

**Sentiment Analysis of News Articles for Stock Price Prediction**

By

**Bolanle Esther Onifade**

**(beo01u)**

**School of Computer Science**

**University of Nottingham**

**Supervisor: Michel Valstar**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Abstract

Several theories regarding the behaviour of the stock market exist. These theories put forward ideas ranging from those claiming that the stock market cannot be predicted and to those claiming those that with the right assumptions, the stock market can be predicted. Based purely on numbers, it’s hard to see how hard a predictor can predict the stock market as often the numbers are reflective of non-numerical factors. In this dissertation we aim to consider those non-numerical factors that affect the price of specific stock. These non-numerical factors come in form of news articles. In the digital age of rapid response to ongoing situations, news articles about events tend to be released as soon as they occur and the assumption is that if we can track news sources, monitoring them for the release of relevant news articles, we can use the sentiment orientation of the article (whether positive, negative or neutral) to predict the price of the stock market before the market has a chance to react to the article. This of course poses an important question on how the sentiment orientation of articles. There are several approaches that can be taken towards determining the class of articles but for the purposes of this dissertation, we will focus on the use of Support Vector Machines to perform classification. The output of classification will then be fed into a second layer of support vector machines to perform the price prediction.

# Acknowledgements

I would like to firstly thank my supervisor Michel Valstar for the encouragement to explore areas of analysis that I otherwise wouldn’t have thanked. Helping me think outside the box has helped take the dissertation much farther than I thought it would go. It would so be very poor form to neglect to thank Dr Robert Young who has helped me with aspects of finance and economics that I have found difficult to wrap my head around. I also would like to thank the few anonymous students from business school that helped with labelling the news articles.

To my parents, sisters and friends, I would like to say a massive thanks for all these years of supporting me through my education. A lot of time has passed since I started school 18 years ago and this dissertation is a representation of all the knowledge I have gained in that time. I truly appreciate all the support you’ve all expressed for me and this project.

**Table of Contents**

[Abstract 2](#_Toc415768573)

[Acknowledgements 3](#_Toc415768574)

[Table of Figures 6](#_Toc415768575)

[1. Introduction 9](#_Toc415768576)

[1.1. Motivation 9](#_Toc415768577)

[1.2. Objectives 10](#_Toc415768578)

[1.3. Assumptions 10](#_Toc415768579)

[1.4. Organisation of Document 11](#_Toc415768580)

[2. A Brief Review of the Literature 12](#_Toc415768581)

[2.1. Sentiment Analysis 12](#_Toc415768582)

[2.2. Stock Price Prediction Using Sentiment Analysis 14](#_Toc415768583)

[2.3. Concluding Remarks 17](#_Toc415768584)

[3. Background Knowledge 18](#_Toc415768585)

[3.1. Sentiment Analysis 18](#_Toc415768586)

[3.1.1. Feature Generation 18](#_Toc415768587)

[3.1.2. Feature Representation 19](#_Toc415768588)

[3.1.3. Feature Reduction 20](#_Toc415768589)

[3.2. T-Test Based Split-and-Merge Piecewise Linear Approximation 20](#_Toc415768590)

[3.2.1. The Splitting Phase 20](#_Toc415768591)

[3.2.2. The Merging Phase 22](#_Toc415768592)

[3.3. Support Vector Machines 23](#_Toc415768593)

[3.4. Hidden Markov Models 23](#_Toc415768594)

[3.5. Hybrid SVM-HMM Model 23](#_Toc415768595)

[4. Scientific Method 24](#_Toc415768596)

[4.1. Method Overview 24](#_Toc415768597)

[4.2. Sentiment Classification 24](#_Toc415768598)

[4.2.1. Data Acquisition 24](#_Toc415768599)

[4.2.2. Data Labelling 26](#_Toc415768600)

[4.2.3. Data Pre-processing 28](#_Toc415768601)

[4.2.4. Document Representation 28](#_Toc415768602)

[4.2.5. Feature Selection or Reduction 28](#_Toc415768603)

[4.2.6. Classification 29](#_Toc415768604)

[4.3. Price Prediction 29](#_Toc415768605)

[5. Evaluation and Results 29](#_Toc415768606)

[5.1. Method Overview 29](#_Toc415768607)

[5.2. Data Acquisition 29](#_Toc415768608)

[5.2.1. News Article Acquisition 29](#_Toc415768609)

[5.2.2. Stock Data Acquisition 30](#_Toc415768610)

[5.3. Labelling 30](#_Toc415768611)

[5.3.1. Manual Labelling 30](#_Toc415768612)

[5.3.2. Automatic Labelling 32](#_Toc415768613)

[5.3.3. Labelling Discussion 34](#_Toc415768614)

[5.4. Data Pre-processing 34](#_Toc415768615)

[5.5. Feature Selection or Feature Reduction 35](#_Toc415768616)

[5.6. Document Classification 36](#_Toc415768617)

[5.6.1. Manual Classification 36](#_Toc415768618)

[5.6.2. Automatic Classification 41](#_Toc415768619)

[6. Conclusion 42](#_Toc415768620)

[6.1. Keys for Transforming Sentiments to Numbers 42](#_Toc415768621)

[7. Appendix 43](#_Toc415768622)

# Table of Figures

[Figure 4.1 – XML Format for Scrapped News Articles 25](file:///C:\Users\Ester\Documents\Fourth%20Year\G54MIP\Documents\Dissertation.docx#_Toc415768554)

[Figure 5.1 – Number of articles collected for each company 29](#_Toc415768555)

[Figure 5.2 – Table of correlation between sentiment and stock price using manually labelled data 30](#_Toc415768556)

[Figure 5.3 – Actual Stock Price and Projected Price of Disney 32](#_Toc415768557)

[Figure 5.4 – Actual Stock Price and Projected Stock Price of Exxon 33](#_Toc415768558)

[Figure 5.5 – Aligned News Articles with Trends 33](#_Toc415768559)

[Figure 5.6 – Correlation values of the automatically generated data 34](#_Toc415768560)

[Figure 5.7 – Initial features and pre-selected features 35](#_Toc415768561)

[Figure 5.8 - Number of Features depending on whether feature selection or extraction is used 35](#_Toc415768562)

[Figure 5.9 – Words selected for progress classification (manually labelled data) 36](#_Toc415768563)

[Figure 5.10 – Words selected for feeling classification(manually labelled data) 36](#_Toc415768564)

[Figure 5.11 – support for the various classes (manual/progress) 37](#_Toc415768565)

[Figure 5.12 – Confusion matrices (Manual/Progress). Top left – Unigram, Top Right – Bigram, Bottom – Unigram + Bigram 38](#_Toc415768566)

[Figure 5.13 – Table of performance of linear SVM measured by cross validation (manual/progress) 38](#_Toc415768567)

[Figure 5.14 – Support for the classes (Manual/ Feeling) 39](#_Toc415768568)

[Figure 5.15 – Confusion matrices (Manual/ Feeling). Top left – Unigram, Top Right – Bigram, Bottom – Unigram + Bigram 40](#_Toc415768569)

[Figure 5.16 –Table of performance of linear SVM measured by cross validation (Manual/ Feeling) 40](#_Toc415768570)

[Figure 5.17 – Support for the classes (Automatic / Progress) 41](#_Toc415768571)

[Figure 5.18 - Table of performance of linear SVM measured by cross validation (Automatic/ Feeling) 41](#_Toc415768572)

**Table of Algorithms**

[Algorithm 3‑1 – The Split Algorithm 21](#_Toc415768504)

[Algorithm 3‑2 – The Merge Algorithm 22](#_Toc415768505)

# Introduction

One can arguably say that state of the world economy has been built with the stock market serving as a base. Hence, while there are other means of evaluating a country’s economy, the state of the stock market is perhaps one of the more important means of doing so. Predicting the stock market is general direction of stock prices therefore is very important in order to make decisions regarding where and what kind of investment takes place. The current work attempts to predict the direction of movement of the (close) stock price of the next working day given the current day’s information. We propose a prediction system which incorporates news articles other technical data such as the stochastic %K, stochastic %D and other selected features.

The advent of the Internet brought with it improvements in many fields ranging from technologies that influence our daily lives to those that may one day take humans to other planets. One of the more mundane improvements is access to news articles – completing changing the way we now make decisions. We no longer have to wait until the following day to find out about events that took place today. Stock market traders are one group of people who rely heavily on the news articles. Often times, trades are executed using information from the news are made unconsciously. This is because news articles need not necessarily carry extremely significant news in order to be usable to traders. News articles will often bear information about annual earnings, acquisitions and mergers, changes in administration and management as well as stock splits.

Traders will often base their trades on pieces of information from news articles regardless of how seemingly important it is. We therefore argue that the key to predicting the stock market is by monitoring extensive sources of news articles and extracting valuable information from the articles which can then we used to predict the stock market. The process of extracting information automatically from textual data is referred to as sentiment analysis. The current work therefore is an interdisciplinary work that ties together sentiment analysis, machine learning and finance for the prediction of the stock market.

The next section details the motivation for this project. We then carry on describing our objections, the assumptions we make and structure of the current work.

## Motivation

A lot of past work has been done using technical and fundamental data such as the current investment, general economy, recessional periods, currency and industry – this is known as fundamental analysis. However, very few articles have attempted to go beyond that and while some of these methods perform reasonably well, very few articles have attempted to go beyond historical data. Even still, within the set of articles that attempt to use recent (external) information such as news articles, very few articles attempt to incorporate historical technical data. With this work, we aim to fill in this hole by utilising both historical data (refer to section that describes the features)that provide us with an indication of how well the stock price has done in the past as well as current news that provide us with a sense of what the general sentiment regarding a certain stock is. Thus, it stands to reason that we will make much higher profits with finding the middle ground between the two extremes. A few related works have attempted to use both methods to predict the stock market (Deng, Mitsubuchi, Shioda, Shimada, & Sakurai, 2011); however, the backbone of the current proposed system uses a Support Vector Machine and Hidden Markov Model (SVM-HMM) hybrid model for the predictions.

## Objectives

The primary goals are to propose a system by which the stock price can be predicted as well as perform experiments that test the performance of the system. We also explore the effectiveness of the hybrid system in predicting stock prices.

Furthermore, we aim to extensively and conclusively review related, previous work on the prediction of the stock market both technical indicators and sentiment-based indicators. Although our discussion of sentiment analysis is heavily favoured towards machine learning based techniques, we aim to give a well-rounded discussion by taking non-machine learning based techniques into consideration.

## Assumptions

There are several schools of thought regarding whether the stock market can at all be predicted and in order to continue, we must first discuss the current hypotheses and highlight which hypotheses we have based the project on. Our hypotheses come in the form of the three levels of the Efficient Market Hypothesis.

The weak-form efficient market hypothesis assumes that the market is efficient. In addition, it also assumes that the rates of return are independent, meaning that past return has no bearing on future return. Following from this, traders both algorithmic and human make invalid assumptions when trading.

The semi-strong form efficient market hypothesis assumes that the market, at all times reflects all publicly available information. The stock market hence responds very quickly to new information. Conclusively, this implies that potential investors cannot profit on the stock market as investors can only trade based on new information, after the market has adjusted to it.

The strong form efficient market hypothesis assumes that the market, at all moments reflects both publicly and privately available information. This incorporates both the weak and semi-strong form of the efficient market hypothesis and following this hypothesis, no one can make money.

