DEPARTMENT OF COMPUTER SCIENCE

Semantic Analysis of Financial Headlines Based on Realised Stock Returns

Research

Joshua Felmeden

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Saturday 16th April, 2022

Abstract

A compulsory section, of at most 300 words

This section should précis the project context, aims and objectives, and main contributions (e.g., deliverables) and achievements; the same section may be called an abstract elsewhere. The goal is to ensure the reader is clear about what the topic is, what you have done within this topic, *and* what your view of the outcome is.

The former aspects should be guided by your specification: essentially this section is a (very) short version of what is typically the first chapter. If your project is experimental in nature, this should include a clear research hypothesis. This will obviously differ significantly for each project, but an example might be as follows:

> My research hypothesis is that a suitable genetic algorithm will yield more accurate results (when applied to the standard ACME data set) than the algorithm proposed by Jones and Smith, while also executing in less time.

The latter aspects should (ideally) be presented as a concise, factual bullet point list. Again the points will differ for each project, but an might be as follows:

- I spent 120 hours collecting material on and learning about the Java garbage-collection sub-system.
- I wrote a total of 5000 lines of source code, comprising a Linux device driver for a robot (in C) and a GUI (in Java) that is used to control it.
- I designed a new algorithm for computing the non-linear mapping from A-space to B-space using a genetic algorithm, see page 17.
- I implemented a version of the algorithm proposed by Jones and Smith in [6], see page 12, corrected a mistake in it, and compared the results with several alternatives.

Dedication and Acknowledgements

I am very fortunate to work with my superviser Rami Chehab, who assisted in my research greatly and inspired the vast quantity of this project. He has been a great mentor.

I would also like to thank my friends and family for always being there for me, especially when the going gets rough. You guys really helped make this university experience what it was.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

Joshua Felmeden, Saturday 16$^{\text{th}}$ April, 2022

Contents

List of Figures

List of Tables

Ethics Statement

A compulsory section

In almost every project, this will be one of the following statements:

- "This project did not require ethical review, as determined by my supervisor, [fill in name]"; or
- "This project fits within the scope of ethics application 0026, as reviewed by my supervisor, [fill in name]"; or
- "An ethics application for this project was reviewed and approved by the faculty research ethics committee as application [fill in number]".

See Section 3.2 of the unit Handbook for more information. If something went wrong and none of those three statements apply, then you should instead explain what happened.

Supporting Technologies

- I used a sample of headlines from Kaggle as training and validation data (https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests)
- I used the Natural Language Toolkit to assist the preprocessing of data, using their English words and stop words data (https://www.nltk.org/)

Notation and Acronyms

NLP	:	Natural Language Processing
SESTM	:	Semantic Extraction via Screening and Topic Modelling

SESTM Specific Notation

m	:	Number of words in sample
n	:	Number of articles in sample
$d_{i,j}$	:	Number of times word j appears in text i
$d_{[S],i}$	:	Subset of columns where the only indices are those with sentiment
$D = [d_1, \ldots, d_n]$	:	$m \times n$ Document term matrix
$sgn(y)$	:	Sign of returns of article y
$\hat{x}$	:	Expected value of variable x

Chapter 1

Introduction

Financial news is a widely available resource from which many investors and businesses alike glean information. It is an insight into how many different businesses function and the health of markets. If properly used, can be taken as a measure of ... Analysing the sentiment of an article has been used for a significant amount of time and is utilised for a myriad of purposes, including forecasting financial stocks.

An article is naturally formed of two parts, the headline and the article body. The bulk of the information conveyed by a given article is in the body, however, since headlines are often a summarisation of the body, it has been proven that data mining from the headline itself can be at least as useful as mining the body {citation needed}. The average word counts of a headline are very low, however, the vocabulary is often much higher impact on average, as this is what initially grabs the attention of a reader.

1.1 What to do

This chapter should introduce the project context and motivate each of the proposed aims and objectives. Ideally, it is written at a fairly high-level, and easily understood by a reader who is technically competent but not an expert in the topic itself.

In short, the goal is to answer three questions for the reader. First, what is the project topic, or problem being investigated? Second, why is the topic important, or rather why should the reader care about it? For example, why there is a need for this project (e.g., lack of similar software or deficiency in existing software), who will benefit from the project and in what way (e.g., end-users, or software developers) what work does the project build on and why is the selected approach either important and/or interesting (e.g., fills a gap in literature, applies results from another field to a new problem). Finally, what are the central challenges involved and why are they significant?

