

Sentiment Analysis  
for Long-Term Stock Prediction

by

Tyler Joseph Reeves

A Thesis Presented in Partial Fulfillment  
of the Requirements for the Degree  
Master of Science

Approved April 2016 by the  
Graduate Supervisory Committee:

Hasan Davulcu, Chair  
John Cesta  
Chitta Baral

ARIZONA STATE UNIVERSITY

August 2016

## ABSTRACT

There have been extensive research in how news and twitter feeds can affect the outcome of a given stock. However, a majority of this research has studied the short term effects of sentiment with a given stock price. Within this research, I studied the long-term effects of a given stock price using fundamental analysis techniques. Within this research, I collected both sentiment data and fundamental data for Apple Inc., Microsoft Corp., and Peabody Energy Corp. Using a neural network algorithm, I found that sentiment does have an effect on the annual growth of these companies but the fundamentals are more relevant when determining overall growth. The stocks which show more consistent growth hold more importance on the previous year's stock price but companies which have less consistency in their growth showed more reliance on the revenue growth and sentiment on the overall company and CEO. I discuss how I collected my research data and used a multi-layered perceptron to predict a threshold growth of a given stock. The threshold used for this particular research was 10%. I then showed the prediction of this threshold using my perceptron and afterwards, perform an f anova test on my choice of features. The results showed the fundamentals being the better predictor of stock information but fundamentals came in a close second in several cases, proving sentiment does hold an effect over long term growth.

## TABLE OF CONTENTS

	Page
LIST OF FIGURES.....	iv
LIST OF EQUATIONS.....	v
CHAPTER	
1. SENTIMENT ANALYSIS FOR MEASURED SECURITY GROWTH.....	1
1.1 Introduction.....	1
1.2 Problem Description.....	2
1.3 Fundamental Analysis.....	3
1.3.1 Problems with Fundamental Analysis.....	4
1.4 Quantitative, Qualitative, and Economic Analysis.....	5
1.4.1 Quantitative/Technical Analysis.....	6
1.4.2 Qualitative Analysis.....	9
1.5 Methodologies.....	10
1.5.1 Sentiment Analysis.....	10
1.5.2 Information Extraction and Web Scraping.....	12
1.5.3 Neural Networks.....	13
1.6 Our Contribution.....	14
2. INFORMATION EXTRACTION / WEB SCRAPING.....	16
2.1 Basic Linguistics.....	16
2.2 Information Extraction.....	19

CHAPTER	Page
2.2.1 Sentence Segmentation.....	21
2.2.2 Tokenization.....	22
2.2.3 Parts of Speech Tagger.....	24
2.2.4 Entity Detection.....	25
2.2.5 Relation Detection.....	27
2.3 Rule Versus Pattern Implementation.....	27
2.3.1 Choice of Structure.....	32
3. NEURAL NETWORKS.....	33
3.1 Introduction.....	33
3.2 Human Brain and Computational Machines.....	34
3.3 Artificial Neural Network.....	37
3.4 Logistic Regression.....	40
3.5 Artificial Neural Network Learning.....	42
3.6 Literary Conclusion.....	44
4. NEURAL NETWORKS SENTIMENT ARCHITECTURE.....	46
4.1 Related Research- Time Variant.....	46
4.2 Related Research- Multi-Agent Architecture.....	47
4.3 Related Research- Type of Data.....	48
4.4 System Architecture.....	48
4.4.1 Main Agent.....	50
4.4.2 SEC Web Scraper Agent.....	50

CHAPTER	Page
4.4.3 External Fundamental Agent.....	52
4.4.4 Tweet Agent.....	52
4.4.5 Prediction Agent.....	54
4.5 Conclusion.....	55
5. TESTING AND ANALYSIS.....	56
5.1 Test Program.....	56
5.2 Results.....	60
6. CONCLUSION.....	62
REFERENCES.....	63
APPENDIX	
A APPLE DATA COLLECTED.....	65
B MICROSOFT DATA COLLECTED.....	67
C PEABODY ENERGY DATA COLLECTED.....	69

## LIST OF FIGURES

Figure	Page
1. Apple Dividend Trend.....	8
2. Apple Stock Trend.....	8
3. Single-Layer Perceptron.....	14
4. Example Grammar.....	17
5. Natural Language Processing Process.....	20
6. Example Parse Tree.....	26
7. Brain Neuron.....	35
8. Axon of Brain Neuron.....	36
9. Correlation of Brain Neuron to Artificial Neuron.....	37
10. Building Blocks of Artificial Neuron.....	38
11. Logistical Regression Limit Graph.....	42
12. Related Research: Agent Architecture.....	49
13. Agent Architecture.....	49
14. Data Collection Fields.....	56
15. Python: Main Method.....	57
16. Python: Quantitative Analyzer.....	58
17. Python: Qualitative Analyzer.....	59
18. Apple Anova Test Results.....	61
19. Microsoft Anova Test Results.....	61

## LIST OF EQUATIONS

Equation	Page
1. Asset Balance Equation.....	1
2. Sigmoid Function.....	41
3. Reverse Sigmoid Function Equation.....	41
4. Logistic Regression Feature Weights.....	42
5. Logistic Regression Feature Weight (Matrix Form).....	42
6. Perceptron Error Output Function.....	43
7. Perception Error Output Derivative.....	43
8. Logistic Function Derivative.....	44
9. Perceptron Error (with respect to previous nodes).....	44
10. X Derivative with Output Layer Dependence.....	44
11. Perceptron Hidden Layer Error.....	45

## Chapter One

### Sentiment Analysis for Measured Security Growth

#### 1.1: Introduction

Over the course of the past century, many have developed methods which are meant to predict the future price of a given stock with a certain level of accuracy. Today, there are two different methods which are used to analyze a company and its security price. The first method is named quantitative analysis. Quantitative analysis involves the study of a company's previous numbers which give an indication of what the company's future numbers will be. When computers became more sophisticated and began performing more computationally expensive tasks, people began creating algorithms which could be used to predict a securities price in the near future. A majority of these algorithms involve using machine learning techniques which can measure a time-series and make a prediction with certain degree of belief of what the price will be in the future. Therefore, these methods became very valuable when performing quantitative analysis on corporations. However, the second method used to analyze a company is called qualitative analysis. Qualitative analysis involves a more non-numerical approach in analyzing a firm's growth. A more formal definition would be securities analysis that uses subjective judgment based on unquantifiable information, such as management expertise, industry cycles, strength of research and development, and labor relations.[1] If I were to perform a complete qualitative analysis on a particular corporation, I would study the quality of the company's chief officers such as the chief executive officer, chief financial officer, and even board of directors. Also, I would analyze the quality of



product or service the company provides and how the market receives the company as an organization. The non-numerical analyzation task is a more difficult for a computing machine to perform. The purpose of this research is to create ways to use the computer to not only analyze a company's financial numbers, but also to conduct a more qualitative analysis which can show the effect of the company's non-numerical aspects.

## 1.2 Problem Description

In recent years, there have been many different machine learning algorithms which are used to predict a stock's future price. Some of the more widely used algorithms are neural networks as well as support vector machines. These algorithms have been used for both classification and time-series analysis of a variety of numbers. These numbers include the past price of the stock itself, company financials, and stock ratios. However, there exists problems with these machine learning algorithms. First, a majority of these algorithms only focus on a short term stock value gain. As algorithms try to predict further into the future, training data becomes more scarce which may result in testing accuracy being greatly diminished. Also, many of these machine learning based systems only take into account a limited amount of fundamental data. In more detail, it is widely known a stock's future price (also the company's value) is dependent on financial statements, ratios, and quantifiable data which can be used in statistical analysis and determining trends. However, it is also the case a stock's future price is also dependent on data which is by default unquantifiable. For example, a stock's value is dependent on the value of the company it represents and the value of the company is

dependent on the quality of management which is directing the company. Also, a company's growth is dependent on the quality of product or service offered. These are attributes which cannot be quantifiably measured and yet they can hold significant influence over the future value of a stock. This is because these attributes are part of a qualitative analysis approach. This type of analysis is generally not made by computers but rather experienced investors who know the qualities of a good company versus ones which should not be invested. Therefore, the problem in which we are trying to solve is to find a way to be able to quantify the unquantified qualities of a particular company. By doing this, not only will we be able to write programs which have more predictive power in determining the trends of a given stock, but also programs which take into account both quantitative and qualitative data can help us better understand why certain stock trends occur.

### 1.3 Fundamental Analysis

Our goal is not to find stocks where we can invest to make short term gains. The goal of this research is to find a method to automatically find companies which are good investments as a whole. This means we must classify a company as a "good investment" only if the company shows signs of long term growth. For this reason, we are going to use strategies and tactics of an investor who uses fundamental analysis to analyze companies for possible investment. Fundamental analysis is a widely known method for finding a company's intrinsic value. The intrinsic value of a company is the "true" value of a company based upon specific parameters which are both quantitative and qualitative

[3]. In other words, an investor who is using fundamental analysis to analyze a company for potential investment opportunity would study both the company's financial and managerial aspects. From this analyzation, the investor will compute a number identified as the intrinsic value of the company of interest. The intrinsic value will then show the true value of what the stock should be trading at (or what the investor believes the stock should be trading at).

### 1.3.1 Problems with Fundamental Analysis

Fundamental analysis, if conducted correctly, can be a very effective way to analyze investment opportunity. However, fundamental analysis also comes with two issues which still are prevalent today. One of the main hindrances of fundamental analysis is the time factor. When an investor comes to a conclusion of the true intrinsic value of where a stock price should currently be, even if the investor is correct in their analyzation, it is very difficult to determine an accurate time frame of when the actual stock price will settle to the investor's projected intrinsic value. If an intrinsic value is found, the analyzer may not see the stock price readjust to the new price for a considerable amount of time, if ever. The second problem with fundamental analysis is how undefined the process is. The best method of analyzing and computing fundamental information to obtain the intrinsic value of a company has many different variations. Fundamental analysis and finding the best method to analyze a company's fundamental information has been the heart of many academic studies for many years. One of our primary objectives for this research is to use machine learning to help resolve some of the

issues with fundamental analysis. Also, the way we define our data and how the machine learning algorithms process the data will help us define a strict timeline of when our predictions will be reflected within the stock price. More of these methods will be explained in chapter two and chapter three.

#### 1.4: Quantitative, Qualitative, and Economic Analysis

Among the many different ways of computation, there exists two major analysis which are universally accepted as needed in order to be able to give a proper projection of what a company's intrinsic value is. The first analyzation method is quantitative analysis which in fundamental analysis is the study of the company's financial information and statements such as income statements, balance statements, cash-flow statements, and ratio valuations. The second analyzation is qualitative analysis which considers all the unquantifiable information of the company such as the overall management of the company, the competency of the board of directors, the underlying management throughout the company, and the customer satisfaction with the product or service which the company offers. These are all factors which are not inherently quantifiable but still hold influence on a company's true value and growth. For our particular research, we decided to incorporate a third element of analization to our fundamental analysis to ensure we are taking to account as many factors as possible which influence the company of interest's intrinsic value and stock price growth. Therefore, the last analyzation which was considered in our fundamental analysis are miscellaneous economic factors which may hold an affect of the overall stock price such as the influence of the overall stock

market and the performance of the overall industry. The majority of research performed was finding ways to quantify qualitative information. However, by the definition of fundamental analysis, it is important we consider both quantitative and qualitative information. Therefore, it is important to have a firm understanding of the two primary analysis and how they can be utilized in our research.

#### 1.4.1: Quantitative/Technical Analysis

One of the key concepts in using fundamental analysis is the analyzation of a company's current financial position and trend. In quantitative analysis, our objective is to conduct an analysis on a firm's financial statements to find trends and indicators of where the security price will be in the future. In order to understand how quantitative analysis works, one must first understand the different types of financial statements which companies use to record their financial positions, trends, and indicators. A company will record and update their financial position when a fiscal period ends. The duration of a fiscal period depends on the institution but it is commonly either a quarter basis or annual basis. The first statement is the company's income statement. In the income statement, a company must record its revenues from money making operations as well as its expenses. From these recordings, the income statement will then calculate the company's profits, before tax income, and after tax income. The next statement is the statement of cash flows. In the statement of cash flows, the company will record the current quarter's flow of cash whether it be positive or negative. This number is generally taken from the income statement. However, it is important the statement of

cash flows captures not just income from operations, but also from all investments of the company as well as investor financing. The last statement is the balance statement. The balance statement gives a pictorial view of the company's current equity versus liabilities on the owned assets. In other words, the balance statement shows how much ownership versus how much debt the company has on the currently owned assets. Therefore, it is important to note the company's total assets is equivalent to total liabilities plus equities.

$$Total\ Assets = Total\ Liabilities + Total\ Equity \quad (1)$$

There exists more financial statements which may be recorded but are beyond the scope of this research. Using these three statements, an investor performing a quantitative analysis will study not only the current statements, but also the trends of numbers such as incomes, profits, liabilities and many others. For example, in figures one and two, we can see the trend for dividends and stock growth over a five year period. With the naked eye, there appears to exist a relationship between the dividend yield growth of the company and the overall stock price. For statistical analysis and machine learning, quantitative analysis is simple under the condition that an adequate amount of data is available. This poses a problem for creating algorithms which predict long-term growth for the reason it is difficult to find a sufficient number of



Figure 1



Figure 2

data to be able to predict that far out into the future. With that aside, quantitative analysis is important in determining the growth or decline of a particular stock and will need to have a representation within this research.