Hence, it’s quite clear that one of the first decisions to be made is whether or not the efficient market hypothesis is one of our assumptions and if it is, which level. It’s quite clear that to assume at all the any form of the Efficient Market Hypothesis would invalidate the current work; hence, we do not assume the efficient market hypothesis. The Efficient Market Hypothesis assumes that all investors are rational and that there exists a perfect flow of information which clearly is invalid. Instead, we assume that at the end of each working day, the stock price reflects the available information. It’s safe to make this assumption as should an important piece of news be released that alters the stock price significantly, the price at the end of the day will reflect this. Should a company be say downgraded by standard and poor, the stock price will reflect this downgrade until the rating is increased once more.

We also assume that the effect of information extends into multiple working days. Hence, an Hidden Markov Model can identify patterns based on the trend of the past several working days.

We assume a much simpler model of the stock market – predicting the direction of movement of the closing price of the stock market rather than the intra-day price. It’s important to note at this point that predicting the stock market needn’t necessarily involve predicting the exact values – predicting simply the direction of movement is enough (Elkan, 1999; Lavrenko, Schmill, Lawrie, Ogilvie, Jensen, & Allan, 2000; Thomas & Sycara, 2000)

Finally, we make the assumption that any important trend that changes the rate of return of a stock price in any significant way can be extracted from news articles.

## Organisation of Document

In the succeeding chapter, we provide an in-depth review of the literature both on sentiment analysis and on stock price prediction. In Chapter 3, as we assume that the average reader is unfamiliar with the techniques used for the project, we provide preliminary background knowledge. Chapter 4 details the scientific method used for the project. Chapter 5 details the results of the experiments run. To conclude, chapter 6 details the closing remarks.

# A Brief Review of the Literature

The current chapter is split into three sections. The first section reviews the recent work on sentiment analysis. The second section reviews the work on specifically the domain of stock market prediction with sentiment analysis. In addition, we note that there are several means by which sentiment analysis can be applied for use in stock price prediction and we aim to provide a brief overview of the myriad of techniques used in the literature. Stock price prediction does not begin and end with sentiment analysis. Hence, we have dedicated a section to the discussion of some of the techniques that have been used to predict the stock market without sentiment analysis. As this refers to a broad range of techniques, we restrict our discussion to the articles that are most relevant.

## Sentiment Analysis

Bing Liu (2012) provides three levels of sentiment analysis: document based, sentence based and aspect-entity based. Document-based sentiment analysis pertains to the classification of an entire document and the key assumption is that the entire document focuses on a single entity. Sentence-based analysis is more in-depth analysis of individual sentences. Aspect-entity based analysis focuses on determining the subject of discussion as well as the sentiment. Even further, regular opinion express opinion on a single entity while comparative opinion expresses opinion on two or more entities.

Bing Liu highlights that the most important indicator of sentiment are sentiment words such as *poor, happy, sad, brilliant* or *excellent.* Of course, one glaring issue with simply analysing based on sentimental words is the use of sarcasm in language. This, in addition with the fact that sentimental words can change orientation depending on the context means that sentiment analysis can’t be reduced simply to a keyword search. It should be noted at this point that financial news articles generally do not benefit from the use of sentimental words. Liu discusses extensively sentiment analysis using varying techniques but since we are simply interested in only document-based classification, we focus articles related to such classification. It is recommended to refer to Bing Liu to gain a more complete and up-to-date overview of the current sentiment analysis methods.

One of the more important aspects of document-based classification is the determination of features. Popular ones include terms and their frequency (which is the selected features for this current projected), parts of speech (POS) – introduced by Turney (2002) – and their tags such as sentiment words and sentiment flippers (words that change the orientation of sentiment words such as *not*). Bing Liu also comments that some domains are easier than others, for example movie reviews tend to be easier to analyse than car reviews.

Following generation of labels is feature generation. In the literature, there are two main approaches: a lexicon-based approach and a bag-of-words approach. The lexicon-based approach as used in Whitelaw et al. (2005) typically involves a set of example words which are then used as a seed for building a larger lexicon through the identification of synonyms in a semi-automatic manner. Bag of words approach typically involves the representation of words as a numerical value indicating their presence, frequency or term frequency-inverse document frequency (TF-IDF)score Yong et al. (2011). One of the more often cited criticisms of the bag-of-words approach is the loss of semantic association between terms.

Work has been done to improve on the TF-IDF by introducing the delta TFIDF (Martineau & Finin, 2009 ). Delta TF-IDF improves the importance placed on words that are unevenly distributed in the corpus and discounts those that are evenly distributed – the rationale being that the less unevenly the feature is represented, the higher the chances that it’s relevant for its classification. Delta TF-IDF is formulated mathematically as:

where is the number of times term appears in document d, is the number of positively labelled documents containing term t, is the number of positively labelled documents, is the number of negatively labelled documents with term , is the number of negatively labelled documents and is the value for term t in document . Using feature values derived by delta TF-IDF on movie review classification, a SVM achieved an accuracy of 88.1% compared to traditional TF-IDF based classification accuracy of 82.85%.

Feature selection is critical in sentiment analysis as corpuses tend to be polluted with noise, reducing the performance of any developed system. Core techniques involved in feature selection are stop-word removal and stemming. Furthermore, statistical methods such as the point-wise mutual information (Turney & Littman, 2003; Wilson, Wiebe, & Hoffmann, 2005) and chi-square are used in the feature selection process.

The point-wise mutual information measures the correlation between a term and the class . Assuming mutual independence, PMI can be calculated using the joint distribution and the individual distributions. This is formulated mathematically by Aggarwal and ChengXiang (2012) as:

where the expected co-occurrence of and is and the actual co-occurrence is . A PMI of greater than 0 means a positive correlation between and and the inverse is the case for a PMI less than 0. Chi-square is another method utilised in feature selection; we describe it in section 3.1.3.

Sentiment classification techniques are split into two broad range of techniques: machine learning techniques (Yong, Xu, & Ren, 2011; Pang, Lee, & Vaithyanathan, 2002; Kang, Yoo, & Han, 2012; Kaufmann, 2012; Moraes, Valiati, & Neto, 2013), lexicon-based approach – which is further split into dictionary-based approach (Guang, Xiaofei, Feng, Yuan, Jiajun, & Chun, 2010; Minging & Bing, 2004; Kim & Hovy, 2004) and corpus-based approach (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990). We will focus on the machine learning techniques.

Pang et al. (Pang, Lee, & Vaithyanathan, 2002) using a corpus of reviews that were classified according the number of stars associated with the text reviews, applied naïve bayes, maximum entropy and support vector machines. They compared the performance of unigrams, unigrams + bigrams, bigrams, unigrams + POS, adjectives, top unigrams and unigrams + positions using three-fold cross-validation. The performance of each variation in feature depended heavily on the type of technique used. However SVMs on average performed better than both naïve bayes and maximum entropy. It’s important to note that the classifiers performed best when the terms were represented simply as a presence as opposed to frequency. Unigrams performed best in their experiments using the Support Vector Machine-based model with an accuracy of 82.9%.

Yong et al. (2011) propose a method of sentiment analysis similar to the approach the current work has taken, the pre-process the words by removing stop words and tokenisation. Each word is then sorted based on calculated mutual information and select a number of words as the feature item and classify using an SVM. They neglect to specify the domain and source of the corpus. However, they achieve very high classification rates with total precision of 81.11%, recall of 81.42% and F-value of 81.25 in a closed test.

## Stock Price Prediction Using Sentiment Analysis

Often, it is the case that a useable corpus of financial news isn’t available hence authors are forced to generate their own data (Zhan, Cohen, & Atreya) either by manually classifying or automatically generating data (Fung, Yu, & Lu, 2005; Zhan, Cohen, & Atreya). Zhan et al. in their comparison of manual and automatically generated data highlight that the key to classifying news articles manually is the general information that is being conveyed by the articles. Mergers, lower interest rates are general considered *good news* while corruption, lawsuits, wars are considered *bad news.* Working with automatically generated data based on simply the stock price movement (if the stock price goes up after the article is released then the article is good otherwise, it’s bad). The classifier developed has F1-scores of 0.26, 0.38 and 0.36 for positive, neutral and negative articles respectively. They attributed the poor performance to the lack of data and poor article selection whilst also suggesting that news articles are better suited to long-term prediction as opposed to day-to-day prediction. It’s quite clear that there are while there are issues with automatic generation of data (section 5.3.2.), the method employed by Zhan et al. can be said to be too naïve.

Regardless of the criticisms of automatically labelling news article based on the direction of movement of the next day’s price movement, Kaya et al had a 60% accuracy using the method as well as -based feature selection. Fung et al. (2005) utilise piecewise linear segmentation to automatically classify news articles – a method that has been employed in the current work and will be explained in the next chapter.

Gidofalvi(2001), derive a unique method of automatically assigning labels to news articles by aligning news articles to the intraday stock data and although it doesn’t perform very well, we spend some time on it due to its contribution to the literature. A window of influence is defined which is used to evaluate the possible effect of a news article. The author defines the window of influence of an article with the timestamp as the lower boundary offset and the upper boundary offset from t. An offset is negative is is prior to. In addition, news articles that aren’t published within the opening and closing market times are filtered out as these are said to be ambiguous.

To establish how stable/volatile a stock is, a β-value is calculated using the linear regression on data-points (Δ index-price, Δ stock-price). Hence, a β-value of 1 means that whenever the index price changes by δ, the stock price is expected to change by δ as well. A β-value of 2 means that whenever the index price changes by δ, the stock price is expected to change by 2δ as well. A β-value of greater than 1 are relatively volatile and the inverse is the case for stocks less than 1.

In order to remove the effects of the exponential change in price, the formula is

The movement of a stock within a time interval is

where is the change in the stock price and is the change in index price during the time interval A news article with timestamp can then be measured with offsets to receive a score of .

Movement classes can then be defined from these equations:

Where and are threshold values. Naïve Bayesian can then be used to predict the probability of a document belonging to a class.

The predictive power of the classification/system discussed is low with the system performing worse than randomness. On analysis, low values show that the movement measure model is poor-fitting to the stock price.

On evaluating the labelling of the news articles, they discover the highest statistically significant settings of and . The authors also find that the most statistically significant settings for alignments are [-20, 0] and [0, 20], that is 20 minutes before and 20 minutes after the release of the news article.

The predictive capability of the classifier was very low and the apparent reasoning for this the do not accurately model the relative movement of the stock correctly. They conclude by acknowledging that their results contradict the efficient market hypothesis.

Perhaps, more unconventionally, is the use of tweets from twitter to predict the stock market (Bollen, Mao, & Zeng, 2011). They determine the correlation between the mood of the twitter feeds and the Dow Jones Industrial Average. The moods are determined using OpinionFinder (classifies into positive and negative) and Google-Profile of Mood States (GPOMS). A Granger Causality Analysis and Self-organising Fuzzy Neural Network are then used to determine the validity of the hypotheses that moods predict the stock market. The orientation provided by OpinionFinder is discovered to be less predictive than the GPOMS dimension “Calm”. They also claim that there exists a (3-4 days) time lag between the mood expressed on twitter and the changes in the DJIA values – hence stock price movements can be known well in advance.

