, no. SemEval, pp. 321–327, 2013.

[35] P. Nakov, S. Rosenthal, A. Ritter, and T. Wilson, "SemEval-2013 Task 2: Sentiment Analysis in Twitter," *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2013)*, vol. 2, no. SemEval, pp. 312–320, 2013.

[36] H. Hamdan, "Experiments with DBpedia , WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging," *Seventh International Workshop on Semantic Evaluation (SemEval 2013) - Second Joint Conference on Lexical and Computational Semantics*, vol. 2, no. SemEval, pp. 455–459, 2013.

[37] N. F. F. Da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[38] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[39] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 53–56, 2011.

[40] A. Aue & M. Gamon, "Customizing Sentiment Classifiers to New Domains: A Case Study.," *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, vol. 3, no. 3, pp. 16–18, 2005.

[41] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

[42] C. N. dos Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts," *Coling-2014*, pp. 69–78, 2014.

[43] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning Sentiment-Specific Word Embedding," *Acl*, pp. 1555–1565, 2014.

[44] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A Deep Learning System for Twitter Sentiment Classification," *Semeval-2014*, no. SemEval, pp. 208–212, 2014.

[45] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification," *Acl-2014*, pp. 49–54, 2014.

[46] D. T. Vo and Y. Zhang, "Target-dependent twitter sentiment classification with rich automatic features," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2015-Janua, no. Ijcai, pp. 1347–1353, 2015.

[47] A. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*, pp. 959–962, 2015.

[48] D. a. Shamma, L. Kennedy, and E. F. Churchill, "Tweet the Debates: Understanding Community Annotation of Uncollected Sources," *Proceedings of the First SIGMM Workshop on Social Media*, pp. 3–10, 2009.

[49] R. Chen and M. Lazer, "Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement," pp. 1–5, 2013.

[50] A. Devitt and K. Ahmad, "Sentiment polarity identification in financial news: a cohesion-based approach," *Proceedings of Annual Meeting of the Association of Computational Linguistics*, no. June, pp. 984–991, 2007.

[51] M. Baker and J. Wurgler, "Investor sentiment and the cross-section of stock returns," *Journal of Finance*, vol. 61, no. 4, pp. 1645–1680, 2006.

[52] A. Davis, J. Piger, and L. Sedor, "Research Division," *City*, vol. 21, pp. 1791–1814, 2006.

[53] X. Yu, G. Mitra, and K. Yu, "Impact of News on Asset Behaviour: Return, Volatility and Liquidity in an Intra-Day Setting," *SSRN Electronic Journal*, no. July, 2013.

[54] T. O. Sprenger, P. G. Sandner, A. Tumasjan, and I. M. Welpe, "News or noise? Using twitter to identify and understand company-specific news flow," *Journal of Business Finance and Accounting*, vol. 41, no. 7-8, pp. 791–830, 2014.

[55] S. Shen, H. Jiang, and T. Zhang, "Stock Market Forecasting Using Machine Learning Algorithms," pp. 1–5, 2012.

[56] R. Snow, B. O. Connor, D. Jurafsky, A. Y. Ng, D. Labs, and C. St, "Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks," no. October, pp. 254–263, 2008.

[57] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[58] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[59] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," pp. 1–12, 2013.

[60] R. Johnson and T. Zhang, "Deep Pyramid Convolutional Neural Networks for Text Categorization," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 562–570, 2017.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[62] X. Zhao, C. Wang, Z. Yang, Y. Zhang, and X. Yuan, "Online news emotion prediction with bidirectional lstm," pp. 238–250, 2016.

[63] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014.

[64] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013.

[65] C. Hitchcock, "Probabilistic causation," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, winter 2016 ed., 2016.

[66] P. Suppes, *A probabilistic theory of causality*. Amsterdam : North-Holland Pub. Co, 1970. Bibliography: p. [121]-124.

[67] C. W. J. Granger and N. Aug, "Investigating Causal Relations by Econometric Models and Cross-spectral Methods," vol. 37, no. 3, pp. 424–438, 1969.

[68] J. Geweke and J. Geweke, "Measurement of Linear Dependence and Feedback Between Multiple Time Series Measurement of Linear Dependence and Feedback Betwveen Multiple Time Series," vol. 77, no. 378, pp. 304–313, 2018.