#### 1.4.2: Qualitative Analysis

When using fundamental analysis, it is important to know the financial positions and trends of a given company. However, it has been consistently proven these are not the only factors which can have a significant effect on a company's overall growth or decline. There exists many other factors which can have significant impact on a given company's stock price. For an investor who is conducting a qualitative analysis on a particular company, the investor will research the company's aspects of the company, which are inherently not quantifiable, and then make a determination whether or not the company represents an investment opportunity or not. There are several common aspects which are examined when conducting a qualitative analysis. First, an investor will examine and review the current managing officers of the company. The possible extent of research may include the history managerial practices of the current chief executive officer, chief financial officer, chief operations officer, chief investment officer, and chief marketing officer. Also, an investor may choose to analyze the quality of the current board of directors managing the company. An investor should know not only the history of how these particular individuals have become managers, but also how what management culture they bring to the company. In other words, it is important to examine the company's current business model and decide whether or not the business model is



well suited for the business. Another factor which an investor would inspect is a company's ability to compete within its given market. On top of this, it is important to also research the industry itself and analyze whether the industry is growing or contracting.

### 1.5: Methodologies

These are all factors which can have a significant impact on the growth or decline of a given stock. However, what makes it difficult for a computational machine to use these factors in an analysis to predict the direction of growth for a given company is these factors are inherently unquantifiable. In order to include these within a feature set, we must be able to both obtain these features and be able to quantify them which can then be inputted within a machine learning algorithm for prediction. In order to do this, we had to utilize natural language processing to analyze natural language input, quantify it, and input it to a classification algorithm. In the following sections, I will give a brief summary of the natural language processing algorithms used within this research and the machine learning algorithms which was used to classify and predict whether or not a company would be “a good investment opportunity” or “not a good investment opportunity”.

#### 1.5.1: Sentiment Analysis

As stated in a previous section, one of the main shortcomings of machine learning algorithms today is the fact they can only measure quantitative features of companies. It

is important to measure what the current popular opinion is about the company and its products or services. Therefore, we want to find a way to quantify the quality of a company and use that measurement along with innate quantitative features of a company to measure the long term trend of a stock. One strategy we can use to accomplish this is to use sentiment analysis (also called opinion mining). Sentiment analysis is the extraction of people's opinions, appraisals, and emotions towards organizations, entities, people, issues, and topics. Mining opinions can be done on three different levels which all will all. The first level is document level tasks which are documents inputted into classification algorithms and classified into certain classifications. In subjectivity classification, a document can either be classified as positive, negative, or neutral. The same can be done at the sentence level of classification. This is where we can classify individual sentences to be classified as positive, negative or neutral. However, we must classify at a lower level of classification because documents or even sentences may be overall positive or negative, but there may be aspects within the sentence which may be classified differently than the overall classification. Therefore we must classify text at the phrase level as well.[4] If we are to classify an entire document on a particular company (such as a news article or list of comments) as positive or negative, we may overlook several comments and sections which themselves classify differently than the classification of the entire document. It is therefore assumed we must be as specific as possible when classifying text.

### 1.5.2: Information Extraction and Web Scraping

It is a relatively new concept of a machine having the ability to extract information through raw text. Information extraction is the extraction of raw data and organizing the data into a structured format. There currently exists two popular methods in writing an information extraction system and creating structured information from raw data text. One method is to extract information using a rule-based system. The rule based system is a methodology where the programmer creates a set of rules for the program to abide by. These rules will determine which words represent entities of interest and what relations they have with other entities. The other method will be to create a system which is based upon supervised learning methods and using a training set to train a classification learner to predict which words are entities of interest and what relationship they hold with other entities. We will discuss both methods and why we chose to use a rule-based system. We must use information extraction because one of the hindrances of using supervised learning algorithms to predict future trends of stock prices is the lack of data available. In our research, we created a web scraping program to mine information from SEC filings. A web scraper is a program which can process web pages and extract information from websites. To successfully extract raw data from the text, we created the web scraper by utilizing natural language processing techniques and information extraction to extract and create a dataset. In the end, this dataset is used to train and validate our machine learning algorithm which classifies the trend of a given stock. In the following chapter, we will discuss the concepts and methods of how we created the web scraper and created our dataset used in training.

### 1.5.3: Neural Networks

In our analysis, we had to choose a machine learning algorithm to properly classify and predict whether a company should be classified as a “good investment” or “not a good investment”. For our purposes, we chose to use the neural networks machine learning algorithm. Neural networks are built based upon the biology and functionality of the human brain. In a neural network implementation, we must build a perceptron to properly split our data into the two classification categories of interest. To build the perceptron, there are several elements which are critical to the functionality. Figure 3 displays a single layer neural network perceptron[5]. In the single first layer, we have input nodes feeding information into a sigmoid function. As displayed in the figure, we can see the inputs are set with weights which are also present in the sigmoid function. The sigmoid function then returns a value which is greater than, equal to, or less than zero. A model such as this one can be trained with a given training set. Several training models exist such as backpropagation learning, deep learning, and base function learning. Also, as mentioned previously, this figure

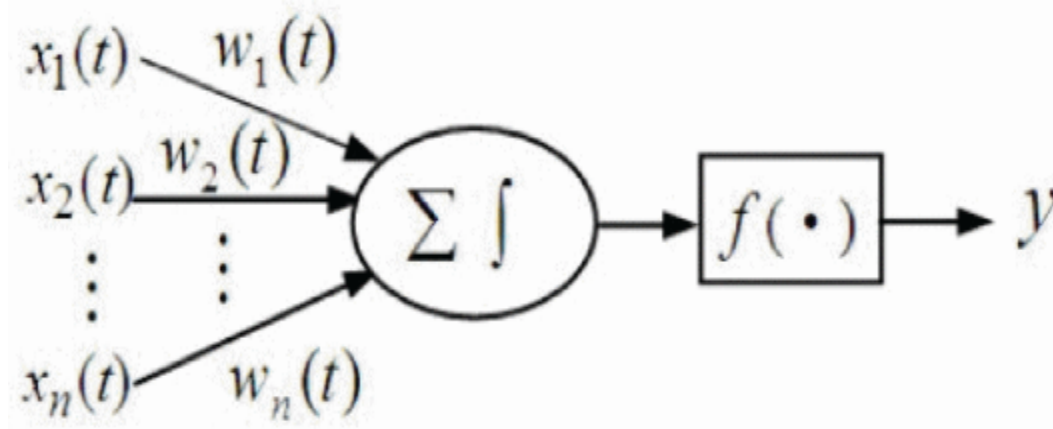


Figure 3

only

displays a single-layer neural network perceptron. It is more common a neural network will have multiple layers where the input nodes will feed into what is called a hidden layer and possibly more hidden layers. Each layer has its own set of weights which can be trained using the neural networks learning algorithms mentioned above. In chapter three, we will give a more detailed explanation of how more complex neural networks work, how the different learning algorithms function, why we decided to use the learning algorithms we did, and how we utilized neural networks as a whole within our research.

### 1.6: Our Contribution

The contribution we have made in this research is the methodology we have chosen to study the trends of stock prices. There exists two major hurdles in the study and prediction of stock growth. The first hurdle is the lack of existing data for quantifiable information for a given company. If we are looking to predict the long term

growth of a company (or stock value), then it is evident a company's quantifiable information such as incomes, profits, dividends, and cash flows can hold a significant impact on the magnitude and speed of company growth. However, most of this financial information can be difficult to obtain. Also, it can be even more difficult to obtain historical financial data. Our contribution to this problem will be the algorithms we create to obtain this data to construct a structured dataset. The algorithms will utilize natural language processing techniques to mine the semi-structured United States Securities and Exchange Commission website where companies are required by law to post all financial information every fiscal period starting from the company's original incorporation. The second problem we contribute to solving is the problem of considering unquantifiable data in our analysis and prediction of stock growth. This is also addressed through the use of natural language processing techniques. We will utilize sentiment analysis to mine opinions from social media to analyze the quality of certain aspects of a company and how the quality affect the overall long-term stock growth.

## Chapter 2

### Information Extraction/Web Scraping

#### 2.1: Basic Linguistics

One cannot understand the concepts of Information Extraction without first having the basic foundations of linguistics, the study of natural language. Linguistics differs among different languages and therefore for the purposes of this research, we will focus on the English language. There exists two components to the structure of a natural language, syntax and semantics. Before one can begin processing these components, one must first process phonological and lexical (morphological) components. The processing of phonological components is the process of analyzing sounds to generate text based on intended meaning. This is beyond the scope of this research since we plan to only deal with text. The lexical stage of natural language processing is finding the most basic form of a word. This will be further discussed in the next section. The syntax of a language describes what is acceptable language structure and what is not. Human beings are born not knowing every single possible sentence that can be generated in a language but we are born with an innate set of rules we can comprehend and understand. Therefore, we can describe the structure of natural language through generative grammars (first developed by Noam Chomsky). A generative grammar is a set of rules which describe the structure of a given language. Noam Chomsky developed a simple grammar called the phrase structure grammar which constructs the structure of natural language through the different parts of speech. Figure 4 [6] shows an example of a phrase structure

grammar for the English language. The symbols represent acronyms of different parts of the English language. We begin with the most general structure of English (the sentence) and define the structure of the sentence until we eventually are defining specific words (terminals). The elements in parenthesis represent elements that are optional. The bar in the first definition of the Verb (V) element is an 'or' statement. Defining a rule that states  $V \rightarrow V_i | V_t + NP | V_c + Adj | V_c + NP$  is an equivalent statement to the three rules

S → NP + VP  
 NP → Mod + N(PP)  
 Mod → (Art) (Adj)  
 VP → V(ADV)  
 VP → Aux + V  
 V →  $V_i | V_t + NP$   
 V →  $V_c + Adj$   
 V →  $V_c + NP$   
 ADV → PP  
 ADV → Adv  
 PP → Prep + NP  
 N → person, building ...  
 Art → a, the, ...  
 Adj → bright, grand, beautiful ...  
 Adv → subtly, slowly, ...  
 Prep → before, on, under, ...  
 $V_i$  → walk, run, ...  
 $V_t$  → kick, brush, ...  
 $V_c$  → was, become, ...  
 Aux → will, can, ...

Figure 4

written in the figure. A more complex grammar (also developed by Noam Chomsky) is called the the transformational generative grammar. The transformational generative grammar takes phrase structures (generated from the phrase structure grammar) and adds transformational components to the sentence which allow sentences to be transformed into different forms (such as declarative statements to interrogative statements, present



tense to future or past tense, active to passive, etc.) with an application of morphophonemic rules to ensure subject verb agreement. Up to this point, the concepts of syntax have been discussed, however, proper English sentences do not entirely depend on pure syntax. The sentence, “The chair bit the song” is a perfect as far as syntax but semantically does not make any sense. We need to be able to derive meaning from sentences. We can use Chomsky’s Standard Theory to take semantical aspects and context into account. The Standard Theory is based upon the concept the deep structure of text is mapped to the surface structure of text through transformational generative grammar. Sentences that are produced by phrase structure rules are known as the deep structure. The surface structure sentence is a sentence that would be used when naturally speaking or writing. The Standard Theory consists of three different components. The first is the Syntactic Component, which contains the phrase structure rules lexicon and transformational component. The phrase structure rules are equivalent to Chomsky’s phrase structure grammar but we now have the addition of the lexicon. The lexicon adds features to words to express appropriate context. Using the example “The chair bit the song”, possible features that would be applicable to the word “bit” would be {Animate+, N+, Physical ...}. These features state the subject that is performing this specific verb needs to have the properties “is Animated” and “is a Noun” and the predicate that holds the same properties. Therefore, it can be said the sentence “The chair bit the song” does not make logical sense because the markers of the word “chair” do not agree with the markers of the word “bit” (The “Animate” feature in is false in “chair” but the markers of “bit” require it to be true). This ensures the language is now case sensitive. The

phonological component will not be considered for this paper because we are only specifically dealing with text. Chomsky's Standard Theory implies sentences have transformational properties, however if the deep structure is the same among two sentences that only hold a transformational difference. We can prove this statement to be false with the simple sentence "Alan sprayed paint on the car" and "The car was sprayed with paint by Alan". These sentences only differ transformationally but have two different meanings. The first sentence implies Alan sprayed paint only partially on the car (it could be interpreted as either partially or fully) and the second sentence implies the entire car was spread with paint by Alan. Therefore, it is prudent we use an enhanced form of Chomsky's Standard Theory that takes into account semantics both in the deep structure as well as the surface structure. We can use Chomsky's Extended Standard Theory to accomplish this. The Extended Standard Theory is relatively the same as the Standard Theory except now the surface structure of the text is translated into a logical form which is then translated into semantic representation. Therefore, semantic representation is being generated from both the surface structure and the deep structure. Enough basic knowledge has been presented where we can now look into the concepts of information extraction.