Bar-Haim et al. (2011) propose a method of identifying expert investors from twitter feeds, which can then act as a basis for predicting the increase in stock prices. They compare two extreme methods: focusing on tweets that explicitly state transaction details as well as learning the correlation between the stock price and the tweet’s contents. The second approach removes restrictions on the applicable tweets but with the caveat that a lot of noise is likely to be introduced into the training process. They show that making the process user-sensitive improves the prediction accuracy. The algorithm involves a classifier which classifies a time-annotated set of tweets by each user and classifies each as bullish, bearish or neutral. Each tweet can then be evaluated for correctness by determining that the stock market behaves in accordance with the classification of the tweet. Tweets finally are ranked according to correctness. Another method utilised is unsupervised learning based on the timestamp of the tweet. Using several methods: joint-all model (a single SVM model trained on all tweets), transaction model (finds expert users in based on the correlation of their tweets to the movement of the stock price), per-user model (removes noise by unsupervised learning for each potential expert), joint-experts model (using the per-user model, train a single SVM model). They conclude that the most accurate models are the per-user and the joint-experts (which rely on unsupervised learning) perform the best.

Zhang and Skiena (2010) compare blogs and news as basis for prediction, perform large-scale analysis of the stock market and propose a trading strategy based on sentiment data. Data from Dailies (an aggregator of news), twitter, Spinn3r RSS feeds and LiveJournal was processed by Lydia (a text processing system), resulting in time series consisting of a time series of words and their orientation. They discovered that the media exposure correlates more to the stock market of certain industries (Aerospace and Defence) and less so for others (Software and Computer Services).

AZFinText (Schumaker & Chen, 2005) works with the assumption that 20 minutes after a news article is released, the stock price reflects the effect of the news. As opposed to labelling with a polarity (up or down), it labels using the stock price 20 minutes after the news article is published. The system uses proper nouns as the features and filters the selected features by only further selecting proper nouns that occur three or more times. It uses support vector regression to try to predict the price of the stock in 20 minutes. For a period of 23 trading day, using 2809 articles and predicting for the S&P 500 Index, they achieved a 8.50% return on investment while the S&P 500 Index achieved only a 5.62%.

NewsCATS (Mittermayer & Knolmayer, 2006) categorises news articles into “good”, “bad” and “no movers”. Using a thesaurus as a curator, they classify press releases as good is the maximum gain in 15 minutes after its release is large () and the maximum loss is (). The inverse is the case for “bad” news. Neutral news are classified as those whose gain and loss do not exceed 3%. NewsCATS achieved an average of 0.29% over 2602 trades while a simulated random trader achieved a profit of 0.07% over 2599 trades. Mittermayer and Knolmayer conclude by cautioning that the results do not factor transaction cost and hence do not accurately reflect the potential profit.

## Concluding Remarks

Sentiment analysis has been identified to have a broad range of applications from determining the public sentiment of films to the current application of stock price prediction. It’s quite clear that the problem of sentiment analysis faces different challenges depending on the domain or the dataset source. Our review of the literature has been heavily biased towards techniques that are specifically applied to this project and readers are encouraged to read the survey by Medha et al. (2014) for more complete overview of the domain. In addition, another reasonable conclusion from the literature review is that sentiment analysis is not only a reasonable, quick method of text mining but also can be applied as a step in more complex processes such as stock price prediction.

This concludes our review of the literature on sentiment analysis for stock price prediction.

# **Background Knowledge**

In the previous chapter, we introduced, in passing several key concepts, especially in the context of sentiment analysis. In this chapter, we aim to arm the reader with the key knowledge required to gain an understanding of the rest of this document. In addition, some concepts that were mentioned in the previous chapter are fully explained here. Hence, this chapter (especially section 3.1.) is also a good precursor to chapter 2.

## Sentiment Analysis

Sentiment analysis, according to Wikipedia[[1]](#footnote-2), is defined as “*the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials*.” We have seen in chapter 2 that the possibilities in terms of approaches to the general problem of sentiment analysis are extremely varied; hence, once more we focus on the steps required to a machine learning-based solution. We split our discussion into clear steps that one would expect when performing a machine learning task.

### Feature Generation

Regardless of whether the chosen approach to acquiring labelled is manual or automatic, feature generation from the corpus[[2]](#footnote-3) has to occur. Tokenising is the preferred approach to feature generation. Tokenisation refers to the splitting of a document into single words, phrases, nouns or other parts of speech which are called tokens. Tokens often disregard any form of punctuation. These tokens usually come in form of unigrams, bigrams, trigrams and other n-grams.

Unigrams are n-grams that are of size one. For illustrative purposes, we show the feature generation from a document consisting of only one sentence.

**D1**: The fat cat sat on the fat dog.

**:** The, fat, cat, sat, on, the, dog

Higher-order n-grams were introduced to solve the complete disintegration of any semantic relationship between the terms; unigrams perform quite well, despite this criticism (Pang, Lee, & Vaithyanathan, 2002).

Bigrams are n-grams that are of size two. Bigrams introduce a relationship between the individual words:

**D2**: Standard and Poor changes Goldman recommendation from A to BBB

**:** Standard and, and Poor, Poor changes, changes Goldman, goldman recommendation, recommendation from, from A, A to, to BBB.

In the case of the D2, we can see why bigrams can be beneficial: a classifier might be able to identify the negative orientation of the document due to the bigrams *from A* and *to BBB*. Conversely a classifier might be able to detect the positivity should the bigrams be *from BBB* and *to*. As far as unigram-based classifiers are concerned, there isn’t any difference. While one might think that the higher the order of the n-gram, the better a classifier would be, this isn’t the case. Higher-order n-grams bear more similarity to the source document than an overall pattern; in short, higher-order n-grams become less useful in determining sentiment orientation.

N-grams are the only means of generating features. Features can also consist of noun phrases or pronouns. These are usually accompanied by other parts of speech like adjectives. In order to extract nouns, documents have to be processed by a part of speech tagger such as the one provided by Stanford[[3]](#footnote-4). Even more specific parts of speech such as Proper Nouns or Name Entities[[4]](#footnote-5) can act as features.

As part of feature are the key steps of stop-word removal and stemming. Stop-words are words in the document that occur very frequently across the corpus and often bear no weight in the orientation. Examples of such words in the English language are *a*, *the*, *rather*, *but*, *also*. Textfixer provides a full list of such words[[5]](#footnote-6). Removal of these words is key to reduction of noise across the corpus. In some cases, stop words are key to the determining the orientation of sentences (often relevant at sentence-level classification). For illustrative purposes, the following sentence shows the case in which stop words are useful:

**D3:** If only Mr Tim was a decent actor.

The removal of the stop-words *if* and *only* completely alter the sentiment orientation of the sentence.

Stemming refers to the linguistic normalisation of words from their inflected form to a common form. Words needn’t be mapped to its exact morphological stem, a relation is usually all that’s needed. The example below shows the effect of stemming.

Cooperation, Cooperating, Cooperative, Cooperates ↦ Cooperate

Popular stemming algorithms are the Porter Stemmer and the Snowball Stemming Algorithms. Like stop-word removal, stemming reduces the noise in the features and allows for better performance.

### Feature Representation

In order to make the documents useful to numeric classifiers such as the neural network or support vector machine, the terms need to be transformed into their weighted forms. The weighting of the terms is shown to be at least as important as their selection (Strzalkowski, 1994).

The vector space model introduced by Salton et al. (1975) is the preferred algebraic method of representing textual documents. The document is represented as a single vector where each dimension in the vector is a single feature. Features that do not exist in a particular document are simply given a value of 0. The weights are usually calculated using simply their presence (binary), counts (integer), frequency(float) or their term-frequency- inverse document frequency (TF-IDF) score (float). Hence we can say that in a corpus of documents and features, a document is represented as:

We described an improvement on TF-IDF in chapter 2 and now, we provide the mathematics behind the traditional TF-IDF (Salton, Wong, & Yang, 1975). Given, the document, the weight is calculated as:

where is the number of documents in the corpus, is the number focuments which contain the term . 1 is added to prevent divide-by-zero errors. is the term frequency and is given by the formula:

where is the number of times feature occurs in document and is the total number of terms in the document . The higher the value assigned to a feature, the importance, it is given.

### Feature Reduction

Due to the incredibly large number of features generated by tokenisation, it’s necessary for feature reduction to take place; the preferred methods for this are and singular value decomposition (Golug & Kahan, 1965), although we only discuss based feature reduction.

is used to test the level of independence between the features and the classes. A value of 0 indicate a lack of dependence while a large value implies a large dependence. Features with a value greater than a set threshold are selected. is formulated mathematically by Aggarwal and ChengXiang (2012)as:

where is the number of documents in the corpus, is the conditional probability of documents being assigned to label , given that they contain , is the number of documents assigned label c. and is the number of documents which contain word .

## T-Test Based Split-and-Merge Piecewise Linear Approximation

In their paper, The Predicting Power of Textual Information on Financial Markets, Fung et al, present a means of generating labelled data by detecting trends in a time series and aligning news with the trends; we present their algorithm here.

### The Splitting Phase

The split-phase of the algorithm handles the discovering the trends in a time series while the merge phase helps avoid over-segmentation. Every time series can be represented as a list of tuples – each tuple containing the price and time:

where is the price at for and the time series has a length .

The process starts by representing the trend with a line defined by the first and last points (a segment). In order to decide if the line is enough to represent , a one-tail t-test is defined:

where is defined as the expected mean square error:

where is the number of points in the segment. is the projected price (derived from the line ) at time. The t-statistic is defined by:

Where is the standard deviation. The t-statistic is compared with the t-distribution using a degree of freedom and an . Hence, there is a 0.05 probability of the null hypothesis being accepted given that is incorrect. If is accepted, then the mean-squared error is low and the projected trend line is very similar to the actual time series . If is accepted then, the line is not enough to represent . If is accepted, then the line is split where the error norm is maximum – – resulting in two segments. The process iterated by repeating the process on the two segments. The algorithm is represented in figure 3.1, as produced by Fung et al.

|  |
| --- |
| split(]) – split a time series of of length  from time to time where |
| 1:  2:  3:  4: **for**  to **do**  5:  6: **if**  **then**  7:  8:  9: **end if**  10:  11: **end for**  12:  13: **if** -test.reject **then**  14:  15:  16: **end if**  17: **return** |

Algorithm 3.1 – The Split Algorithm

### The Merging Phase

Over-segmentation is the existence of adjacent two segments whose slopes bear enough similarity to not warrant two separate segments. To fix this issue, the merging phase merges these segments into one large one. The merging phase aims at combining adjacent segment whose merging would not result in being rejected.

Consider a time series (where ) , the result of splitting time series . is a segment in . Should potential merge of the segments and not result in being rejected then they are regarded as candidates for merging. Let the list contain all such pairs of candidates. The selected candidates are the pair that would result in the lest increase in . This process is repeated until the -test is rejected. Hence . The merge phase is shown as provided

|  |
| --- |
| merge() – merge two adjacent segments on time series T |
| 1: **while** *true* **do**  2:  3: **repeat**  4:  5:  6: **if**  **then**  7:  8:  9: **end if**  10: **until** end of time series  11:  **if -**test.accept() **then**  12: drop  13: **else**  14: break  15: **end if**  16: **end while**  17: **return** |

Algorithm 3.2 – The Merge Algorithm

## Support Vector Machines

Techniques in machine learning generally are classified into two main divisions: supervised and non-supervised learning. Support Vector Machines (SVMs) are classified under the former. This means that the process by which SVMs solve tasks is by first undergoing a training phase in which input data (referred to as a set of examples or instances) are used to learn a model that can then be used to classify new instances into a category of a set of categories. SVMs typically solve binary classification[[6]](#footnote-7) problems.