[69] S. Wright, "Correlation and causation," *Journal of agricultural research*, vol. 20, no. 7, pp. 557–585, 1921.

[70] S. Russell, "Bayesian networks," pp. 1–7, 2001.

[71] "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems Author (s): S . L . Lauritzen and D . J . Spiegelhalter Source : Journal of the Royal Statistical Society . Series B (Methodological), Vol . 50 , No . 2 Published by : Blackwell Publishing for the Royal Statistical Society Stable URL : http://www.jstor.org/stable/2345762," vol. 50, no. 2, pp. 157–224, 2009.

[72] J. Binder, D. Koller, S. Russell, and K. Kanazawa, "Adaptive probabilistic networks with hidden variables," *Machine Learning*, vol. 29, no. 2-3, pp. 213–244, 1997.

[73] N. Friedman, "The bayesian structural em algorithm," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 129–138, Morgan Kaufmann Publishers Inc., 1998.

[74] C. Street, "by Judea Pearl," pp. 675–685, 2003.

[75] S. Morgan and C. Winship, *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. Analytical Methods for Social Research, Cambridge University Press, 2007.

[76] J. Pearl, *Causality: Models, Reasoning, and Inference*. New York, NY, USA: Cambridge University Press, 2000.

[77] "Inferring causal impact using bayesian structural time-series models," vol. 9, no. 1, pp. 247–274, 2015.

[78] F. Calefato and N. Novielli, "The role of social media in affective trust building in customer - supplier relationships," *Electronic Commerce Research*, vol. 15, no. 4, pp. 453–482, 2015.

APPENDIX A: FEATURE VECTORS: ADDITIONAL WORDS AND WORD-COUPLES

A.1 Positive words added to Loughran's list

"cover, cool, top, yes, smart, smartly, epic, highs, recover, profit, profits, long, upside, love, interesting, loved, dip, dipping, secure, longs, longput, rise, able, okay, buy, buying"

A.2 Negative words added to Loughran's list

"avoid, notokay, little, less, cray, no, crash, crashes, leaves, terrible, struggles, struggled, stall, stalls, stalled, lows, fakenews, mess, exit, not, cheaper, cheap, slaughter, slaughtered, slaughtering, disgusting, cult, brutal, fucked, suck, decay, bubble, bounce, bounced, low, lower, selloff, disgust, meltdown, downtrend, downtrends, censored, toppy, scam, censor, garbage, risk, steal, retreat, retreats, sad, dirt, flush, dump, plunge, plunged, crush, crushed, crying, unhappy, drop, dropping, drops, cry, dumped, torture, short, shorts, shorting, fall, falling, sell, selling, sells, bearish, slipping, slip, sink, sinked, sinking, pain, shortput, bullshit, shit, nervous, damn, broke, breakup, overbought"

A.3 Negative Word-Couples replaced by "notokay"

(no, long), (pay, well), (no, higher), (lower, high), (terrible, market), (lose, momentum), (lost, momentum), (loses, momentum), (not, enjoy), (not, good), (lower, profit), (fall, short), (dont, trust), (poor, sales), (not, working), (cut, pay), (cuts, pay), (fake, news), (wasnt, great), (lost, profit), (losses, profit), (lose, profit), (new, low), (cant, growth), (cant, profitable), (terrible, idea), (short, sellers), (raises, concern), (raise, concern), (not, recommend), (not, recommended), (not, much), (big, debt), (high, down), (lipstick, pig), (doesnt, well), (bounce, buy), (isnt, cheap), (fear, sell), (cant, down), (not, good), (wont, buy), (dont, trade), (buy, back), (didnt, like),

(profit, exit), (go, down), (not , guaranteed), (not, profitable), (doesn't, upward), (not, dip), (pull, back), (not, optimistic), (go, up, okay), (not, affected, okay), (not, concerned, okay), (short, trap, okay), (exit, short, okay), (sell, exhaust, okay), (didnt, stop, okay), (short, cover, okay), (close, short, okay), (short, break, okay), (cant, risk, okay), (not, sell, okay), (dont, fall, okay), (sold, call, okay), (dont, short, okay), (exit, bancruptsy, okay), (not, bad, okay), (short, nervous, okay), (dont, underestimate, okay), (not, slowdown, okay), (aint, bad, okay), (first, second, replacement)

A.4 Positive Word-Couples replaced by "okay"