## 2.2: Information Extraction

Information Extraction is the concept of extracting and structuring data from an unstructured data source. The general architecture and process of Information Extraction

is given in Figure 5[7]. We begin with a document of raw text (usually fetched with a web crawler) and we begin the process of sentence segmentation. Sentence segmentation is the process of generating a list of strings of sentences within the text. We then take this list of strings and begin the process of tokenization. There exists several kinds of tokenization used depending on the needs of the user but generally tokenization is the process of taking the list of sentences and segmenting each word within each sentence. We then end up with a list of lists of strings (which are words). Once we have segmented each word, we then pass the list to a parts of speech tagging function. The parts of speech tagging process is going through each word and classifying each word as a particular part of speech. Some examples may include tagging a phrase such as “the

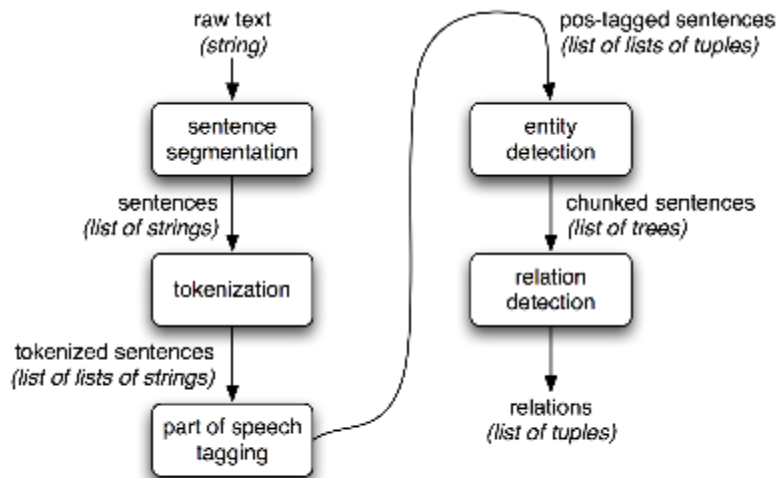


Figure 5

coffee cup” as a noun phrase (usually denoted as NP) or a phrase such as “is working” would be classified as a verb phrase (VP). This function will generate a list of lists of tuples which we then must pass to perform entity detection. The proper name is Named-Entity Recognition assigns each word as a specific type of named entity. A

named entity is defined by user and differs depending on the information that is trying to be extracted. Examples of named entities could include Person, Organization, Animal, Movie, etc. After each word has been properly classified as an entity (and chunked which will be explained in a following section) a list of trees is passed to a relation detection algorithm. The relation detection algorithm will then create a set of tuples that describe relationships between a set of entities. An example of such a relationship might be company A acquired company B which may look something like `Acquired(A, B)`. Before we can describe the two different types of information extraction, we must first explain in detail the logistics of each section and how it works.

### 2.2.1: Sentence Segmentation

One of the main issues with segmenting sentences is ambiguity. A sentence that ends in either a question mark or an exclamation mark is relatively unambiguous. However, a period may be the mark of several things including the mark of a number (such as 0.02), a title (Mr., Ms., Dr.) or may be the mark of the end of a sentence. We must therefore construct a binary classifier that, once it comes upon a period, is able to decide if we are at the end of a sentence or not. One possible algorithm we can use to determine whether or not we are at the end of a sentence in a given text is a simple decision tree algorithm. Therefore, it is important to select relevant features that can be used to decide whether or not we have reached the end of a sentence. What features to select depends on whether we are trying to build a rule-based or machine learning based classifier and will be further discussed in detail in the next section.

### 2.2.2: Tokenization

Tokenization is composed of two processes, segmenting words in a given text and then normalizing word formats. In the task of word normalization, we must take each word and reduce it down to its most basic form. Words can have different word forms such as singular and plural forms of words but we must be able to recognize words that have similar lemmas. Words that have similar lemmas are words that are of the same part of speech and have a similar stem (an example of words with the same lemma would be “guys” and “guy”). Usually tokenization segments based on the punctuation which means we may run into problems with punctuations that occur within words. For example, words that include an apostrophe might be separated into two different words (“Phil’s cat” will be segmented as “Phil, “s”, “cat”). Another example would be segmenting the period within acronyms may cause several issues. We can fix these issues by performing word normalization. Word normalization ensures that all words within the given space are in the same form. One of the most common techniques of normalizing words is reducing all characters in the text to lowercase. We do this because most characters and words are in lower case. This technique is called Case-Folding. However, it would be in our best interest to keep uppercase words that appear in the middle of a sentence because it reduce ambiguity. An example of this would be distinguishing between the words “Fed” and “fed” where “Fed” implies the Federal Reserve and the word “fed” is the past tense of feed. Also we would want to keep the names of organizations and people uppercase because this may help us with their

classification in a later stage. Another technique of normalization is called Lemmatization. Lemmatization is the task of reducing every word in the text to its base form. Therefore, words such as “character”, “characters”, “character’s”, and “characters” would be reduced to the word “character”. Another case that is not as obvious to reduce are words such as “am”, “is”, and “are” which are mapped to the word “be”. The other method of reducing words down to their base form is Stemming. Stemming is based off of Morphology which is the study of Morphemes. Morphemes are the different basic parts that make up a word. The two main parts of a word are the stem and affixes. The stem of a word is the unit that holds the main meaning of the word whereas the affixes are grammatical components we add to create meaning (the substring “es” in the word “affixes” is an “affix” where “affix” is the stem). The objective of stemming is to remove all affixes from words in the text to reduce them to their basic form. One of the most used stemming algorithms is Porter’s Algorithm. The Porter’s Algorithm is a rule-based algorithm that iterates through different steps and eliminates affixes on each step. The first set of rules state all characters “sses” map to “ss”, “ies” maps to “i”, and “s” maps to  $\emptyset$  (the empty set). In the next set of rules, the objective is to delete all appropriate “ing” and “ed” substrings. An example of an inappropriate word for deleting “ing” would be “bring”. Therefore, the rule states we must map all “ing” and “ed” substrings to  $\emptyset$  if the substring is preceded by a vowel. In this case, a word such as “crossing” would be mapped to “cross” because of the ‘o’ character. In the third step, we are removing affixes from longer stems. The rules of the step three are substrings “ational” map to “ate”, “izer” maps to “ize”, and “ator” maps to “ate”. The fourth and

final step of Porter's algorithm is the removal of affixes on the longest stems. Therefore, the rules are substrings "al" maps to  $\emptyset$ , "able" maps to  $\emptyset$ , and "ate" maps to  $\emptyset$ . The Porter's Algorithm is only appropriate for the English language. We would need more complicated stemming algorithms to reduce words in other languages such as German or Chinese. These are the most common techniques of word normalization. Once we normalize all words by removing ambiguous punctuation and reducing words to their basic stem, we can segment words within the given sentences and create a list of lists of word strings ready to be tagged in a parts of speech tagger.

### 2.2.3 Parts of Speech Tagger

The definition of a "part of speech" is a class of words which hold similar grammatical properties. The parts of speech tagger will be highly, if not solely, dependent on what is defined in the grammar (an example of such a grammar is given in Figure 5.). When defining the grammar of a given natural language, it is important to make as many distinctive classes of words as possible. The more classes used in defining the lexicon of the language, the easier it will be to define and extract relational information from the text. We can begin by classifying words into two different categories. The first class of words is open class words (lexical words). Open class words are words which belong to a class which is continuously being updated with words as time moves forward. Examples of open class words would be nouns, proper nouns, and adjectives. The second class of words are closed class words. Closed class words are words which belong to a class of words which do not obtain new words through the

passage of time. Examples of such classes are pronouns, prepositions, conjunctions, and modal verbs. Classifying words that belong to a unique class of words are simple to classify, however, there exists words in the English language that belong to multiple classes of words (i.e the word “hit” is both a verb and a noun) which bring ambiguity to classifying the words. Ambiguity does not pose a major issue considering the fact there are only a small percentage of ambiguous words in the English Corpus. There are several methods that can diminish error in parts-of-speech tagging. These methods will be discussed in the next section when discussing rule-based natural language versus machine learning natural language.

#### 2.2.4: Entity Detection

Entity Detection is the process of finding entities and classifying text. An entity is a discrete object in the world such as a specific person or school but things such as sand or air are not specific objects and are not entities. One method for detecting entities is called chunking. The process of chunking involves taking the parts of speech tagged tokens and grouping them into labels such as noun phrases and verb phrases. The first type of chunking we can obtain is noun-phrase chunking. Noun-phrase chunks do not necessarily have to be a single noun-phrase. Noun-phrase chunks can have multiple nested noun-phrases within them. We do this for the purpose of avoiding having



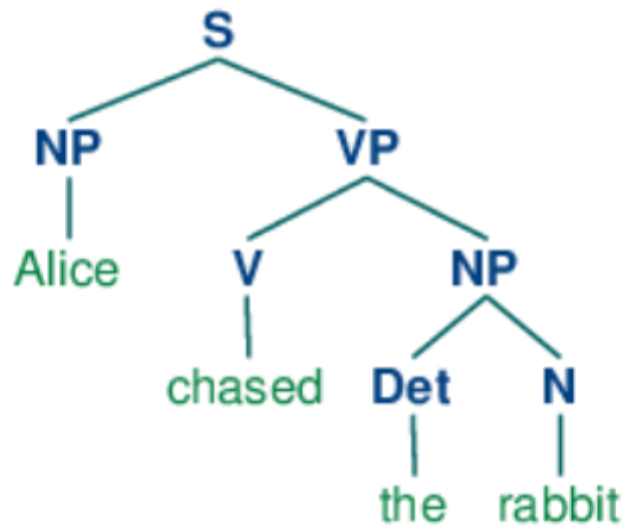


Figure 6

noun-phrase chunks within noun-phrase chunks. We do the same for verb phrase chunks and others we intend to search for. We then will need to refer to the grammar we have previously created to generate rules which classify different types of chunked phrases. For the purposes of finding specific entities such as revenues and profits, it would be prudent to include a name-entity recognition system. For a named-entity recognition system, it is required we label words in the text in IBO (inner, beginning, outer) format. A word that begins a chunk is tagged as 'B' and a tag occurring in the same chunk is labeled as 'I'. Once we have labeled these words, it is necessary to use a trained classifier to classify text as specific Named-Entities. Once we have our entities labeled as well as our named entities, we then pass the labeled trees (as seen in Figure 6) to a relation detector.

### 2.2.5: Relation Detection

In the relation detection, or relation extraction, process the goal is to find tuples of relations among different entities. This is why it is important we have named-entities available to us from the previous process. We can find relations through rules or supervised learning algorithms as we could with the previous processes. However, we can also extract these relations through Ontological Engineering. Ontological Engineering is the process of representing abstract data in the form of graphs. An example of a type of relation would be the IS-A relationship. We can mine for copulative verbs to find IS-A relationships between different named entities.

### 2.3 Rule Versus Pattern Implementation

In the previous section, there were a number of processes mentioned in which both a rule-based and supervised learning approach could be implemented. When applying a rule-based solution in natural language processing (or formal language processing), we implement rules through the use of regular expressions. The regular expression is a tool used for processing text strings. There are many functions inherent in regular expressions that make it simple to identify properties in text we are specifically looking for. An example of a regular expression would be “[A-Z]?[a-z]\*(\.)[“”][A-Z]”[8]. The brackets are symbols that represent disjunction. The section “[A-Z]” is stating only to find capital alphanumeric characters. The question mark occurring subsequently after the phrase is implying the precedent phrase is only optional. We then see a similar phrase in brackets as seen in the first part of the regular expression. This section of the expression is simply stating to find a lowercase alphanumeric characters.

The asterisk is a symbol that states the preceding element can occur zero to infinite number of times. The last elements of the regular expression are implying to find strings which have a period, a space, and a capital alphanumeric character. Using this regular expression, we can find all strings in the text which are title specifiers (Mr., Mrs., Dr., etc.). If we eliminate these types of strings, we eliminate a large majority of ambiguity when segmenting sentences within text. In the parts-of-speech tagging process, a rule-based approach would be relying solely on the grammar to handle ambiguous words. We can minimize the error of incorrectly classifying text by constructing a more complex grammar. Another rule-based approach that can be implemented are semantic networks. Semantic networks are a form of knowledge representation in which semantic relations are matched between concepts. We can implement these semantic relations to properly classify ambiguous words based on their contextual surrounding. Lets take the example sentence "John is back". The word "back" is an ambiguous word that has many different meanings and different grammar classifications. However, using a semantic network, we can deduce the true intended meaning of the word. The sentence represents an IS-A relationship where John is back. However, we can deduce "back" cannot be represented as a noun or an adjective since it is impossible for an animate noun to be a noun or the description of a noun that is not animate. Also, "back" cannot be a verb because a noun cannot also be a verb. Therefore, the only classification available for the word "back" would be an adverb describing the state-of-being verb "is". We can, therefore, implement a highly accurate rule-based system to classify parts-of-speech. These methods are similar in a rule-based relation extraction system. In pattern recognition, the

objective is to find the relation between entities X and Y. Different textual patterns can be used to extract the IS-A relation between two entities. Using Marti Hearst's pattern algorithm, we can find IS-A relationships between entities. These textual patterns include "Y such as X", "X or other Y", "X and other Y", "Y including X", and "Y, especially X". The IS-A relationship is very useful when classifying a specific text as a company aspect of interest. We can also, use named-entities mixed with word classes to construct patterns which extract relation between entities. Therefore, a rule-based approach would be to construct a set of rules such as "Revenue X" or "Profit X" when the income statement states "Revenue" in a single table then a number in an adjacent table or variations of this pattern. When using a rule-based system, we usually have a high precision. However, the patterns created are more than likely tailored only to a specific domain. Applying this approach to a large scale domain would result in high inaccuracy. The number of patterns that would need to be created is exponentially high. This would also imply using a rule-based system works well with websites which already have a semi-structure of information built into the system. The alternative to rule-based systems are supervised learning and probabilistic modeling algorithms. The objective is to use trained classifiers that are able to classify text and other information without having to write explicit patterns and rules for governance. In the process of sentence segmentation, the machine learning algorithm logistic regression is very useful. As said previously, the primary issue when segmenting sentences is segmenting the ambiguous periods between real sentences and words which require the use of a period. We would first construct a training set of segmented sentences along with non-segmented sections

of sentences. It is critical to select proper features which would be used to classify the end of a sentence or not. Example features would be the word class before the period, the word class after the period, the capitalization of the word before the period, the number of spaces after the period, and number of characters in the word before the period. To ensure the features selected are the most relevant, we would run a leave-one-out cross-validation algorithm on the classifier. Once the classifier has fully run through the training data, we would then test the classifier on a test set. The objective is to train a classifier with minimal test error without overfitting the data. When classifying text, there are two different models that can be used, joint generative models and discriminative conditional models. Similar methods can be applied in the process in tagging text as a part-of-speech. When tagging parts-of-speech, we can use the words themselves and different properties of the words such as knowing the case of the word, the prefixes, the suffixes, and the word shape. To this point we have mostly discussed joint generative models. These types of models place probabilities on both seen data and unseen (hidden) data. Examples of joint generative models would be n-gram models, Naive Bayes Classifier, hidden Markov Models, and probabilistic context-free grammars. Discriminative conditional models are the alternative to using joint generative models. Discriminative conditional models take the data as it is given and puts a probability over the hidden structure of the data. Examples of discriminative conditional models would be logistic regression models, conditional loglinear or maximum entropy models, conditional random fields, and support vector machines. Generative models seek to maximize a joint probability whereas the objective for discriminative conditional models