SVMs work by projecting and mapping (by using a kernel function) training data instances into a high-dimensional feature space, in order that the maximum gap between the two categories is created. Hence, new instances can be classified by side of the gap the point falls on. *Gap* is a vague term and is difficult to define. Instead, SVMs work by constructing a hyperplane that separates the two categories – this is referred to as the maximum margin hyperplane. The maximum margin hyperplane is sought as theoretically, such an hyperplane should lead to the least generalisation error.

A hyperplane which defines the decision boundary of the classes is in turn defined in terms of a set of points whose set of points satisfy the equation.

where is the weight vector to the hyperplane, is the kernel method that maps the data points to feature space and is the bias.

For linearly separable data, two hyperplanes can be defined which completely separate the data such that there are no points between them. The area in space that is bordered by the two hyperplanes is called a margin. The two hyperplanes are defined by the following equations:

In order to define and , vectors must be selected which just touch the margins – these vectors are referred to as support vectors as these are the only data instances required to represent the model. The margin can also be defined more formally in terms of and which are the shortest distance between the maximum margin hyperplane and the closest positive support vector and the shortest distance between the and the closest negative support vector, respectively.

Figure 3.1. – Maximum margin hyperplane and margins of a binary classification SVM-based model

In figure 3.1., is defined as the offset between the maximum margin hyperplane and the origin. We can thus say that is the width between and . Given that the distance between and is , the total distance is then . We therefore need to minimise in order to maximise the margin, . This problem can this be formulated mathematically as:

where is an example, is the target class. Minimising can be reduced to

Having formulated the linearly separable case, we note that not all SVM-applicable problems are linearly-separable, in fact most of them are not. For this, we introduce a slack variable which acts as a penalty term for misclassified examples. is thus formulated mathematically as:

Hence, can be rewritten as . This new formulation is referred to as a soft margin and the modified optimisation goal is:

Often, we find that there are more than a single category to classify and there are two main approaches to multiclass classification problems: one-versus-one and one-versus-rest. One-versus-one attempts to differentiate between each two pairs of class while one-versus-many differentiates between a single class and the other classes.

We conclude our discussion of support vector machines with the kernel function, . The resulting feature space is heavily dependent on the exact function mapping and there are a few popular kernel functions, are simply listed here.

1. Linear kernel is defined as
2. Polynomial kernel is defined as where c is a constant which accounts for the influence of higher-order terms versus the lower-order terms is the degree
3. Radial basis function kernel where is an hyper-parameter referred to as the kernel bandwidth.
4. Gaussian kernel:

## Hidden Markov Models(HMM)

Hidden Markov Models (written as )are simply a set of parameters which explain a pattern for a known class or category. HMMs can then be further used to classify a test-pattern for which is has the highest posterior probability. Sequences can be considered as a series of states , written as . There are no restrictions on the number of states to be visited or the number of times a state can be visited. In order to define any sequence, transition probabilities –the probability of at ­ given at - must be evaluated. This is formulated mathematically as

At each step in the sequence, the state emits a symbol . Symbols are otherwise referred to as observations as they are visible. Hence, we might has a sequence of observations **.** Given the states and observation, we can then make another deduction: the probability of observing a certain symbol given the state at time :

|  |  |  |
| --- | --- | --- |
|  |  | Equation 3.1 |

is thus referred to as the emission probabilities. At each time step, a transition must occur and a symbol must be emitted, resulting the redundant formulations:

|  |  |  |
| --- | --- | --- |
|  | for all  for all | Equation 3.2 |

There are three main problems addressed with the Hidden Markov Model: The Evaluation problem, the decoding problem and the learning problem.

### Evaluation

Given an HMM, the transition and emission probabilities, evaluate the probability of a sequence being generated by the model:

|  |  |  |
| --- | --- | --- |
|  |  | Equation 3.3 |

The problem can be solved by sum of all possible sequences of the conditional probability of the transitions multiplied by the emission probability the sequence. However, this is a computation intensive procedure. Instead the forward algorithm (Algorithm 3.3) is used to solve the problem. For the forward algorithm, we define , which defines the probability that an HMM is in state at step having already generated the first elements of the sequence .

The time-inverse algorithm (Algorithm 3.4) can also be used to solve the evaluation problem, and it’s necessary to define :

|  |
| --- |
| forward() – calculate recursively |
| 1: **initialise** , visible sequence,  2: **for**  3:  4: **until**  5: **return**  for the final state  6: **end** |

Algorithm 3.3 – the forward algorithm

|  |
| --- |
| backward() – calculate recursively |
| 1: **initialise**, visible sequence  2: **for**  3:  4: **until**  5: **return**  for the known initial state  6: **end** |

Algorithm 3.4 – the backward algorithm

### Decoding

Suppose we have a observations sequence and a model **,** determine the most likely sequence of hidden states that generated the observations. The Viterbi algorithm is used to solve this problem (Algorithm 3.5.)

|  |
| --- |
| decoding() – determine the most likely |
| 1: begin **initialise**  2: **for**  3:  4: **for**  *j*  5:  6: **until**  7:  8: Append to Path  9: **until**  10: **return** Path  11: **end** |

### Learning

Given the number of states and the number of visible states and a set of training observations, determine the emission and transition probabilities. Starting estimates of and , we can calculate improve our estimates by first calculating the probability of transition from to given by :

where defines the probability that the model generated . The improved estimates and can then be used in the forward-backward algorithm (also known as the Baum-welch algorithm):

and:

|  |
| --- |
| Forward-backward() – determine and |
| 1: begin **initialise** , , training sequence , convergence criterion ,  2: **do**  3: compute from and by eq 140  4: compute from and by eq 141  5:  6:  7: **until**  8: **return**  9: **end** |

## Hybrid SVM-HMM Model

The variation in hidden markov models comes in the form of which the emission probabilities are calculated; hence, HMMs can be based on the Poisson distributions, Gaussian distributions or Gaussian mixture models and in this case SVMs. For continuous data, Gaussian mixture models are typically the preferred method for deriving the emission probabilities; however, they are often criticised for their poor discriminatory capabilities. Conversely, SVMs are popular for their great discriminatory capabilities.

SVMs do not directly output probabilities, instead they output the measured distance between the example data and the hyperplane: . In order to create a link between the posterior probability of a instance having a class : and the result of the SVM, we need to utilise Platt’s (1999) proposal for generating pseudo-probabilities by the use of a sigmoid function:

Where and are derived by using maximum likelihood estimation fron a training set. And is the unthresholded output of the SVM defined by:

In the binary case, we can thus compute the posterior conditional probabilities and of classes and given the symbol () . Finally using Bayes’ rule, we can compute the emission probabilities from the outputs of the SVM using the prior probability calculated from the frequency of the class in the training data:

# Scientific Method

## Method Overview

## Sentiment Classification

As emphasised in previous sections, the first task to be performed is the classification of news articles. As with classification of other types of data, the following steps need to be performed: data acquisition, data labelling, data pre-processing, data analysis and then finally classification. The range of human sentiment is very wide and includes sentiment such as happiness, sadness, calmness, anger, anxious. This range is much too wide for the application at hand. In fact, we have taken a much simpler approach and simply classified the sentiment of the news articles into three categories: happy, sad and ambivalent.

Unlike sentiment classification for domains such as films, cars or music, happiness, sadness and ambivalent news articles may not bear much information about the progress of the entity. While the sentiment of the article is what we aim to extract, a cursory look at any news article that bears financial information will show that news articles aren’t very sentimental. This indicates that classification based simply on human sentiment while might be accurate might not be as successful given that we aim to predict the stock market. Therefore, to supplement classification based on sentiment, we also classify based on the progression of the company. This means that the articles get classified into an additional set of categories (positive, negative and neutral).The aim with the progress classification is that articles get classified based on what the news article evaluator expects that the entity’s stock price will go up or down or simply stay the same.

This new direction of evaluation however raises the question of how to gauge the effect of a news article. For illustrative purposes, if the price of a specific stock has been on the rise for the past three days and then a news article is released and it’s classified as “positive”, how do we factor that in? Does it simply not matter as we have a direction of progress or could we instead watch for the rate of the change of the stock price? We aim to answer some of these questions in the following sections and the over the next chapter.

### Data Acquisition

#### News Articles Acquisition

It’s clear that the very first task to be performed is the acquisition of news articles whether labelled or non-labelled. Although a fair bit of work have been done using this particular approach to stock price prediction, we were unable to find any publicly available datasets that fell in line with the purposes of this dissertation. Hence, a dataset was generated using from online news sources.

Although selecting news sources seems like a trivial task, it requires careful consideration as the news sources has to be able to satisfy the following requirements:

1. Has to be popularly read, especially by traders. This is particularly important because a high level of trust needs to be placed in the news source, enough to determine that significant changes in stock price trend will be reflected in the news articles.
2. The news sources has to have decent coverage of the news sources, again to ensure that we gather as much data as possible.

These requirements are particularly important especially when we consider that we will be aligning news articles with stock prices. Selecting news sources that report events a few days after the fact might skew results as by then the market is sure to have absorbed the new information and any relevant changes in (rate of change of ) price will be missed. Given these requirements, investors on online forums (as well as individuals with knowledge of finance) were asked which news sources were read and the following sources were given: Reuters, Bloomberg, Financial Times, Market Watch, Yahoo Finance.

The next step is to scrap selected websites (Bloomberg, Reuters) for news articles. We do not discuss the exact process of scraping websites as it’s not relevant to this project. However, scraping can involve interesting problems such as logging in to websites via a program (in this case, python) and extracting data. We do not use all of this data however because of time constraints – there’s no way for a single person to manually label the 12000 articles in the time frame of 4 weeks. Hence, we had to discard a lot of the news articles and aim for classifying a fraction of the news articles (about 2000 articles) – we note also that the number actual number of articles which are used during classification is further reduced to 1690 after preprocessing – this is further discussed in section 4.2.3. The zip file accompanying this document contains all code for scraping the websites.

Prior to the scraping of news articles, we must first determine what it is we hope to find – in this case, we want to scrap enough news articles in order to perform classification on the news articles. Hence the gathering of news articles has to be targeted so that we have enough data for each of the companies we aim to classify. There is therefore a need to determine which companies we aim to classify. The companies selected were chosen from the Dow Jones Industrial Average (DJIA) because the companies listed on the index are major American companies which tend to get a lot of attention from the media.

Finally, the news data is extracted and put in the following xml format:

<?xml version='1.0' encoding='us-ascii'?>

<news>

<entry author=''…'' datetime=''…'' url=''…''>

<headline>

…

</headline>

<body>

…

</body>

</entry>

</news>

Figure 4.1 – XML Format for Scrapped News Articles

#### Stock Price Data Acquisition

The acquisition of the stock data simply involves selecting the companies which are of interest and extracting the data from Yahoo Finance. In this case, the API (ystockquote) was used to extract the data for relevant companies. The data collected for each company includes the following: Date, Adjusted Close, Close, High, Low, Open, and Volume. As interesting as the use of the intraday stock price is, there are other complex aspects of the project and the introduction of intraday stock prices would mean the consideration of other aspects of both finance and implementation details that would only increase the difficulty of the project. An example of such an issue would be determining the period after which we can say a news article’s effect has taken place. On a day-by-day basis, we assume that at the end of the working day, the effects of a news article should already be reflected in the movement of the stock price. In addition, a day-by-day basis means that any projected trend line generated does not have to account for the minute movements during the day. [Check back on this.]