(go, up), (not, affected), (not, concerned), (short, trap), (exit, short), (sell, exhaust), (didnt, stop), (short, cover), (close, short), (short, break), (cant, risk), (not, sell), (dont, fall), (sold, call), (dont, short), (exit, bancruptsy), (not, bad), (short, nervous), (dont, understimate), (not, slowdown), (aint, bad)

APPENDIX B: RESULTS OF GRANGER CAUSALITY

B.1 F-test and P-value for Model 1

Symbol	AMT Lag	F-value	P-value	ML Lag	F-value	P-value
AABA	NS			6	2.76	0.023
AAL	2	3.99	0.024	2	4.2	0.02
AAPL	3	4.23	0.01	3	5.68	0.002
AVGO	2	3.85	0.027	6	2.87	0.02
BABA	NS			7	2.86	0.016
BAC	2	3.44	0.039	NS		
CREE	4	3.11	0.024	NS		
CSCO	9	2.55	0.024	NS		
CSX	9	2.47	0.028	NS	2.17	0.049
EA	4	3.13	0.023	NS		
EBAY	6	2.39	0.045	6	2.33	0.05
ENDP	5	2.53	0.042	5	2.7	0.032
FAST	10	2.28	0.039	NS		
FB	4	2.84	0.034	NS		
FDX	2	3.41	0.04	NS		
GALE	9	2.47	0.028	NS		
ISRG	3	6.31	0.001	3	4.01	0.012
KNDI	2	3.71	0.031	2	3.81	0.028
LUV	2	3.93	0.025	2	2.23	0.117
MAR	2	3.49	0.038	NS		
MNKD	2	3.75	0.03	2	3.57	0.035
MSFT	2	3.8	0.029	4	2.94	0.03
NFLX	2	4.64	0.014	2	4.16	0.021
NXPI	5	3.93	0.005	5	3.12	0.017
QCOM	7	2.6	0.027	9	2.31	0.038
SBUX	4	2.7	0.042	5	2.35	0.048

B.2 F-test and P-value for Model 2

Symbol	AMT Lag	F-value	P-value	ML Lag	F-value	P-value
AAPL	2	5.86	0.005	2	3.98	0.024
AGN	4	2.65	0.045	4	3.1	0.024
AMZN	3	2.93	0.042	3	3.01	0.038
BABA	6	2.61	0.03	NS		
CELG	10	2.57	0.022	10	2.58	0.022
COST	2	4.16	0.021	2	3.89	0.026
CSCO	NS			2	3.59	0.034
FB	2	3.83	0.028	2	4.31	0.018
FFIV	NS			3	2.95	0.041
GALE	4	3.65	0.011	4	4.14	0.006
GILD	6	2.72	0.025	6	2.54	0.035
MSFT	5	3.06	0.018	5	2.5	0.044
PLUG	10	2.37	0.033	10	2.19	0.047
REGN	7	2.45	0.035	6	2.38	0.046
SINA	5	2.5	0.044	NS		
STX	NS			3	2.98	0.04
TWTR	5	3.81	0.006	5	4.89	0.001
YELP	2	3.34	0.043	6	3.07	0.014
ZNGA	NS			6	2.53	0.035

B.3 F-test and P-value for Three year data: Stock Return Causes Sentiment

Stock Ticker	Granularity	Fvalue	Pvalue	LagNo
AMZN	15Min	4.314886091	0.013375864	2
AMZN	30Min	2.069420721	0.043239253	7
AMZN	1H	4.590555386	0.010167487	2
AMZN	3H	11.85706787	7.31E-06	2
APPL	1H	2.395314948	0.014132495	8
FB	15Min	6.240267558	0.00195362	2
FB	1H	2.633886744	0.032417672	4
FB	3H	6.264219585	0.00193428	2

B.4 F-test and P-value for Three year data: Sentiment causes the Stock Return

Stock ticker	Granularity	Fvalue	Pvalue	LagNo
AMZN	1D	3.64085225	0.026755012	2
AMZN	3H	4.339620147	0.013096581	2
AMZN	1H	2.895445985	0.033813733	3
AMZN	30Min	2.065940695	0.04361101	7
APPL	30Min	2.081200907	0.034077796	8
FB	15Min	4.004818473	0.018244098	2
FB	3H	14.74917723	4.30E-07	2
FB	30Min	2.317443647	0.040980975	5