is to maximize conditional likelihood. Conditional models have been known to be more effective in accurately classifying word sense disambiguation and can be very useful in classifying revenues and profits (named-entities). In our particular research, discriminative conditional models can be used to classify fundamental information. The first task would be to create a text corpus of income statements and fundamental information mined. We then construct features that can classify an entity as a parameter of interest. Constructing features is similar to constructing definite rules only now we must assign a maximum probability to this rule. An example feature would be  $f(c = \text{“REVENUE”} \wedge w_{i+1} = \text{“20”} \wedge \text{isRevenue}(w_i))$ . This feature is stating an entity classified as Revenue will be preceded by the word “Revenue” and preceded by the word “has”. Therefore a text stating “Profit” would classify the next segment as a profit number because it follows the constructed rules of the feature. However, it is the objective to be able to construct this feature without explicitly programming it in. We first would create a training set from the collected text corpus and classify text as revenues. We would then train a classifier with the training set to generate the feature rule. If the above feature is labeled to probabilistically be accurate 89% of the time, then we can say with confidence the feature is accurate. We therefore can accurately label named-entities. Once named-entities have been identified, we could use supervised-learning methods to extract relations among the entities. We must first choose a set of relations we want to extract. We then must find a relevant set of named entities. The named-entities we are interested in are profits, dividends, and revenues. We then must find the labeled data. We must first take our corpus and label the relations between the relevant named-entities. The set

we create can then be split into training set, development set, and test set and we can train a classifier to the training set. In addition to this, we must find pairs of named entities in the same sentence. If the two entities are related, then we classify the relation. The features used for finding relationships are similar to the ones used in naming entities. These features include parts-of-speech tagging, word features, and particular words which appear in the sentence. We then can create a list of tuples that represent a list of named entities.

### 2.3.1: Choice of Structure

It is generally the case where supervised learning methods of natural language processing yield better accuracy than rule-based approaches. However, for this research, the data mining will be limited to a single website. Also, the website which we will mine already holds an inherent semi-structure. In more detail, the United States Securities and Exchange Commission's website holds 10-K forms (labeled as such). Each form holds all information within tables. Therefore, if we write a set of rules to extract the tables and write a grammar which can extract information from the tables, then we can build our dataset with high accuracy.

## Chapter 3

### Neural Networks

#### 3.1: Introduction

For the prediction of whether or not the company of interest is a worthy investment, we will use neural networks. Artificial neural networks are a paradigm model which foundations are based upon the principles and biological makeup of the human brain. We can use the neural networks model to cater to our application and find patterns within the data set for determination of growth[9]. Like other machine learning algorithms, we can input the training set to properly train the artificial neural network. One of the main reasons we decided to use artificial neural networks is the paradigm's ability to not just find patterns within abstract data, but to also find trends within data which is precisely what we are looking for. The advantages of using neural networks are the learning aspect and being able to adapt to perform certain tasks. Artificial neural networks can also create organizations and representations of information it is given over time. In other words, as time continues, we can keep a constant input of data and the neural network will adapt to new situations and new information. Lastly, if need be, multiple machines may act as a single artificial neural network to improve computational performance in certain situations. In this chapter, we will discuss how the artificial neural network is built and how it is used within this research.



### 3.2: Human Brain and Computational Machines

Artificial neural networks behave differently than the traditional algorithmic approaches of a computer. A regular computer will solve a particular problem by a series of steps which are specified by the programmer. However, this approach can lead to problems which cannot be solved algorithmically. Even problems such as reading and understanding text which seems to be a simple problem to humans can be a near impossible problem for computational machines. This is because our brains learn differently from traditional algorithmic methodology. Our brains are composed of

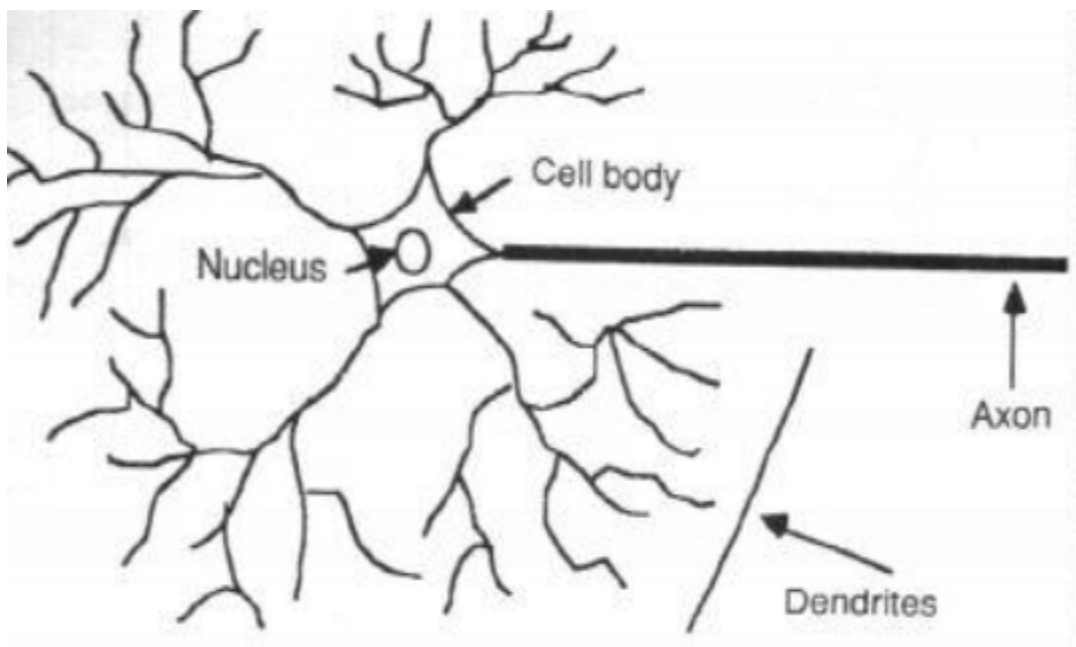


Figure 7

billions of neurons which are meant to process information. As seen in Figure 7, we can see all the basic components which make up a single neuron. The main components of a

neuron are the nucleus (composed within the cell body of the neuron), the dendrites, and the axon. Each component holds a specific purpose in the computational process. First, the neuron needs to receive some form of signal or stimulus to activate. This is done through the dendrites. The dendrites are a series of thousands of branches which collect an electric stimulus from another neuron. This stimulus is passed through the dendrites and into the cell body and nucleus. The cell body will then create a computational stimulus of its own based on the electric input from the dendrites. From the cell body, a signal will be sent through the a component of the neuron called the axon. The purpose of the axon is to send a signal from the cell body to other neurons within the network.

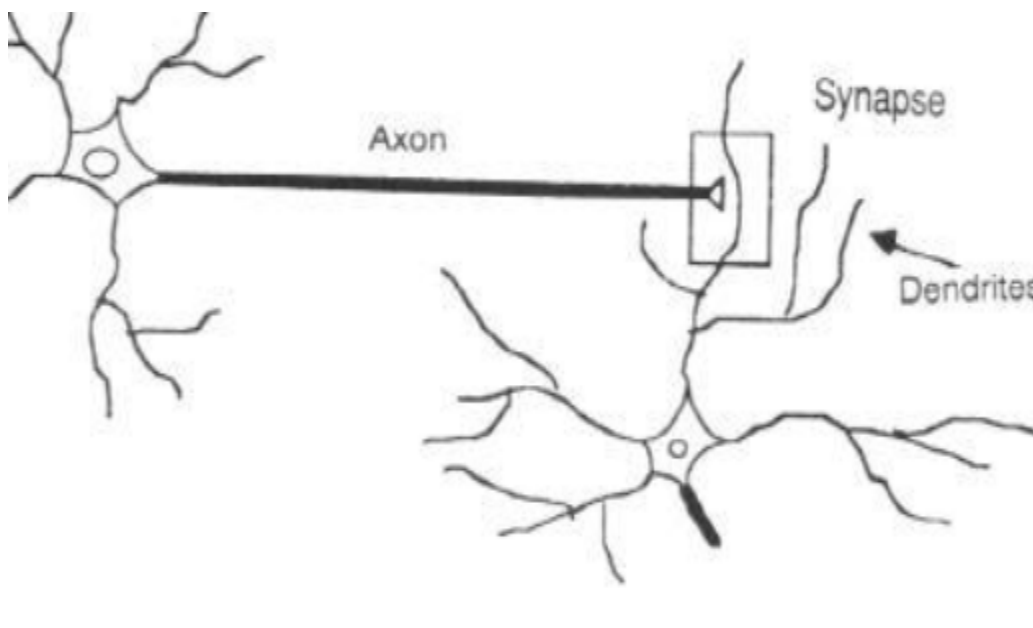


Figure 8

When a signal is passed from one neuron to another, the electric stimulus must be passed through a component called the synapse which can be seen in Figure 8. The synapse is

one of the most important components of the neuron because this is where learning occurs. The synapse resides at the ends of the dendrite branches of another neuron. The synapse will take the signal from the axon of the activating neuron and create electrical impulse which will be used to create computational signals for the neuron of the receiving dendrites. The size of excitement stimuli within the neuron is dependent upon the size of electrical impulse created within the synapse. Therefore, the learning aspect of a neuron is implemented through the synapse by altering the amount of stimuli which is sent through a neuron based upon the input the dendrites receive from another axon. This process of learning and algorithmic representation is what differentiates neural networks and traditional computational methods. The neural network (or artificial neural network in our case) is a connection of many of these computational elements all working together to solve a single problem or performing a certain tasks whereas a computational computer takes more of a step-by-step process to solve a given problem. This raises both pros as well as cons for each method of computation. One of the greater benefits of working with neural networks is the paradigm's ability to solve highly complex problems. For a computational machine, reading a simple text message such as "Hello World" is nearly impossible to understand. But using a neural network, we are able to use many features to properly learn the structure of each letter and the what the combination of letters and words mean. This in itself creates certain problems when you have a complex paradigm such as neural networks. Due to the number of computational factors and high number of simultaneous workings of different components, the outcomes and operations of certain tasks can be unpredictable. This unpredictability will make it

difficult to find any errors within the processes of computation. Also, unlike the traditional computational machine, the neural network learns by example. Therefore, if we obtain bad examples within the learning process, then it makes it difficult to correct the networks learning and may lead to skewed results. With computational machines, we take a more systematic approach to solving different problems. This allows these type of paradigms to be very predictable in their operations. However, opposite of the neural network, the traditional computational machine does not learn from example but rather needs to be told the problem which presents obvious problems of its own. This is why it is difficult for traditional computational machines to perform certain complex tasks because it is near impossible for a programmer to create an algorithm which covers all tasks for a machine to perform complex operations correctly. However, neural networks and traditional computational paradigms can be used together to create efficient learning systems. We can use the neural networks to represent and classify information whose learning methods are supervised by traditional computational paradigms. In the following section, we will show how artificial neural networks are built and how they take advantage of both the traditional computational paradigm and the neural network paradigm used within the biological brain.

### 3.3: Artificial Neural Network

When we approach the creation of these neural networks, we must keep in mind all the components mentioned within the previous section. One of the general

conceptions of the neuron is the ability to know when or not a neuron should fire a stimuli. The neuron fires based upon the input from other neurons. Therefore, we must design our artificial neuron in a way where the outcome is dependent on the inputs we get. Our algorithmic model must have a set of inputs which will act as our dendrites. From the dendrites, the input must be processed within our artificial cell body. From the cell body, we must then be able to transfer the our computation to another artificial neuron where this same process can be implemented. This model can be seen on in Figure 9. The neural network is created through layers of nodes. As mentioned in a previous section, the artificial neural network is comprised of layers. All neural networks have a set of input layers and output layers. However, an artificial neural network may or may not be comprised of what is known as hidden layers. Hidden layers are layers of nodes which are inputted information in between the input layer and the output layer. Figure 10 shows a node with no hidden layers. The layers are simply input layers and output layers. The nodes in between input layers and output layers are layers which obtain input from the preceding layers (whether it be the input layer or another hidden layer) and outputs to either the output layer or another hidden layer. The nodes within the hidden layer (much like the output layer) have an activation sequence of their own. The activation of each node is