### Data Labelling

At the start of the project, the intention was to crowd source the labelling of the articles. This would be done by asking individuals with knowledge of economics and finance to evaluate the news articles. This involved uploading the corpus to a website for easier classification. (sentimentanalysis.bolanleonifade.me). However, the rate at which the articles were getting classified was very slow so the alternative approach taken was to use prior knowledge to label the article. This of course meant that experiments might suffer due to lack of enough financial knowledge. Therefore, in order to provide a baseline or at the very least, a means of evaluating the manually labelled data, a set of automatically labelled data was created as well.

* + - 1. **Manual Labelling**

Manual labelling of data is simply reading each news article and labelling them by hand. The evaluators are asked to estimate the company’s progression based on the news article. Their estimates can fall into the following categories: (up, down, neutral). The evaluators are also asked to provide the sentiment of the article (happy, sad, neutral). From henceforth, for clarity purposes, we shall refer to the former as “progress sentiment” and the latter as “feeling sentiment”

One might think that there is a perfect correlation between the two sets of categories. However, there can be differences between the two. For illustrative purposes, we will examine a few cases in which there are differences:

The headline “Exxon Mobil reports Fire, oil spill at Nigeria’s terminal”, evokes a feeling of sadness but due to the established nature of Exxon, it’s unlikely that this event is going to lead to a massive dent in the stock price, we give the article a progress sentiment of neutral (because the article doesn’t go on to indicate that Exxon will suffer from this incident). Another article that wouldn’t be expected to change the stock price much is “JP Morgan falls to death from building roof in Hong Kong”. It’s however clear that the article is “sad”

If the previous two examples give the impression that from headlines, we can always tell the feeling sentiment of an article, it would be wrong. In fact, a seemingly neutral headline such as ”Coca-cola names Waller Finance Chief as Fayard Retires” goes on to discuss the recent struggles of coca-cola, therefore giving it a feeling sentiment of sad and a progress sentiment of neutral. In the same strain, we discovered articles can both be up for progress sentiment and feeling sentiment; this would be the case for articles that discuss an entity’s growing business.

Given these apparent differences in the labelling of the progress and feeling sentiment, one might be concerned that the one of those sentiments is useless. We find out that this isn’t the case when the perform analysis on the dataset (Section 5). In addition, we find out

* + - 1. **Automatic Labelling**

In order to perform automatic labelling of news articles, we need to generate projected trends, this gives us an idea of the overall outlook of the stock price – that is, for example, we can safely say that the overall projected trend of the stock price is an upwards movement if the price over a period of time has changed positively, ignore every minor dip in the price trend. We can perform automatic labelling by using piecewise linear approximation (described in section 3 and the results of which are discussed in section 5) and proposed by Fung et al. (2005). This allows us to align news articles with the projected stock price and simply labelling the articles based on the projected stock price.

There are obvious inaccuracies that can occur from the use of such a method – in fact, as shown in the literature review, classification based purely on price differences, tends not to be very accurate – this is because one cannot say for sure that all articles released during periods of overall upwards price movement are positive and vice versa. However, we are operating under the assumption that news articles strongly reflect the direction of movement of the stock market. We expect the price to move up when news articles discuss increases in sales, innovation, positive restructuring and we expect the price to move downwards when news articles discuss fines, bankruptcy, legal problems, sanctions etc.

Automatic labelling, therefore provides us with a baseline. The higher the similarity between the results of automatic labelling and that of manual labelling, the more *trust*, we can place in the results of manual labelling.

Another issue of note with automatic labelling is the fact that automatic labelling cannot be classified based on feeling sentiment. Feeling sentiment by definition requires an evaluator to label articles based on what feelings are evoked by reading the article. However, since the method by which we automatically label stock data is based on the progression of the stock price, we can automatically label articles based on stock price trends.

* + - 1. **Evaluating Labelled News Articles**

Sentiment labelling generated via automatic categorisation is a reflection of the price movements, not a reflection of the articles themselves. However, since the articles themselves are manually labelled to reflect precisely the sentiment which they carry, we can conclude that if there exists a high similarity the results of manual labelling is to the results of automatic labelling, then we can say confidently that the labelled articles can led to positive results in later classification. It’s important to note however, that any comparison that we do can only be by comparing progress sentiment of automatically labelled articles with progress sentiment of manually labelled articles (as opposed to both progress and feeling sentiment of manually labelled articles).

How therefore do we evaluate the results of labelling? An easy method of doing is is by calculating the Pearson’s correlation coefficient (already discussed in section 3). We comprehensively discuss the results of the calculations and the other considerations (specific to calculating the correlation) when discussing the evaluations in section 5). We finish off this section by pointing out that the more similar the correlation values are between the projected trend line (generated via piecewise linear approximation) and the actual stock price trend line, the higher the similarity is between the two trends. Hence, this also provides additional validation for the results of piecewise linear segmentation.

### Data Pre-processing

When we first retrieve data and to display them, the news articles are kept in HTML format. The news articles therefore need to be tokenised (into unigrams and bigrams), stemmed and have stop words removed (section 3) before classification can take place. All these methods have been extensively discussed in section 3 but we summarise them here.

Tokenisationis the splitting of words into smaller groups whilst ignoring all punctuation. We perform tokenisation of news articles into unigrams, bigrams and unigrams and bigrams. In order to perform tokenisation we use nltk’s word\_tokenize. Before tokenisation, we ensure that the words all have the same casing (in this case, lowercase). Stemming is performed using nltk’s implementation of the Porter stemmer. This allows for removal of words that mean the same. Stop words don’t bear enough information to be useful for classification purposes and these are removed by using the scikit-learn in-built capabilities.

After tokenisation, feature extraction from the corpus must take place. Tokenising – the process by which features are extracted – 2242 separate documents can lead to quite a lot of features, most of which are unlikely to be useful for classification even when stop word removal is taken into account. Therefore, a step that’s carried out in pre-processing is selecting the max\_features based on term frequency is a parameter used in order to select only the features that are most likely to be informative during the classification process.

### Document Representation

After the completion of all pre-processing steps, the documents are now ready to be transformed into vectors. The library Scikit-learn provides the TfidfVectorizer that converts the news articles into tf-idf-weighted document-term matrix. Tf-idf has been discussed in (section 3). The process thereby transforms document into number – ready to be used by a classifier – after feature selection or feature reduction. Using all the features can lead to very low accuracy so in this step, we already pre-select the number of features to use. The number of features we select depends on tokens of the documents (bigrams, unigrams or both). It’s important to note that this is not feature selection as is performed later on; this pre-selection is implemented by scikit-learn and allows us to pre-select only the highest tf-idf scores. We discuss the results in section 5.

### Feature Selection or Reduction

In the literature, the chi-squared method for feature selection and the SVD method are popular and we decided to use test these two methods against each other. We very briefly compare the results of both methods in section 5. Section 3 provides an in-depth explanation of both methods. The library Scikit-learn provides the class TruncatedSVD class as well as chi2 function for performing feature selection and feature reduction respectively. SVD is a very expensive method that in practice using our relatively small corpus of approximately 2500 document resulted in training time of about 2 hours. Clearly this is unacceptable for most purposes and while the compare the results of both methods all things being equal, chi2 is much less time consuming giving it an edge over SVD.

### Classification

The finally step is to train the SVM to predict the news article. We assume that the problem is linear (and instead we briefly compared the linear SVM with other types) and the linear SVM seemed to perform best (discussed in section 3). In order to truly evaluate the SVM, cross validation over the data set was performed. 10 fold cross validation ensured we got an average of all performance of the hyper-parameters of the SVM. The results of each fold were evaluated using the following metrics: confusion matrix, recall, precision and f-measure. These results can then be averaged over the number of folds to determine the overall performance of the hyper-parameters. This therefore completes our discussion of the process of sentiment classification of the news articles. Section 5 details the results.

## Price Prediction

# Evaluation and Results

## Method Overview

In this section, we use detail the results of each of the main activities that have been discussed in section 4. The format of this section is the same as that of section 4, this is for easy referencing and comprehension. As there are a lot of possible data, we try as much as possible to reduce the data to only that which illustrates the general idea or result and any redundant but interesting information is put in the appendix.

## Data Acquisition

### News Article Acquisition

As we have already discussed in section 4.2.1.1 the exact process of acquisition, in this section we simply delve into the results. The table below shows the number of articles that were gathered for each company. Please note that the numbers reflected here are numbers after pre-processing had been performed.

|  |  |  |
| --- | --- | --- |
| Company Name | Number of Articles | |
| Chevron | | 88 |
| Cocacola | | 52 |
| Disney | | 108 |
| Exxon | | 120 |
| Goldman | | 731 |
| IBM | | 118 |
| JP Morgan | | 613 |
| Microsoft | | 259 |
| Pfizer | | 115 |
| Visa | | 48 |

Figure 5.1 – Number of articles collected for each company

### Stock Data Acquisition

The price daily values were collected for the period from the 1st of January, 2013 to the 30th of September, 2014. Of course, the stock price data is only released for working days so this accounts for only 440 working days.

## Labelling

### Manual Labelling

We show the results of manual labelling in this section. Principally, this comes in the form of the Pearson’s correlation coefficient carried out between the stock price and the aggregate sentiment. The aggregate sentiment is calculated simply by adding the sentiment for every previous day in the time series. In order to do this, we need to calculate the aggregate sentiment for each working day. This is simply done by adding together all the sentiment values for each article released in the working day.

Supposing therefore that we have 4 articles released on any day labelled as such: (please refer to the appendix – section 7.1 – for how this is derived from the sentiment labels) using progress sentiment. The aggregate sentiment would be given as by adding all the sentiments together. In addition to this, supposing we have any set of 8 days for which the aggregate sentiment for each day is given, we have aggregate sentiments for each day to be the sum of previous sentiments:

We’ve aggregated the sentiments to better show the rise and fall of the sentiment (gotten from the news articles). With this transformed sentiment, we calculate the correlation between the stock price closing values and the sentiment. The table below shows the correlation values between each company’s stock data and the aggregated sentiments.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Company Name | Correlation of Progress Sentiment(Actual) | | Correlation of Feeling Sentiment(Actual) | Correlation of Progress Sentiment(Projected) | Correlation of Feeling Sentiment(Projected) |
| Chevron | | 51.7 | 51.8 | 49.7 | 50.1 |
| Cocacola | | 32.8 | 32.6 | 32.99 | 32.84 |
| Disney | | 97.6 | 97.6 | 97.10 | 97.13 |
| Exxon | | 79.0 | 78.8 | 79.7 | 79.6 |
| Goldman | | 76.74 | 77.42 | 71.98 | 72.5 |
| IBM | | 53.2 | 53.3 | 51.7 | 51.8 |
| JP Morgan | | 84.2 | 84.4 | 84.5 | 84.7 |
| Microsoft | | 95.8 | 95.8 | 95.8 | 95.8 |
| Pfizer | | 51.1 | 51.6 | 50.9 | 51.3 |
| Visa | | 89.14 | 89.20 | 88.80 | 88.86 |

Figure 5.2 – Table of correlation between sentiment and stock price using manually labelled data

Looking at the table above as well as figure 5.1, there are several points raised regarding the values shown. We attempt to discuss some of them here but leave others until we show the results of automatic labelling. In addition, we also generate correlation coefficients between the projected trend lines and the sentiment – for purposes of comparison with the results of automatic labelling. We have chosen to a few of the figures that show the results of piecewise linear approximations in the succeeding section as the results are more relevant there. However, please refer to the appendix for the rest of the figures. [Appendix Reference]

We see that in almost all cases, the values of the correlation projected values are almost always lower than the correlation between the actual stock prices. It is expected that there would be a difference between the correlation between the two pairs of values but assuming that piecewise linear approximation generates trends as accurately as possible, the difference between the correlation values should never be large enough to raise questions. Exxon as we can see has correlation coefficients that are larger in the projected trend line than in the actual trend line. A possible reason for this is that news articles capture better changes in prices over a longer period of time than the daily changes in price. [Check this out later]

In addition, we can see that although feeling sentiment and progress sentiment are intended to be different measures of labelling articles, they lead to in all cases, similar correlation coefficients. However, we note that in almost all cases, progress sentiment has a higher correlation value than projected sentiment.

Finally, we address the fact that something must be done for those days in which news articles are not released (this applies as well when calculating the correlation for automatically labelled data). There are three assumptions we can make when classifying based on progress:

1. There is an underlying feeling of positivity – that is, these cases, the stock price is assumed to go up meaning that the progress sentiment is always positive
2. There is an underlying feeling of neutrality – the stock price will stay the same when there’s no news.
3. An underlying feeling of neutrality – the stock price will go down when there’s no news

This can of course have drastic effects on the correlation values because in most cases, the number of articles released is significantly lower than the number of days in the time period. While it might seem that an underlying feeling of neutrality is the most appropriate, this is not true. The underlying feeling depends on the company being discussed. Prices of most companies change positively when there’s no news hence we assume an underlying feeling of positively. However, for companies such as IBM, there’s an underlying feeling of negatively. This is determined by simply looking at the overall projected trend line. IBM over the course of the time period has shown a gradual decrease in price. This is sentiment is well reflected in articles as IBM over the course of the time period had trouble keeping up with other technology firms who have incorporated cloud computing into the services offered. The complete set of trend lines are in the appendix – [Appendix Reference].

News articles aren’t released solely during weekdays; however stock markets are closed over the weekend. This doesn’t mean that trading does not occur over the weekend – extended hours trading does – however prices changes aren’t released for the weekend. Hence, in order to calculate the correlation, we have to eliminate news articles that are released on days that fall on weekends. This of course is detrimental to our calculations but despite this we see that there still is a high correlation between stock prices and news articles.

### Automatic Labelling

The first few steps needed to be carried out are similar to the steps carried out for manual labelling – summarily, calculating the aggregate sentiment of the news articles over the time period. As previously mentioned in section 4.2.2.2, we only calculate the progress sentiment of the news articles. The sentiment is then correlated both with the projected trend and the actual price trend. Figure 5.3 and 5.4 show generated projected trend lines as well as the actual stock price movements for Disney, Exxon and Pfizer – similar figures for other companies are shown in the appendix – [Appendix Reference].

Using the Disney trend line as a sample case, we show how news articles are classified based on where they fall in the projected trend line – Figure 5.5

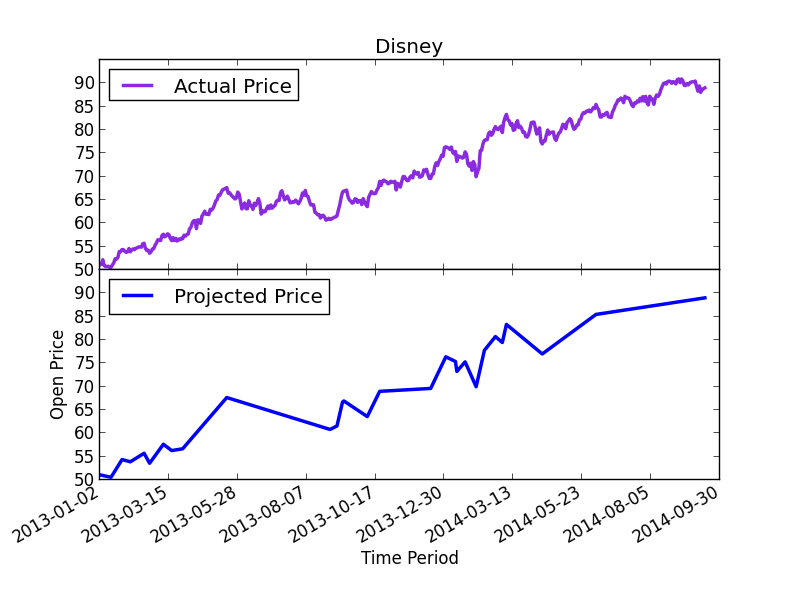


Figure 5.3 – Actual Stock Price and Projected Price of Disney

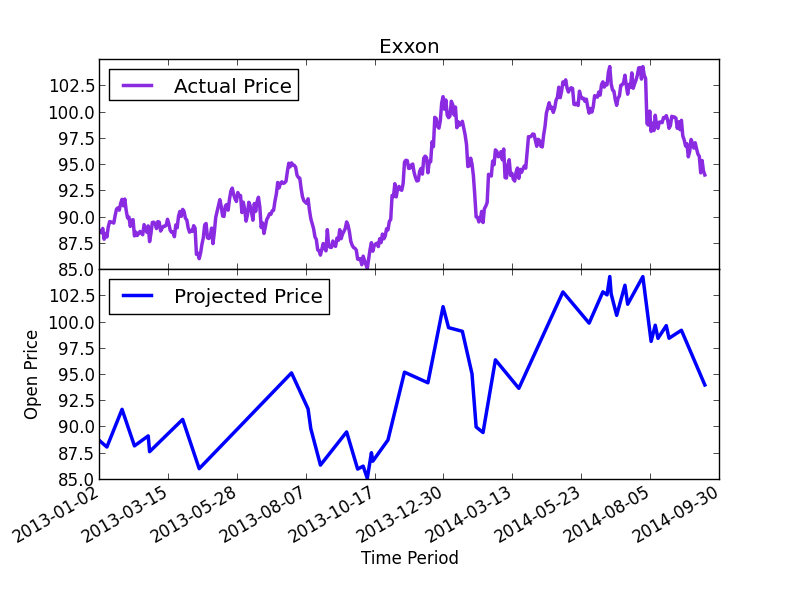


Figure 5.4 – Actual Stock Price and Projected Stock Price of Exxon

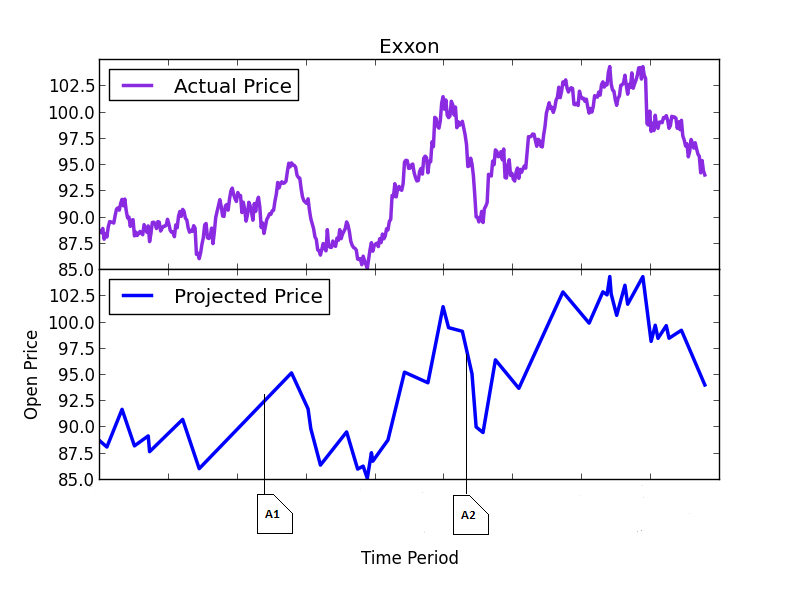
****

Figure 5.5 – Aligned News Articles with Trends

Using Figure 5.5 as a sample case, we would classify article A1 as “up” or positive while A2 would be classified as “down” or negative. After achieving this step, we can then proceed to calculate the correlation between the automatically calculated prices and the labelling. One would assume that it would be a correlation value of 1 as labelling in fact is done with the trend line; however, this is not so as news articles aren’t released every day. The table below shows the correlation results.

|  |  |  |  |
| --- | --- | --- | --- |
| Company Name | Correlation of Progress Sentiment(Actual) | | Correlation of Progress Sentiment(Actual) |
| Chevron | | 50.2 | 51.8 |
| Cocacola | | 32.1 | 32.0 |
| Disney | | 97.0 | 97.5 |
| Exxon | | 80.4 | 79.7 |
| Goldman | | 50.0 | 52.60 |
| IBM | | 52.12 | 53.41 |
| JP Morgan | | 85.37 | 85.14 |
| Microsoft | | 94.99 | 94.99 |
| Pfizer | | 53.08 | 53.33 |
| Visa | | 89.27 | 89.62 |

Figure 5.6 – Correlation values of the automatically generated data

Looking at the data above and comparing them, we see that the correlation values are quite similar to that of the manual data except for the correlation values of Goldman Sachs which is vastly different from the result of manual classification. The reason for this is that Goldman Sachs often is the source of the news (for example, Goldman Sachs often advises on buying and selling other companies) and the news is not about Goldman Sachs. This means that automatic labelling is blind to these issues as it labels both news by Goldman as well as news about Goldman without using any filter. However, when manually labelling, we ensure to label those articles are “neutral” in terms of progress of the entity and the relevant feeling sentiment.

### Labelling Discussion

We see that the results of labelling is in general very positive – providing a reason to go on with classification of the news articles – preferably using the manually labelled data. The high similarities between the progress coefficient of the automatic and manual data also is a form of validation for the manually labelled data and idea that news articles correlate with the stock price. However, the results also make harder to overlook the issues with automatic labelling.

## Data Pre-processing

Using the method described in section 4.2.3, the dataset was pre-processed. However, there’s a decision to be made about which method of tokenisation is best. In our experiments, we performed only 3 types – unigrams, bigrams and combination of both. These types are the most popular in the literature. Unigrams, also known as bag of words are criticised often for not bearing enough information but we see that in all areas, they perform quite well. Bigrams, as we will see also perform comparatively to unigrams. The combination seems to perform the best of all three. In addition, when tokenising, we only select words that have greater than or equals to three characters.

The table below shows the number of features that are extracted and the number pre-selected (based on term frequency) before feature extraction or selection. These values were selected via trial and error, matching the various parameters with results and selecting the parameters that produced the highest result.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unigram | Bigram | Unigram + Bigram |
| Initial number of features | 26322 | 310660 | 336982 |
| Selected amount of features | 11000 | 30000 | 30000 |

Figure 5.7 – Initial features and pre-selected features

## Feature Selection or Feature Reduction

During the experimentation phase, we compared the results of SVD with chi2 based feature reduction. The results are, very similar with no distinct advantage provided by SVD (rather, it’s disadvantageous as it took up to 2 hours to compute bigrams), we decided to go opt for feature selection based on chi2, hence the classification results in the succeeding section detail the results based on feature selection and not feature reduction simply because the results are virtually the same. However, we provide the settings used for SVD and chi2 (Figure 5.8) as well as confusion matrixes of classification using both methods (Figure 5.10).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unigram | Bigram | Unigram + Bigram |
| Chi-2 | 5000 | 17000 | 17000 |
| SVD | 4000 | 13000 | 13000 |

Figure 5.8 - Number of Features depending on whether feature selection or extraction is used

Examining figure 5.8, we see that more features are utilised for a comparative result – however, the amount of time taken to perform extraction (45 minutes for unigrams and 2 hours for bigrams and unigram + bigrams) makes it unfeasible when performing classification. This will be even more pronounced as one of the issues with the current project is the lack of data - a realistic system would expect to use many more thousands of news articles.