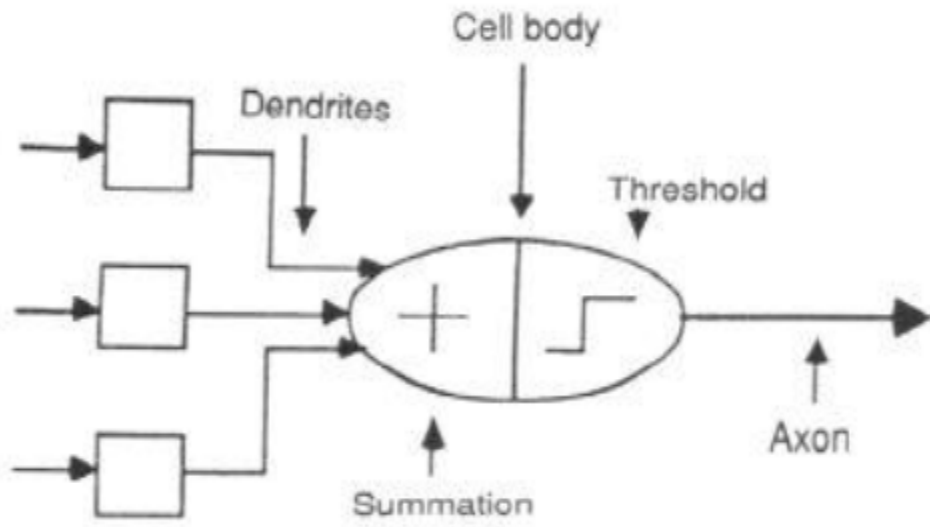


Figure 9

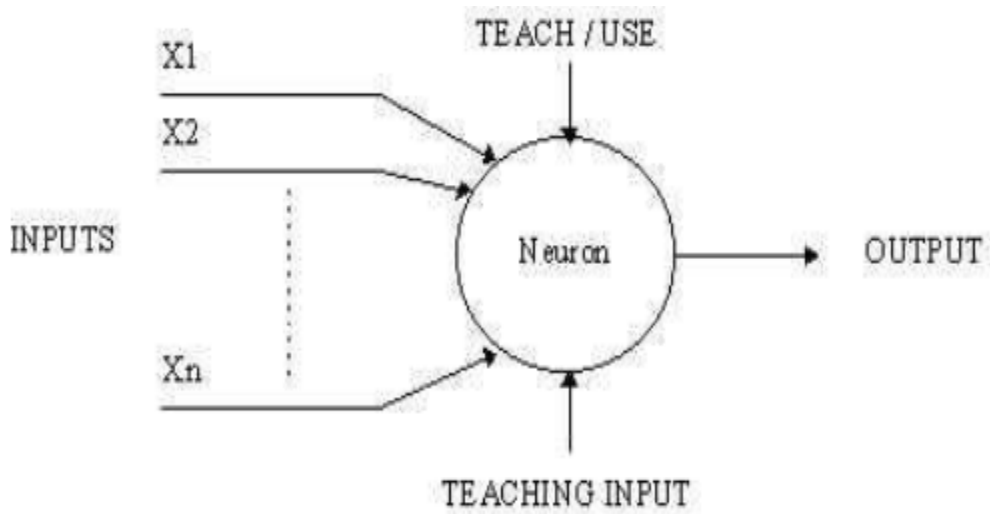


Figure 10

dependent upon a certain threshold. This threshold can be determined in many different ways. However, for our specific case, the threshold will be determined by a logistic regression function. In the following section, we will discuss how a logistic regression function works.

### 3.4: Logistic Regression

Logistic regression is a machine learning algorithm which is used to classify data. The logistic regression algorithm is a methodology which creates a probability of a classification given a certain set of parameters. The algorithmic function which determines the probabilistic situation is called the sigmoid function. The sigmoid function for logistic regression is given as follows [10]:

$$G(z) = \frac{1}{1+e^{-z}} \quad (2)$$

This equation will give us a probabilistic model representation of what a set of data of interest will classify as our true classification. In other words, if we are looking to determine if our data supports the classification of one, our equation  $G(z)$  will give us the probability that this is the case. Therefore, it is common sense to say  $1-G(z)$  will give us the probability where the data of interest is a false classification. It is also very common to see the  $G(Z)$  function be represented as follows:

$$G(Z) = \frac{e^z}{e^z+1} \quad (3)$$

This given equation will output the a similar graph to the following[11]:

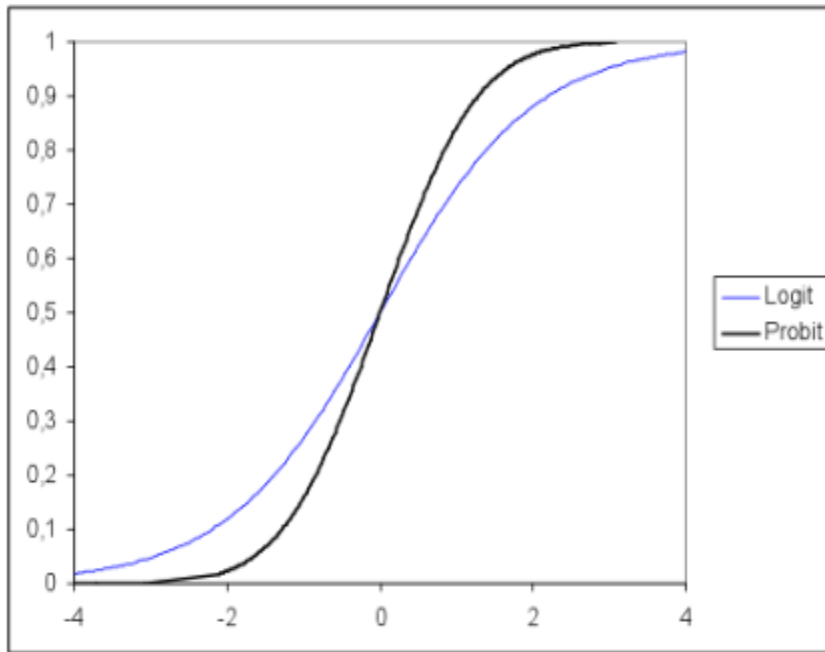


Figure 11

The limits to negative infinity approach zero while the limits to positive infinity approach one where  $G(Z)$  represents the y-axis and  $Z$  represents the x-axis. Therefore our probabilistic values within the logistic function will be dependent on our  $Z$ . The  $Z$  is simply the summation of feature values and their associated weights.

$$Z = \sum_{i=1} w * x_i \quad (4)$$

It is usually the case where this equation is represented within vector format. We represent a vector of weights and a vector of input values.

$$Z = w^T * x_i \quad (5)$$



Therefore, there exists many different ways which we can use to train this function. In the following section we will see how we train artificial neural networks using these models and how this will be utilized within this research.

### 3.5: Artificial Neural Network Learning

Now that we have discussed how the threshold for each node is computed, we will now discuss our learning algorithm to train our network to output the most accurate result of worthy investments and unworthy investments. For our purposes, we are going to use the backpropagation algorithm to train our learner to the dataset. Our first task is to create a proper error function to represent the error in our model. If we label our output vector  $O$  and our training vector  $Y$  we can represent our error function as the following:

$$error = \frac{1}{2} \sum (O_j - t_j)^2 \quad (6)$$

This equation is simply indicating the error between our output layer and our target values. To minimize this equation, we must take a derivative with respect to the weights which were used to transition from the previous layer to the output layer. Therefore, assuming  $W$  represents the weights which were used in transitioning to the output layer, our process would be as such. First we must take a derivative of our error function with respect to our weights  $W$ :

$$\frac{d}{dW} error = (O_j - t_j) * \frac{d}{dW} O_j \quad (7)$$

As one can see, we have used the chain rule to derive to the point where we must now derive our output layer threshold function. If we represent  $f(x) = 1 / (1 + e^{-x}) = O_j$ , then we can say the derivative of  $f(x)$  equals:

$$\frac{d}{dW}f(x) = f(x) * (1 - f(x)) * \frac{d}{dW}x \quad (8)$$

In this case we know  $x$  represents the input to the output layer  $O_j$  multiplied by the weights  $W$ . Therefore,  $dx / dW$  is simply equivalent to the hidden layer which acts as input to the output layer  $j$   $O_{j-1}$ . Therefore our derivative of our error is:

$$\frac{d}{dW}error = (O_j - t_j) * O_j * (1 - O_j) * O_{j-1} \quad (9)$$

This has been the process of finding the error rate for the output layers with respect to the weights of the previous layer nodes. We must now find the error of the hidden layer nodes with respect to the weights given in the transfer function to the current hidden layer. In order to do this we must again look at the same error equation (6). Our operations are the same until we reach equation (8). Because “ $x$ ” represents the input to the output layer, we cannot simply derive  $dx / dW$  because we are now taking the derivative with respect to the weights of the hidden layer before the output layer. Therefore, using the chain rule, we can deduce the following equation is true:

$$\frac{dx}{dW} = \frac{dx}{dO_{j-1}} * \frac{dO_{j-1}}{dW} \quad (10)$$

This equation is not only true because of the chain rule, but we can also logically come to the conclusion that the input “ $x$ ” (which as a reminder represents the input to our output layer  $O_j$ ) has a direct dependence on the output which comes from the layer before it, whether it be the input layer or another hidden layer. Because we know the hidden layer output  $O_{j-1}$  is simply the input “ $x$ ” multiplied by the weights used in the transfer

function between the hidden layer and the output layer,  $dx / dO_{j-1}$  is simply the weights from the hidden layer to the output layer (represented as  $W_j$ ). Now our final task is to solve  $dO_{j-1} / dW$  where  $W$  now represents the weights used for the transfer function from the layer before the hidden layer to the hidden layer itself. This derivative looks identical to equation (8) only now we are dealing with the layer which precedes the output layer. For the sake of simplicity, we will define  $(O_j - t_j) * O_j * (1 - O_j) * W_j$  as  $\Delta J$ . Therefore, our full error change equation for the hidden layer before the output layer would be given as follows:

$$\frac{dE}{dW} = \Delta J * O_{j-1} * (1 - O_{j-1}) * O_{j-2} \quad (11)$$

These steps would be performed for all hidden layers which implies the more hidden layers our neural network is composed of, the more complex our training algorithm will be. The training of neural networks is an iterative process which requires a great amount of data to produce acceptable accuracies.

### 3.6: Literary Conclusion

This is the foundational information and research needed in order to conduct this research. We have discussed the different methodologies of analyzing stocks and discussed the basics of how to conduct a fundamental analysis. We also discussed the different natural language processing concepts needed to fulfill our qualitative analysis for stocks. We lastly discussed the foundational principles for neural networks and how to train our artificial neural network using the backpropagation algorithm to optimize accuracy in the validation stage. In the following chapter, we will discuss the conducted

research and then show the analysis of our results. The steps to create our research is to first collect our dataset for our selected stocks. This requires us to first mine information of the United States Securities and Exchange Commission website. We then must mine sentiment from a collection of tweets within a given time period. Lastly, we must run our created dataset through a neural networks algorithm and train for high accuracy.

## Chapter 4

### Neural Networks Sentiment Architecture

#### 4.1 Related Research- Time Variant

There have been a variety of research on the usage of neural networks with the prediction of the stock market. One of the prevalent factors of financial markets is there consistency of change over time. The overall markets and individual corporation stocks are prone to behave differently as time goes on. For example, the Apple corporation behaved much differently in the early and middle 1990's versus 2005 to 2010. This presents a problem for the reason if we use neural networks, the inconsistencies of the data we collect can significantly diminish the accuracy of our results. One solution was presented by Qiang Ye', Bing Liang, and Yijun Li'[13]. The solution was to add an amnesic module to the neural network. The term amnetic is a word which means a module which dilutes the memory of far past data. The way we do this is by weighing all non-current data with lighter weights and more current data with heavier weights. This was done for the reason of forgetting non-relevant customer habits and effectively capturing more recent market reactions. As a result, this testing result of maximum accuracy of 58.25%. For this research, we conducted our data collection in a different manner. It was a goal to instead of collect a mass of data which may or may not be currently relevant, we devise a different methodology of collect more relevant data. For artificial neural networks, obtaining accurate data is a high priority due to the fact that irrelevant, and inaccurate, training data can significantly alter results. When we discuss

our system architecture in a later section, we will show an alternative to diluting irrelevant data.

#### 4.2 Related Research- Multi-Agent Architecture

As described in the previous sections, data collection poses a problem due to the nature of changing markets and habits. We either have the option of collecting less mass data with the inclusion of less relevant data or we must find a way to maximize the amount of relevant data we collect. Gabriel Iuhasz, Monica Tirea and Viorel Negru [14] used fundamental analysis data to make predictions about the stock market and certain companies. An approach was taken to write a program (utilizing the neural networks learning algorithm) as a set of multiple agents. There was a given set of coordinator agents which handled the coordination. We then have fundamental, technical, liquidity, and projection agents which all uniquely either calculate different pieces of data and also the projection and coordination of the data generated. For a research purposes, we will use a similar architecture for a variety of reasons. By using this archetype, we can build off the concepts described in chapter two to create agents specifically designed for collecting data. One solution to the limit of data collection and the time variability problem is to mine information from certain information and structure the data in such a way where we are collecting more relevant data but at the same time maximizing the amount of data we are able to collect. Using different agents helps organize and ensure each segment of code is tasked with one specific task. By using a similar architecture, we

can ensure we are collecting the different types of data needed in order to properly test the results and effects of the data collected.

#### 4.3: Related Research- Type of Data

One of the commonalities between the different researches conducted of how to use neural networks to predict the stock market is the type of data which is used as input. In a research conducted by Nguyen Lu Dang Khoa, Kazutoshi Sakakibara and Ikuko Nishikawa [15] using neural networks to predict market trends and time-series, it was suggested to have a mixed input set between technical factors and fundamental factors. This has been a strategy used by all related researches analyzed. For this particular research, we decided to only input fundamental data. However, the fundamental information used by these researches are all inherently quantifiable. One of the main aspects of our research is to analyze the possibility of using sentiment data in prediction models, specifically using neural networks. In the following section, we will show and thoroughly explain how we designed our system architecture.

#### 4.4: System Architecture

In a related research and mentioned previously, we were presented with the idea of having an agent based architecture. In the following figure, we can see an architectural design which was implemented in the research mentioned in section 4.2. We will use this implementation to create our own architecture (Figure 13) for just strictly using fundamental data.

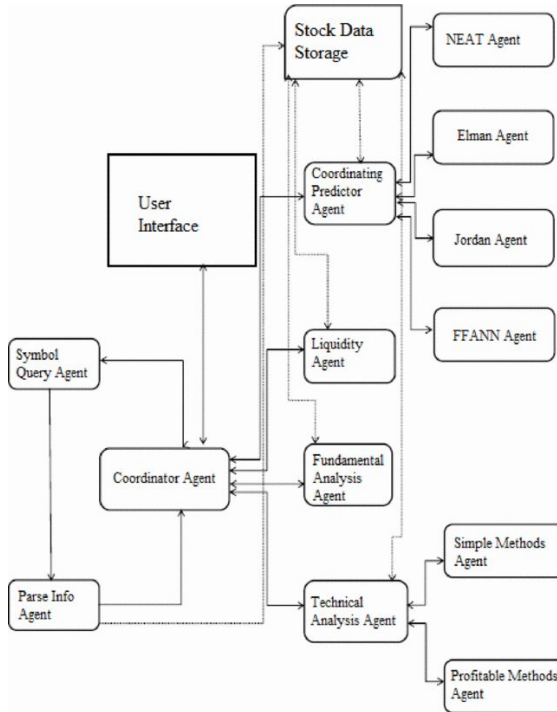


Figure 12

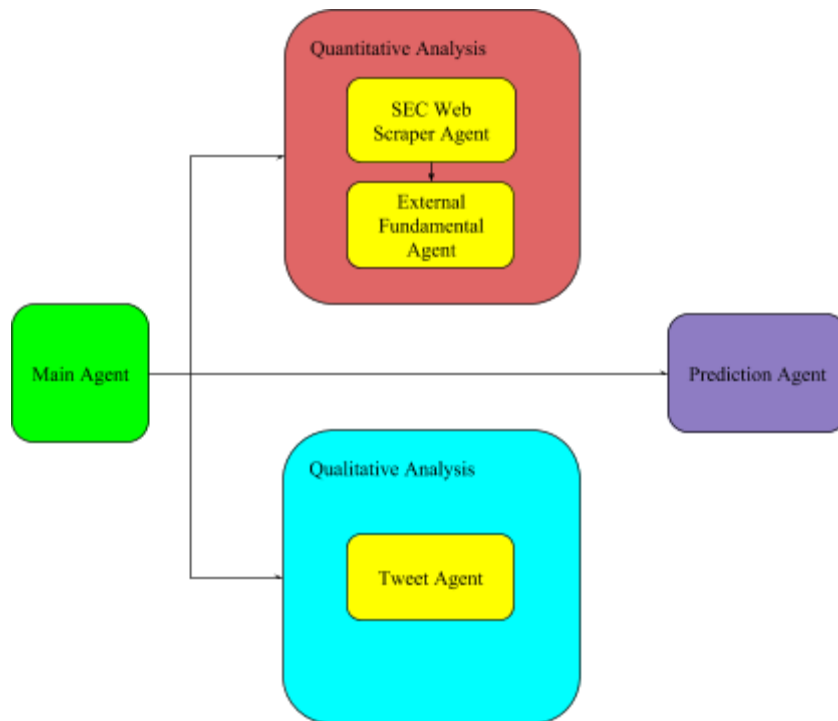


Figure 13



#### 4.4.1: Main Agent

The main agent within this software architecture is the beginning, the middle, and end of all operations. The main agent is responsible for coordinating information between agents of quantitative analysis, qualitative analysis, and the prediction agent. In more detail, the main agent will coordinate three segments of data. First, the agent will obtain the data from the quantitative agents (which are all passed to the main agent at the same time). The agent of qualitative analysis will need to the dates of the data collected by the agents of quantitative analysis. Therefore, the main agent must pass the data from the quantification agents to the qualitative agent. Once the qualitative agent returns data to the main agent, the main agent will pass the data to the prediction agent.

#### 4.4.2: SEC Web Scraper Agent

The SEC Web Scraper agent is one of the two agents of quantitative analysis and is responsible for collecting data from the securities and exchange commission website. As mentioned earlier, one of the problems faced within this research is the limited relevant structured data which is available to us. Therefore, we must employ the natural language processing strategies and techniques described in chapter two to properly mine unstructured data and create structured data which can be used within this program. As mentioned previously, we were choiced with two different archetypes for creating our web scraper, supervised-learning or rule-based. Because we are only planning to mine information from the SEC website and accuracy is a high priority for our data, we decided to implement a rule-based web scraper. The web scraper was built on a set of

grammar rules catered to the SEC website. In other words, we were able to write a set of rules for the html code for the website to mine the information we needed. The steps of implementation for the SEC Web Scraper are given as follows. First, the main agent will pass the company ticker symbol to the SEC Agent which is then mapped to a cik number. (A cik number is a unique number which represents corporation in the SEC website).

The cik number is then used to take the webscraper to a site which lists all the company's posted 10-Q statements. A 10-Q statement is a quarterly posted statement showing different financials of the given company including the revenues and profits which is what we are interested in mining. Therefore, using a set of html grammar rules, the web scraper will mine all 10-Q statements which the company posted all the way back to 2006. For each of these 10-Q statements, an associated date is also displayed for when the statement was posted. The agent mined this information as well along with each 10-Q statement. Lastly, the SEC Web Scraper Agent used another set of html rules to mine the revenues and profits from each of the 10-Q statements. The agent then uses this data along with the dates collected to compute the growth over the given years. The profit and revenue data is then stored until we return to the main agent while the associated dates are passed to the external fundamental agent. The SEC Web Scraper solves the time-variability problem mentioned in section 4.1 by taking each year and creating four annual growths for each year by utilizing the fact that a company must post 10-Q statements four times in a given year. We then have a maximum amount of relevant data which we can use instead of altering our neural networks algorithm in our prediction agent.

#### 4.4.3: External Fundamental Agent

The main purpose of the external fundamental agent is to find all external quantifiable data which relates to the given company. First, the external fundamental agent uses a yahoo website to connect and download a chart of annual dividend yields dating all the way back to 2006. The agent then continues to collect data for the stock price and overall S&P 500 index using a yahoo library. The external fundamental agent then takes the associated 10-Q dates given by the SEC Web Scraper Agent and uses the stock, market, and dividend data to iteratively find the annual growth of each item dating one year back from each date. We then find the future year growth of a given stock stock price for each date as well. Once we have this information, it is passed to the main agent along with the dates and previously collected data to start computing the qualitative sentiment based data in the qualitative agent.

#### 4.4.4: Tweet Agent

The tweet agent is the only agent used for qualitative analysis. The responsibilities of this agent are collecting all the tweets required, creating a sentiment score from the tweets, and then sending the sentiment score back to the main agent for computation. First, the tweet agent receives the post dates of the 10-Q statements transferred from the quantitative agents to the main agents to the tweet agent. Once these dates are successfully transferred to the tweet agent, the tweet agent then parses a structured text file to receive several bits of information. The tweet agent will first collect the company name and CEO information dating back to 2006. The dates of the

active CEOs are also stored with the CEO names themselves. The last information collected is the names of products the company either sells or has association with. With these terms collected, the agent begins the process of collecting tweets for the company overall. This is done by using similar methods as the SEC Web Scraper Agent. The agent first enters the twitter advanced twitter search as a mozilla user agent. Within the url, a query can be made to query a subject along with a date range for when to collect tweets. However, the twitter search will only post an average of nine tweets to the user at a time. Because of this and the fact we do not need to collect all tweets for a given year, the tweet agent resumed to collect nine tweets for every ten days within a given date range. The tweet agent uses a written html grammar to parse and collect the tweets from the collected html. For each date in the listed given from the quantitative agents, we collected an average of 600 tweets dating one year before to the given date. Once the tweets were collected for a given date, a sentiment score was generated from the text using the alchemy api sentiment scorer. These set of steps needed to be done for the company name, each active CEO from 2016 back to 2006, and each associated product of the company. There were several cases where sitting CEOs changed during this time period. When this occurred, the subject of tweets collected was changed to the sitting CEO to ensure what we were collecting was relevant. Also, when collecting for each product, we would take the sentiment for all products and then create an average sentiment score for the products associated with the company of interest. From this information, we were able to generate yearly sentiment scores for the company as a whole, the sitting CEO, and products associated with the company. This information was

then passed back to the main agent where the information was ready to be passed to the prediction agent.

#### 4.4.5: Prediction Agent

The prediction agent is the final agent of action within this program. The main tasks associated with the prediction agent is to use the data collected from the quantitative and qualitative agents to train and tests a neural networks algorithm. First, when the prediction agent receives the full, structured data from the qualitative and quantitative agents, the agent then splits the data in half creating a training set and test set (several tests of different sized training and test sets were made and this gave us the optimal results). Once the training and test sets were created, the prediction agent ran the training set through the neural networks algorithm and trained using a backpropagation learning method described in the previous chapter. After this, the test set was iteratively given to the trained algorithm which calculated a ratio between the number of correctly labeled data versus all data labeled (the percentage number of correctly labeled data). Once this statistic was found, an f-anova test (provided by the sci-kit learning api) was run on both the training set and test set to give a quantified value for how much each data item was contributing to the overall labeling of the data. The agent then presented the final results to the console.

#### 4.5: Conclusion

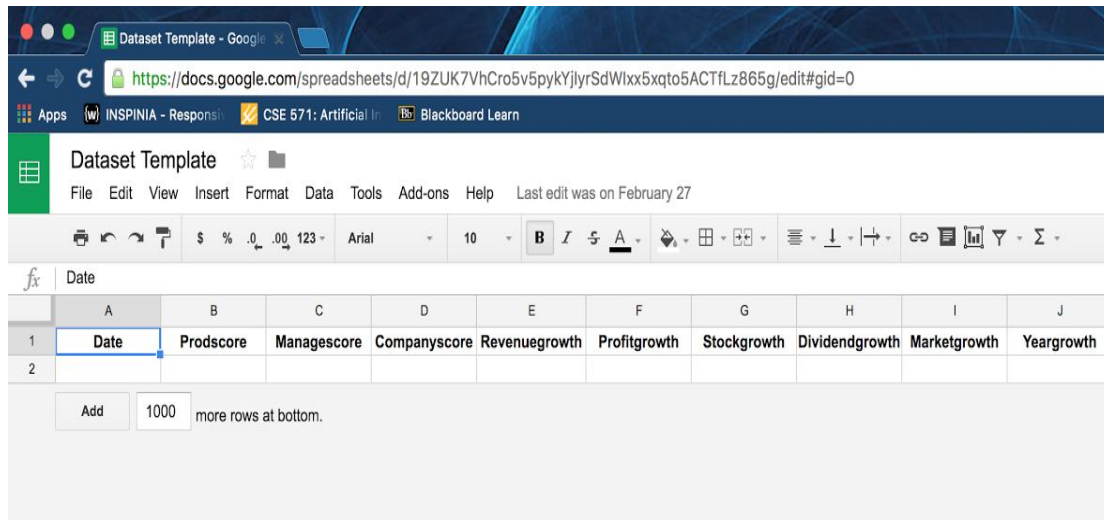
The major difference between this research versus others which have been made is the usage of sentiment in our learning algorithm. In the following chapter, we will discuss the results from our tests and show that sentiment was a prevailing factor in the prediction of these stocks. This proves that sentiment and other machine learning based scoring methods can and should be considered more as the input to other machine learning algorithms.

## Chapter 5

### Testing and Analysis

#### 5.1: Test Program

For our experiment, we first needed to decide which companies to be the focus for this study. We decided to experiment with Apple Inc., Microsoft Corp., and Peabody Energy Corp. For each company, we needed to collect the following data for each company.



The screenshot shows a Google Sheets spreadsheet titled "Dataset Template". The spreadsheet has a menu bar (File, Edit, View, Insert, Format, Data, Tools, Add-ons, Help) and a toolbar with various icons. The spreadsheet content is as follows:

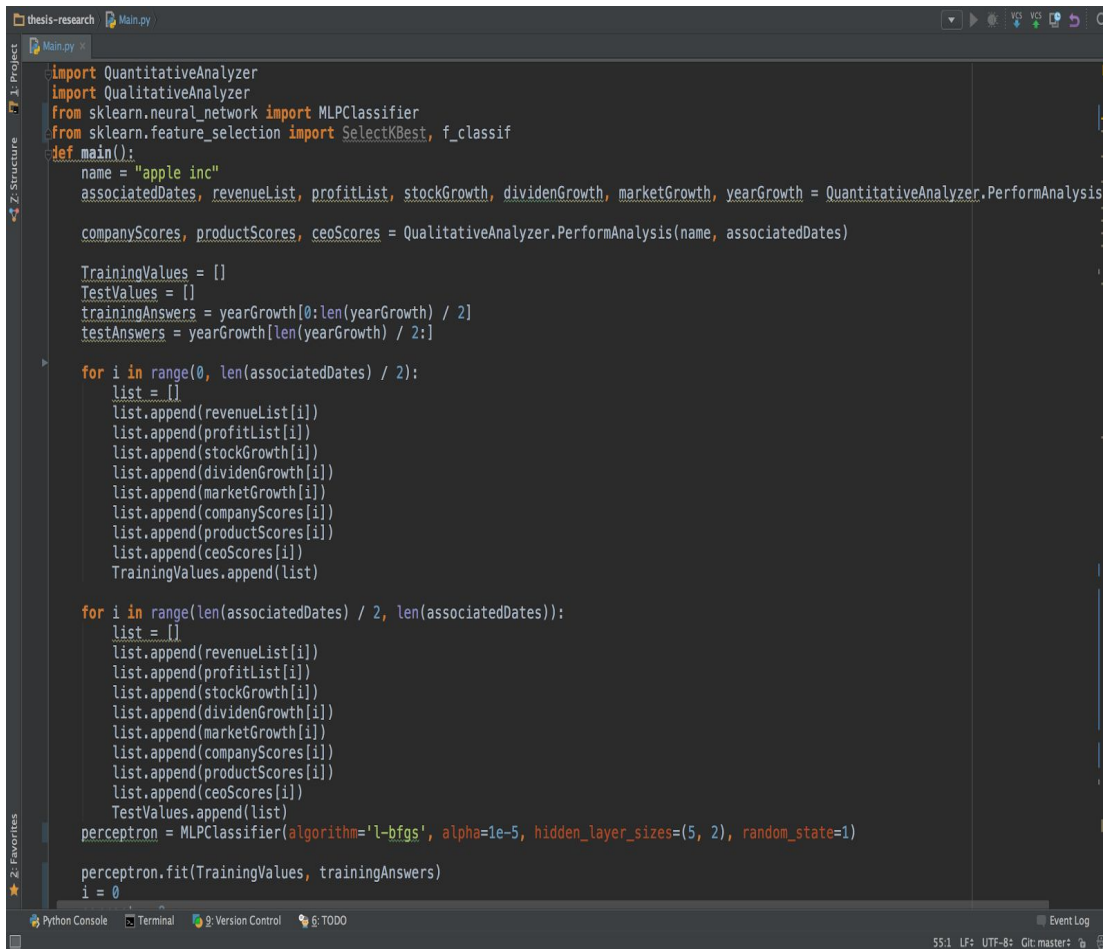
	A	B	C	D	E	F	G	H	I	J
1	Date	Prodscore	Managescore	Companyscore	Revenuegrowth	Profitgrowth	Stockgrowth	Dividendgrowth	Marketgrowth	Yeargrowth
2										

Below the table, there is a "Add" button and a text box containing "1000 more rows at bottom."