In addition to the benefits provided by chi2, there are also benefits in terms of ability to examine the features selected more closely. In figure 5.9 and 5.10, we show 10 tokens selected for both the progress and feeling sentiment (please note that these aren’t in any particular order – instead the table is as a result of words selected across all folds during cross-validation).

|  |  |  |
| --- | --- | --- |
| Unigram | Bigram | Unigram + Bigram |
| aaa | accord announced | aaa |
| surplus | zero percent | abnormal |
| illegal | capital declined | analysts predict |
| destroy | company profit | exciting |
| asian | stock buyback | growing market |
| litigation | Creditworthiness decreased | Largest technology |
| embezzlement | stock gained | Lawsuit jpmorgan |
| examination | wall street | laundering |
| fine | volatility index | straight year |
| value | legal claims | breach contract |

Figure 5.9 – Words selected for progress classification (manually labelled data)

|  |  |  |
| --- | --- | --- |
| Unigram | Bigram | Unigram + Bigram |
| acceptable | aaa credit | abandon |
| grossing | aaa rated | billion asset |
| questionable | abc network | cash flow |
| rallied | income drop | government bond |
| suppress | percent called | shrank percent |
| disagree | seek boost | trading stock |
| distort | later acquire | state law |
| expense | gain ground | jpmorgan led |
| investigation | dollar bond | Exclusive |
| policy | compliance action | development |

Figure 5.10 – Words selected for feeling classification(manually labelled data)

We have neglected to include the corresponding tables for the automatically labelled data as automatically labelled data had lower overall accuracy than manually labelled data. In addition, it should be noted that the words shown in either table are not necessarily exclusive to the table. For example, “jpmorgan led” which appears in the 8 row of the Unigram + Bigram column also appears as a bigram feature when classifying based on progress sentiment.

## Document Classification

### Manual Classification

In this section, we detail the results of manual document classification and the settings used to achieve the results. As we use the linear classifier, the parameters that need to be set are the class weights and the cost. Other parameters to be set are default parameters by the classifier. First, discuss the classification of progress sentiment and show the results, followed by the classification of feeling sentiment. To finish up, use only data for a single company (Goldman Sachs) for classification. This is to determine whether or not classification for a single entity is much better than group classification. It’s highly likely that this might be the case as a classifier is more likely to be fine-tuned to the exact negativity and positivity of the news articles about the company.

#### Progress Sentiment Classification

In order to set the weights, we need to look at the support for each class. Figure 5.11 shows the number of articles supporting each class (Figure 5.11). The hyper-parameters used for configuring the SVM are as follows:

LinearSVC (the class used for classification) implements the one-versus-rest classifier for multiclass problems. The penalty term is l2 rather than the typical (as performed better for this problem) used for linear SVMs. We use the value of “” for the parameter. [need equation reference]. This setting is however unusual as often, requires quite large numbers. Class weights are set automatically by LinearSVC. The values set the parameter for each of the classes, dependent on the class frequencies. This needs to be set as Figure 5.11 shows, the classes are not represented equally in the training sets. Using a StratifiedKFold cross validator, we can preserve the percentage of representation for each sample.

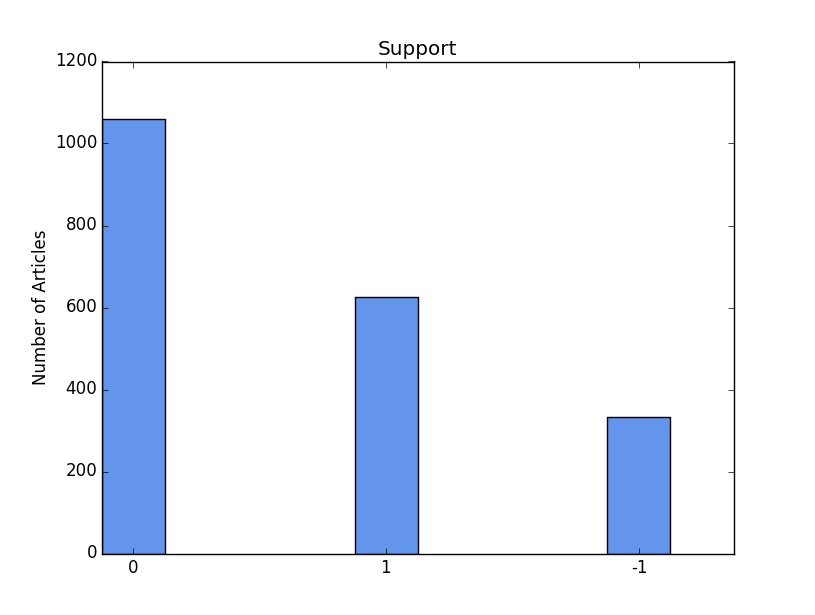
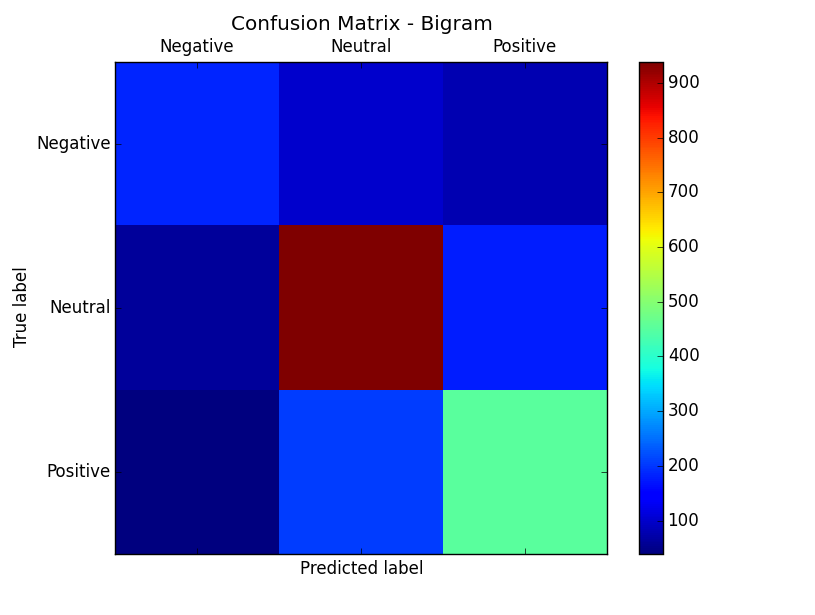
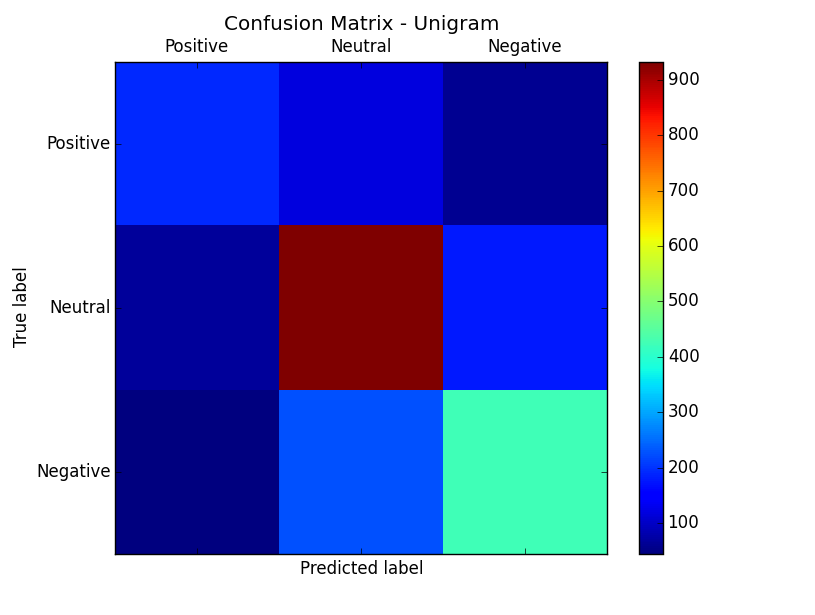


Figure 5.11 – support for the various classes (manual/progress)

Similar settings were used for bigrams, unigrams and combination experiments. In order to determine accuracy, we use cross validation and the following matrices: f-measure, recall, precision and confusion matrix. We compute the average of the scores of all the folds. Please note that we results we show here are for feature selection rather than extraction. In order to view the results for SVD-based feature extraction, please refer to the appendix.The confusion matrixes for unigram, bigram and combination (Figure 5.12) shows an overview of the accuracy for the three classes. For actual numerical values, please refer to the appendix.

The confusion matrixes show that there aren’t very big differences in the performances of the three methods of tokenisation (except when classifying positive articles). It’s very difficult to explain why this is the case except that all three methods carry similar levels of information for this problem.



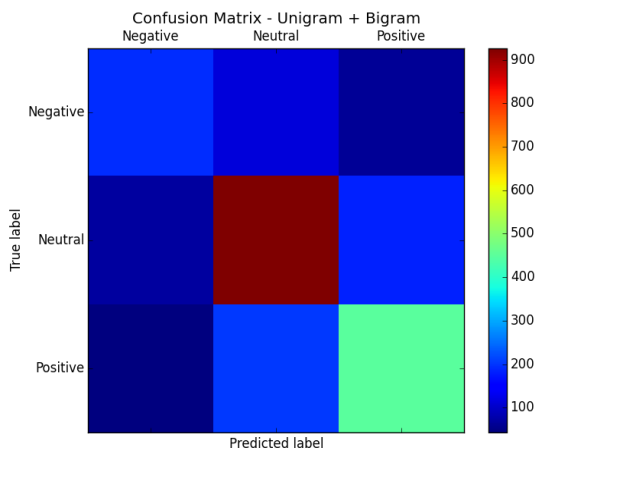


Figure 5.12 – Confusion matrices (Manual/Progress). Top left – Unigram, Top Right – Bigram, Bottom – Unigram + Bigram

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unigram | Bigram | Unigram + Bigram |
| F-measure | 68.62 | 69.82 | 70.51 |
| Recall | 68.68 | 69.97 | 70.58 |
| Precision | 69.04 | 70.17 | 70.62 |

Figure 5.13 – Table of performance of linear SVM measured by cross validation (manual/progress)

Delving into the actual numbers, we see that overall, the bigram does better than the unigram and the combination of both does better than either of them singularly. Combining this information with the confusion matrix, we see that bigrams and the combination perform better due to being able to slightly classify positive news articles better.