Figure 14

For each company we need to collect as much data as possible for the following fields. With this data, we will conduct both a quantitative as well as a qualitative analysis on the collection. This program was written in python. The main methods of operations can be seen on the following pages.

## Main



```
thesis-research Main.py
Main.py
import QuantitativeAnalyzer
import QualitativeAnalyzer
from sklearn.neural_network import MLPClassifier
from sklearn.feature_selection import SelectKBest, f_classif
def main():
    name = "apple inc"
    associatedDates, revenueList, profitList, stockGrowth, dividenGrowth, marketGrowth, yearGrowth = QuantitativeAnalyzer.PerformAnalysis(
    companyScores, productScores, ceoScores = QualitativeAnalyzer.PerformAnalysis(name, associatedDates)

    TrainingValues = []
    TestValues = []
    trainingAnswers = yearGrowth[0:len(yearGrowth) / 2]
    testAnswers = yearGrowth[len(yearGrowth) / 2:]

    for i in range(0, len(associatedDates) / 2):
        list = []
        list.append(revenueList[i])
        list.append(profitList[i])
        list.append(stockGrowth[i])
        list.append(dividenGrowth[i])
        list.append(marketGrowth[i])
        list.append(companyScores[i])
        list.append(productScores[i])
        list.append(ceoScores[i])
        TrainingValues.append(list)

    for i in range(len(associatedDates) / 2, len(associatedDates)):
        list = []
        list.append(revenueList[i])
        list.append(profitList[i])
        list.append(stockGrowth[i])
        list.append(dividenGrowth[i])
        list.append(marketGrowth[i])
        list.append(companyScores[i])
        list.append(productScores[i])
        list.append(ceoScores[i])
        TestValues.append(list)
    perceptron = MLPClassifier(algorithm='l-bfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1)

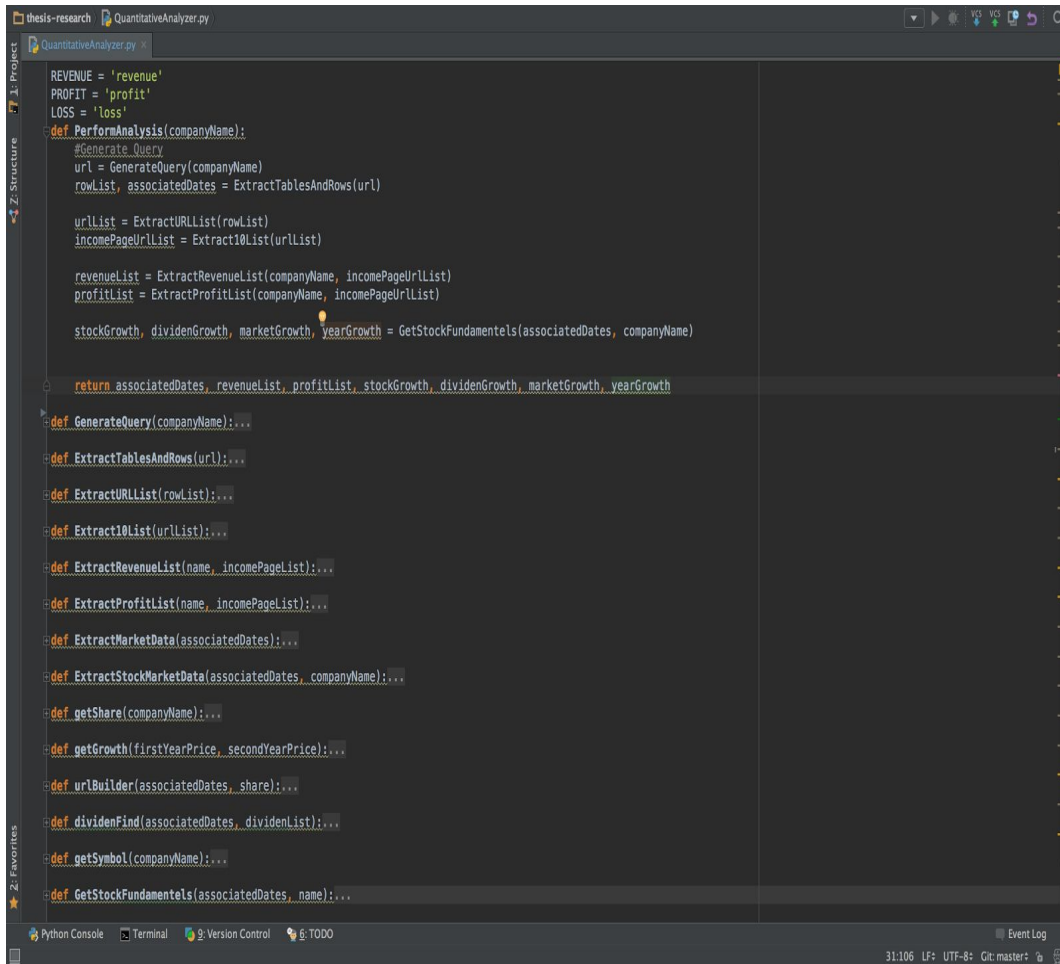
    perceptron.fit(TrainingValues, trainingAnswers)
    i = 0
```

Python Console Terminal Version Control TODO Event Log  
55:1 LF: UTF-8: Git: master

Figure 15



## Quantitative Analyzer



```
thesis-research QuantitativeAnalyzer.py
QuantitativeAnalyzer.py
REVENUE = 'revenue'
PROFIT = 'profit'
LOSS = 'loss'
def PerformAnalysis(companyName):
    #Generate Query
    url = GenerateQuery(companyName)
    rowList, associatedDates = ExtractTablesAndRows(url)

    urlList = ExtractURLList(rowList)
    incomePageUrllist = Extract10List(urlList)

    revenueList = ExtractRevenueList(companyName, incomePageUrllist)
    profitList = ExtractProfitList(companyName, incomePageUrllist)

    stockGrowth, dividenGrowth, marketGrowth, yearGrowth = GetStockFundamentals(associatedDates, companyName)

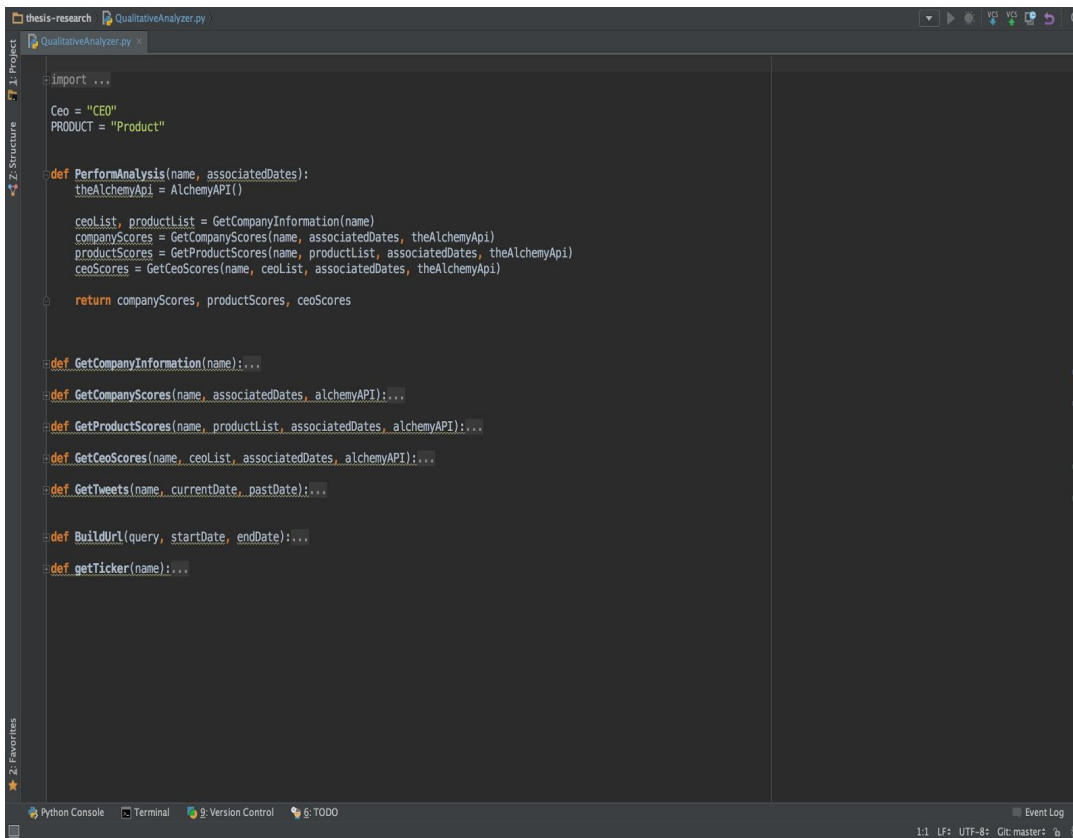
    return associatedDates, revenueList, profitList, stockGrowth, dividenGrowth, marketGrowth, yearGrowth

def GenerateQuery(companyName):...
def ExtractTablesAndRows(url):...
def ExtractURLList(rowList):...
def Extract10List(urlList):...
def ExtractRevenueList(name, incomePageList):...
def ExtractProfitList(name, incomePageList):...
def ExtractMarketData(associatedDates):...
def ExtractStockMarketData(associatedDates, companyName):...
def getShare(companyName):...
def getGrowth(firstYearPrice, secondYearPrice):...
def urlBuilder(associatedDates, share):...
def dividenFind(associatedDates, dividenList):...
def getSymbol(companyName):...
def GetStockFundamentals(associatedDates, name):...
```

Figure 16

Within this class, data is collected for fundamental information. The data we collect are the growths of revenues, profits, stock price, dividends, and the overall market where the revenues and profits come directly from the 10-Q income statements. The year growth field is set to one if the stock's future growth is over ten percent and zero if otherwise. This information is then returned to the main class.

## Qualitative Analyzer



```
thesis-research | QualitativeAnalyzer.py
QualitativeAnalyzer.py
import ...

Ceo = "CEO"
PRODUCT = "Product"

def PerformAnalysis(name, associatedDates):
    theAlchemyApi = AlchemyAPI()

    ceoList, productList = GetCompanyInformation(name)
    companyScores = GetCompanyScores(name, associatedDates, theAlchemyApi)
    productScores = GetProductScores(name, productList, associatedDates, theAlchemyApi)
    ceoScores = GetCeoScores(name, ceoList, associatedDates, theAlchemyApi)

    return companyScores, productScores, ceoScores

def GetCompanyInformation(name):...
def GetCompanyScores(name, associatedDates, alchemyAPI):...
def GetProductScores(name, productList, associatedDates, alchemyAPI):...
def GetCeoScores(name, ceoList, associatedDates, alchemyAPI):...
def GetTweets(name, currentDate, pastDate):...
def BuildUrl(query, startDate, endDate):...
def getTicker(name):...
```

Figure 17

This class collects all tweets needed for a given company. The class takes the dates generated from the quantitative analyzer and extracts tweets of the overall company, the company's products, and the company's CEO. From this a sentiment score is generated for each of the three categories. This is done separately for every date passed to the module.

## 5.2: Results

The program was run separately on the three selected companies and as expected, different results were obtained from each. The first company which was tested was Apple Inc. After gathering the dates of released quarterly reports, which date as far back as 2007, we collected all fundamental annual trends using the quarterly report release dates and one year before as a duration of trends. After an average of 500 tweets was collected for the three sentiment categories for each annual duration, an overall sentiment score was given to the tweet set. The dataset was then split into a training set and validation set and given to a multi-layered perceptron algorithm which the following results were generated. After the training process, the algorithm had a test accuracy of 63.63%. One of the most noticeable factors in the price of apple is the consistency of the stock price. Over the years, the stock was consistently positive until recent years. The next test was on the Peabody Energy Corporation. After collecting data in the same fashion as Apple Inc., the data was split into a training set as well as test set. After training, the algorithm was able to predict at 54.5%. Like Apple, Peabody also showed consistency in its stock price but with a negative trend rather a positive one. The final test was on Microsoft Corporation which showed more mixed trends of the stock price over time. Because of this, the prediction accuracy of the algorithm was 36.4%. However, we performed an F anova test on the data for both Apple Inc. as well as Microsoft Corporation (one company showing consistent stock trends and the other showing more mixed trends). The results showed Apple's most influential aspect was by far the sentiment of the current CEO. However, with Microsoft's more mixed stock

trends, the F anova test showed the previous stock price having the highest impact on the future stock price. But, more interestingly, the overall company sentiment gathered from the twitter feeds came in second. Below are tables of both anova test results.