#### Feeling Sentiment Classification

Poorer results were achieved for the classification of feeling sentiment in general. This is contradictory to the initial belief that feeling sentiment would be easier than progress sentiment to classify. We performed classification using a linear SVM as before. The settings for feeling sentiment were quite different. In addition, classification performance for the feeling sentiment was quite poor overall. As per the previous section, we start by introducing the frequencies for the classes (Figure 5.14)

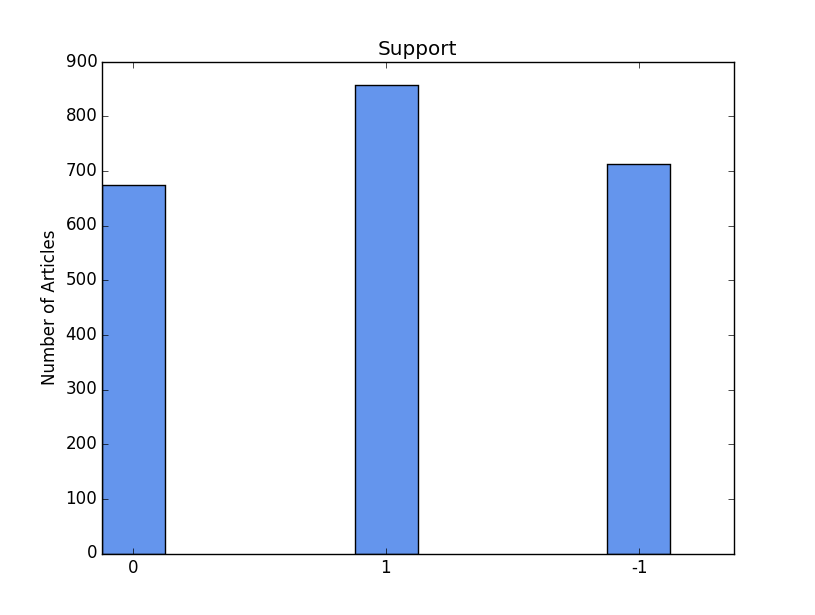
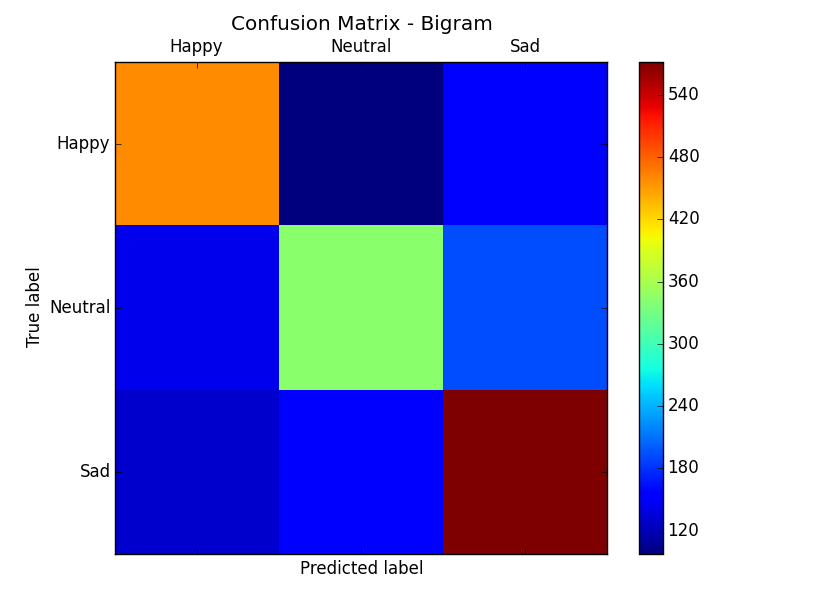
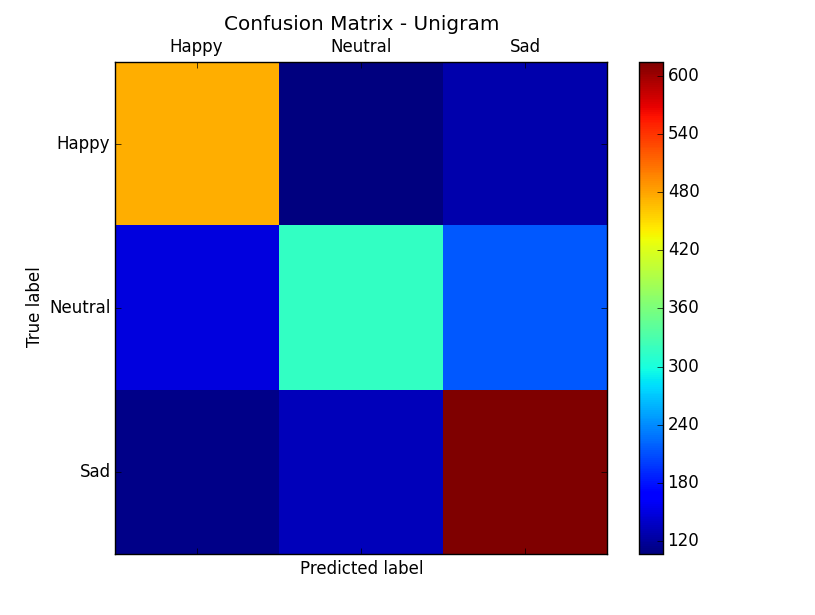


Figure 5.14 – Support for the classes (Manual/ Feeling)

Different hyper-parameter settings were used for classification. The parameter was set to higher levels with a value of. The other parameters, such as the class weight were also set automatically based on the class. We use a loss function of and penalty function of .

Considering the confusion matrices (Figure 5.15), we see that the all three methods of tokenising perform very similarly as before. A possible reason for this is that news articles often bear mixed feelings. On the surface, it may seem that news articles bear feeling sentiment orientations that lean towards one way or the other but this isn’t so. News articles often carry information that lean to both sides. A classic example of such news articles is articles that discuss “happy” sentiment. In a few of these articles, there’s also discussion of past “sad” sentiment that led to perhaps structural changes that result in improvement. Hence, while progress sentiment might be clear, feeling sentiment can often be ambiguous when it comes to classifying neutral articles.



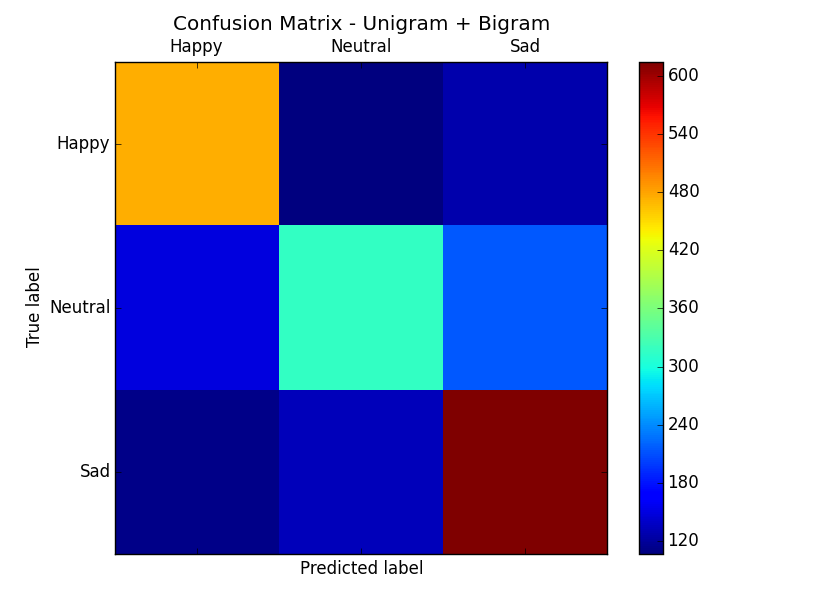


Figure 5.15 – Confusion matrices (Manual/ Feeling). Top left – Unigram, Top Right – Bigram, Bottom – Unigram + Bigram

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unigram | Bigram | Unigram + Bigram |
| F-measure | 62.00 | 60.87 | 63.68 |
| Recall | 62.48 | 61.06 | 63.92 |
| Precision | 62.23 | 61.15 | 63.99 |

Figure 5.16 –Table of performance of linear SVM measured by cross validation (Manual/ Feeling)

Bigrams are typically expected to do better than unigram due to the fact they retain sentence structure but clearly, bigrams doesn’t do very well for this problem looking at the performance measures in Figure 5.16. However, combination of both performs better than either but not by much.

### Automatic Classification

In this section, we follow the same pattern as in section 5.6.1, with the exception that we only discuss progress sentiment. For automatic classification, neutral movements are underrepresented (only about 46 news articles were classified neutral); hence, we only consider positive and negative movements.

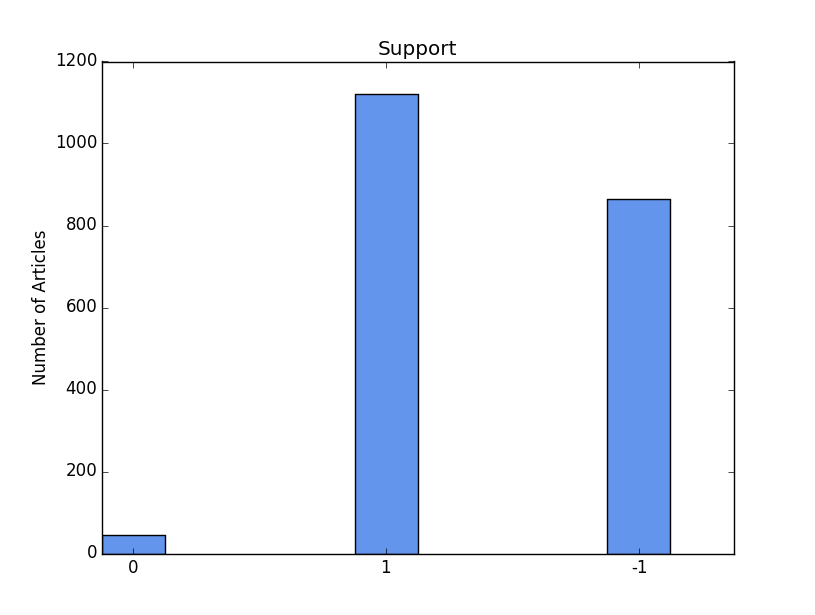


Figure 5.17 – Support for the classes (Automatic / Progress)

Figure 5.18 - Table of performance of linear SVM measured by cross validation (Automatic/ Feeling)

Here, we only present the numerical results to restrict repetitiveness as well as the fact that the manually labelled data will be used (Figure 5.18). It would of course be interesting to consider how well automatically labelled data performs when used subsequently for price prediction, however time restraints prevent this. In addition, the intention was not to use automatic labelling for classification; instead, it provided an adequate benchmark for comparison with the results of manual labelling. We have extensively discussed the pitfalls or issues automatic labelling is susceptible to and we believe that the results here can be explained by these (Reference to discussion 5.3.)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unigram | Bigram | Unigram + Bigram |
| F-measure | 64.50 | 66.30 | 64.49 |
| Recall | 67.97 | 72.55 | 61.36 |
| Precision | 61.43 | 61.12 | 68.05 |

# Conclusion

## Keys for Transforming Sentiments to Numbers

# Appendix

1. Definition acquired from: *en.wikipedia.org/wiki/Sentiment\_analysis* [↑](#footnote-ref-2)
2. Wikipedia (*en.wikipedia.org/wiki/Text\_corpus*) defines a corpus as a large and structured set of texts. [↑](#footnote-ref-3)
3. The Stanford Natural Language Processing Group provide a copy of their tagger at nlp.standford.edu/softwaretagger.shtml [↑](#footnote-ref-4)
4. Name entities are similar to proper nouns but refer to specific entities. Dates, organisations, places or numerical data can serve as name entities. [↑](#footnote-ref-5)
5. Common stop words in the English Language at www.textfixer.com/resources/common-english-words.txt [↑](#footnote-ref-6)
6. Binary classification involves the training of a model to differentiation between only two classes. [↑](#footnote-ref-7)