Apple Anova Test

Revenue Growth	Profit Growth	Stock Growth	Dividend Growth	Market Growth	Company Sentiment	Product Sentiment	CEO Sentiment
0.1202195	0.163001	2.49331334	2.82545079	0.38856215	2.39333488	2.84094941	9.02877147

Figure 18

Microsoft Anova Test

Revenue Growth	Profit Growth	Stock Growth	Dividend Growth	Market Growth	Company Sentiment	Product Sentiment	CEO Sentiment
0.00413533	0.40633091	2.75255961	0.02053024	0.63403994	1.66313795	0.55008673	0.39845534

Figure 19

## Chapter 6

### Conclusion

After analyzing the data findings, we came to the conclusion that sentiment over a period of time can have an effect of the future stock price. However, after a year of growth, the general trend of the stock price is decided by the fundamentals as well as the overall sentiment. One of the greatest mysteries of fundamental analysis is the idea that it is known the stock price will eventually reflect what the fundamentals present but how long it will take for the stock to reflect the fundamentals. From our findings, we found it is more probable the stock growth will be more synonymous with the trends of the fundamentals rather than the trend of the sentiment. However, it is important to still consider sentiment because it can still have just as great of effects on the future stock price even in the long term.

## References

- 1.) <http://www.investopedia.com/terms/q/qualitativeanalysis.asp>
- 2.) Foundations for a Disequilibrium Theory of the Business Cycle:
- 3.) Islam, A. “Automated fundamental analysis for stock ranking and growth prediction”, Computers and Information Technology, 2009. ICCIT '09. 12th International Conference
- 4.) Jeyapriya, A. “Extracting Aspects and Mining Opinions in Product Reviews Using Supervised Learning Algorithm”, Dept. of Comput. Sci. & Eng., Kongu Eng. Coll., Erode, India
- 5.) Wenhao, Huang “Deep Process Neural Network for Temporal Deep Learning”, Dept. of Electron. Eng. & Comput. Sci., Peking Univ., Beijing, China ; Haikun Hong ; Guojie Song ; Kunqing Xie
- 6.) Harris, M. D. 1985 Reston, Virginia: Reston Publishing Company (Ch. 1 Basic Linguistics)
- 7.) <http://www.nltk.org/book/ch07.html> “Extracting Information From Text”
- 8.) <https://class.coursera.org/nlp/lecture/preview>
- 9.) [https://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html#Introduction](https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction) to neural networks
- 10.) <https://www.coursera.org/learn/machine-learning>
- 11.) <http://www.statisticssolutions.com/mlr/>
- 12.) [http://scikit-learn.org/dev/modules/neural\\_networks\\_supervised.html](http://scikit-learn.org/dev/modules/neural_networks_supervised.html)
- 13.) Qiang Ye', Bing Liang, Yijun Li “Amnestic Neural Network for Classification: Application on Stock Trend Prediction”, School of Management, Harbin Institute of Technology, Shanghai, China

14.) Iuhasz, Gabriel. Tirea, Monica. Negru, Viorel. “Neural Network Predictions of Stock Price Fluctuations”, Computer Science Department, West University of Timisoara, Timisoara, Romania

15.) Lu Dang Khoa, Nguyen. Sakakibara, Kazutoshi. Nishikawa, Ikuko. “Stock Price Forecasting using Backpropagation Neural Networks with Time and Profit Based Adjusted Weight Factors”, Graduate School of Science and Engineering, Ritsumeikan University, Japan

APPENDIX A  
APPLE DATA COLLECTED



Apple Inc

Revenue Growth	Profit Growth	Stock Growth	Dividend Growth	Market Growth	Company Sentiment	Product Sentiment	CEO Sentiment	Year Growth (10%)
0.295256450325	0.3788249694	0.4477497208949517	0	0.08977762115993754	-0.192749	-0.184309	0.0123781	0
0.059706140475	0.122898550725	-0.1033084692868898	0.07869913474558764	-0.029317599115664474	-0.127633	-0.179052	-0.0134965	1
0.0468545742265	0.0708075835341	-0.1033084692868898	0.07869913474558764	-0.029317599115664474	-0.108682	-0.285532	-0.0265584	1
0.0565380099795	-0.000458785747056	-0.13664778859608895	0.15093641862799473	-0.10998318445743833	-0.094247	-0.198481	0.0470468	1
0.00856579961739	-0.218041704442	0.14730285545633565	0.15093641862799473	0.0344960771580132	-0.163369	-0.198005	0.0188887	1
0.112718828153	-0.178540698675	-0.21097172517507762	0.15093641862799473	0.08635495830810766	-0.0984791	-0.204336	0.0281385	1
0.176526449831	0.00107164727495	-0.013548765416125888	0	-0.02712523161631944	-0.0520071	-0.192922	0.0650821	0
0.225823387351	0.207443897099	0.03902707952695424	0	0.06560158717328543	-0.000575028	-0.126187	0.07942	1
0.588600154052	0.941205946217	0.03902707952695424	0	0.06560158717328543	-0.0343645	-0.1959	0.151583	1
0.732657716615	1.17588274484	0.03902707952695424	0	0.06560158717328543	-0.0240753	-0.119139	0.225101	1
0.819808917197	1.24654165386	-0.1560706626240106	0	-0.10952010699826316	-0.10399	-0.0782416	0.1656	1
0.827320542262	0.947625243982	-0.29631541034852765	0	-0.16950447147864645	-0.0678608	-0.0419272	0.170895	1
0.705094688516	0.777383066903	-0.3331324803371207	0	-0.09280568179769946	-0.186347	-0.0856158	0.242129	1
0.612903225806	0.7795404814	0.417790198423411	0	0.17665578589165526	-0.114745	-0.0650266	0.0359827	1
0.486019374725	0.897530864198	0.6596913740729	0	0.3087838214780181	-0.143448	-0.0774759	0.0154014	0
0.320117845118	0.49800443459	-0.3688758716056719	0	-0.2232100687117324	-0.0842432	-0.0665479	0.0793733	1
0.116961414791	0.146455223881	0.19513885150869067	0	0.32837943412028103	-0.0567867	-0.121025	-0.0520713	1
0.086661341853	0.153110047847	-0.11203132073480254	0	0.0049417453047745285	-0.105752	-0.183638	-0.0880531	1
0.0581806827644	0.0151802656546	0.2958454994660721	0	-0.10636436943132112	-0.104047	-0.194488	-0.119872	0
0.37966728281	0.310513447433	-0.17326218406878827	0	0.012892290910531398	-0.0983836	-0.227879	-0.09956	0
0.427051671733	0.357142857143	-0.17326218406878827	0	0.012892290910531398	-0.185043	-0.241391	-0.149222	0
0.350386507379	0.574701195219	-0.17326218406878827	0	0.012892290910531398	-0.111504	-0.238205	-0.181508	0

APPENDIX B  
MICROSOFT DATA COLLECTED

Microsoft Corp.

Revenue Growth	Profit Growth	Stock Growth	Dividend Growth	Market Growth	Company Sentiment	Product Sentiment	CEO Sentiment	Year Growth (10%)
0.0795709449814	-0.105977432144	0.27223014436982845	0	0.08977762115993754	0.0216656	0.0553395	0.358005	0
0.252145285768	-0.134248665141	0.002720416103780749	0.10714285714285703	-0.029317599115664474	0.0303472	0.0902761	0.392648	1
-0.00419737420079	-0.065235342692	0.002720416103780749	0.21739130434782614	-0.029317599115664474	-0.0398289	0.0339231	0.439019	1
0.142757270694	0.028383252313	-0.147529012977926	0.21739130434782614	-0.10998318445743833	0.0540957	0.0542364	-0.117339	1
0.157483758121	0.17420510524	0.03319384129296495	0.21739130434782614	0.043234097526144225	0.00960308	0.055434	0.0189676	1
0.177055207675	0.185395458105	-0.0802129329059432	0.15	0.08635495830810766	0.030566	0.0364409	-0.0269995	1
0.0273401963131	-0.037288647343	-0.0426652910187212	0.15	-0.02712523161631944	0.12342	0.0995909	0.0797801	0
-0.078517154041	-0.221680027884	-0.13185335719641422	0.15	-0.04711246200607903	0.0974803	0.0439922	0.0502475	0
0.0595933771609	-0.0237003058104	0.02987534103093412	0.25000000000000006	0.06560158717328543	0.0282207	0.0952483	-0.022629	0
0.0467097679547	-0.00150738619234	0.02987534103093412	0.25000000000000006	0.06560158717328543	0.0457648	0.0186597	-0.0396327	1
0.072676752084	0.0606284658041	0.006367902772141794	0.25000000000000006	0.004801072199819488	0.0546042	0.0556381	-0.0428222	0
0.132731159071	0.306040938592	-0.08928243264290639	0.23076923076923075	-0.16950447147864645	0.129927	-0.047494	0.0020876	0
0.0489433287772	-0.00420294205944	0.02904030537767359	0.23076923076923075	-0.09280568179769946	0.0172684	-0.0405195	-0.0286392	0
0.253482972136	0.513710128707	0.14130494465169333	0.23076923076923075	0.06501452120583563	0.0666066	0.0106949	0.041423	0
0.0626465416178	0.345649983205	0.28488111148535833	0	0.3087838214780181	-0.0162499	-0.0243389	0.0317119	0
0.14390522581	0.596070915189	-0.32478456933145344	0	-0.2232100687117324	0.0945457	-0.0384069	-0.0252629	1
-0.142155235376	-0.182712096959	-0.32478456933145344	0	-0.2232100687117324	0.0283415	-0.0303054	-0.0545667	1
-0.0557631105576	-0.321558796718	0.013916524539883693	0.18181818181818185	0.0049417453047745285	-0.0289274	-0.0839411	-0.0709384	1
0.0160078206147	-0.113235606543	-0.09817212133059881	0.18181818181818185	-0.10636436943132112	0.0157388	-0.0284075	-0.115582	0
0.0943903502398	0.019584984845	-0.10391926530494558	0.18181818181818185	-0.1668344919829511	0.0926727	-0.0656313	-0.0909678	0
0.00388942908737	-0.109216402761	-0.06096365674671192	0	0.012892290910531398	0.0163535	-0.103707	-0.0705438	0
0.304975283049	0.792460015232	-0.06096365674671192	0	0.012892290910531398	0.00481046	-0.132251	-0.0149969	0

APPENDIX C

PEABODY ENERGY DATA COLLECTED

Peabody Energy Corp.

Revenue Growth	Profit Growth	Stock Growth	Dividend Growth	Market Growth	Company Sentiment	Product Sentiment	CEO Sentiment	Year Growth (10%)	
-0.0415554072096	-6.80104712042	-0.4109660487869292		0	0.0885748442187893	0.0524432	-0.135871	0.01	0
0.018953225526	-1.81745120551	-0.05536909665205342		0	-0.029317599115664474	0.177083	-0.0967596	-0.156886	0
-0.0693363844394	-1.27319587629	-0.05536909665205342		0	-0.029317599115664474	0.0761401	-0.0158613	0.304745	0
-0.126870021372	-1.45913461538	-0.18735638863498652		0	0.043234097526144225	-0.0939428	-0.101018	-0.198339	0
-0.129120185755	-0.580846968239	0.12298069518635027		0	-0.04333709950168785	-0.00722718	-0.122659	-0.108431	0
-0.134953234028	-1.10880538418	-0.02183137254415167		0	0.11713030957775082	0.102379	-0.0463849	0.1251	0
0.0394829849541	-0.852220248668	-0.07917876058308479		0	-0.04711246200607903	0.102819	-0.0733804	0.01	0
0.00893713708659	-0.288843258042	-0.26522657622409906		0	0.06560158717328543	0.14394	-0.148383	0.01	0
0.169525557914	-0.00223838836038	-0.26522657622409906		0	0.06560158717328543	0.0488196	-0.0998644	0.01	0
0.0918110151767	0.191282268303	0.3359351753856663		0	0.0686097779045895	0.24888	-0.111961	0.819038	0
0.208619236788	0.364145658263	-0.12632068466752933	0.21428571428571416	-0.10952010699826316	0.249181	-0.157225	0.343334	0	
0.151293217208	0.307242136064	-0.40046795881199426	0.21428571428571416	-0.16950447147864645	0.352027	-0.136381	0.369631	0	
0.118596280744	1.13267148014	0.018708052264984017	0.21428571428571416	0.06501452120583563	0.240363	-0.228953	0.332289	1	
0.24151845763	1.61219512195	0.018708052264984017	0.16666666666666669	0.06501452120583563	0.213955	-0.110338	0.392294	1	
0.0430832759807	-0.219748858447	0.5880020474725458	0.16666666666666669	0.3087838214780181	0.18064	-0.101007	0.453462	0	
-0.117802709568	-0.701990317375	-0.25961004341450594	0.16666666666666669	-0.2232100687117324	-0.121464	0.039439	0.01	1	
-0.121815443055	-0.652247667515	-0.25961004341450594	0	-0.2232100687117324	-0.613301	-0.0814132	0.01	1	
0.153226443409	2.02590673575	-0.160191680765999	0	0.0049417453047745285	0.01	-0.153787	0.01	0	
0.590070921986	10.4427244582	0.15257454384572494	0	-0.12121191846976052	-0.0421712	-0.30102	0.01	0	
0.432354041916	1.16713091922	0.2971129311393885	0	-0.15950025360844644	-0.0455819	-0.231061	0.01	0	
0.149720848591	-0.35411585655	-0.03780760833061221	0	0.012892290910531398	-0.0440452	-0.233139	0.01	1	