The chapter should conclude with a concise bullet point list that summarises the aims and objectives. For example:

> The high-level objective of this project is to reduce the performance gap between hardware and software implementations of modular arithmetic. More specifically, the concrete aims are:
>
> 1. Research and survey literature on public-key cryptography and identify the state of the art in exponentiation algorithms.
> 2. Improve the state of the art algorithm so that it can be used in an effective and flexible way on constrained devices.
> 3. Implement a framework for describing exponentiation algorithms and populate it with suitable examples from the literature on an ARM7 platform.
> 4. Use the framework to perform a study of algorithm performance in terms of time and space, and show the proposed improvements are worthwhile.

Chapter 2

Background

We begin by discussing the technical background that relate to the work in the thesis. This project evaluates and analyses the success of semantic analysis when applied to financial headlines; making use of realised stock returns as a teaching signal.

2.1 Semantic Analysis

Semantic analysis (also known as opinion mining) is the task of identifying opinions or sentiment of authors from an input of text. The ramifications of being able to programmatically extract the intended meaning of an input is extremely powerful in a variety of fields and is particularly prudent in a financial context. The nuances of natural language can sometimes make this difficult to extract, and therefore significant research has been conducted on the topic over the course of the last decades.

2.1.1 Lexicon Based Methods

The fundamental concept revolves around investigating a piece of text, and deciding on a binary classification: positive or negative. The simplest method compiles a list words with weights. Each weight corresponds to the positivity of the word (for example 'great' would have a high positivity, while 'terrible' would have very low). The overall sentiment of a piece of text could then be estimated by summing the individual sentiment scores of each word. The dictionary is not limited to single words, and can be expanded to include n-grams (phrases of n words), as the context in which a word is used can dramatically change the sentiment. This may increase the accuracy of the dictionary at the cost of increased dimensionality. The dictionary itself must be compiled before it is possible to utilise this method. One of the most widely used for English text is the Harvard-IV-4 TagNeg (H4N) lexicon which is a general usage model that can be used to estimate the sentiment of a piece of text.

There are many difficulties faced with creating a dictionary of this sort, as language is often a subjective entity, leading to conflicting opinions in assigning tone to a specific word. Furthermore, the context within which a word is used can drastically change the intended sentiment of a word, for example, the word 'great' is naturally a very positive word, however, if used in a sarcastic manner (e.g. 'It's so great that my flight is delayed!') can invert the sentiment entirely. For this reason, simply summing the sentiment of a piece of text on a word by word scale can give an incorrect estimate.

Certain words that may have no meaning at all in one context, may have significant sentiment in another, particularly in the field of finance, where jargon dominates text. Loughran and McDonald [LOUGHRAN and MCDONALD, 2011] conducted an investigation into the use of standard lexicons for use in analysing 10-K filing reports, which are comprehensive reports filed by companies about their financial performance. They discovered that almost 75% of negative words found in the filings based on the H4N file were not typically negative in a financial context. For example, 'liability' is a negatively charged word in a standard context, while it carries no tone at all in the context of the filings. Loughran and McDonald created their own dictionary based on these results called the Loughran McDonald dictionary that is created for the purpose of classifying 10-k filings. For these reasons, lexicon based methods work far better if the it is bespoke for the topic at hand, needing a specific dictionary for each task.

2.1.2 Usage

Usage of these lexicons can vary greatly, with some simply summing the count of positive words and subtracting the count of negative words to end up with a simple overall word count. However, this is a very trivialised method of use and can be expanded to be far more constructive. Loughran and McDonald themselves suggest using Term Frequency Inverse Document Frequency (TF-IDF) as a method to weight the word counts alone. TF-IDF is one of the most popular term weighting schemes, and it therefore makes sense to augment these strategies with this weighting.

TF-IDF has two distinct parts, term frequency and inverse document frequency. Term frequency (TF) is the relative frequency of a term t in a document d:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d, t' \neq t} f_{t',d}} \tag{2.1}$$

where $f_{t,d}$ is the raw count of term t in document d. This is one method of calculating the term frequency, however there are many variations, such as logarithmically scaled frequency ($\log(1 + f_{t,d})$), boolean frequency (1 if t appears in d at all, and 0 otherwise), etc.

Inverse document frequency measures how much information a word carries by its rarity. The generic equation is as follows:

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2.2}$$

where $N = |D|$ and $D = [d_1, d_2, \ldots, d_n]$.

Using these two components in conjuction, the final TF-IDF of a term is:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \tag{2.3}$$

2.2 Semantic Extraction via Screening and Topic Modelling

Semantic Extraction via Screening and topic model (SESTM) is a novel text mining algorithm that makes us of a teaching signal developed by Ke et Al. in 2019 [Ke et al., 2019]. It makes use of stock returns as a teaching signal to develop a model of sentiment words in maximally positive or negative arguments in order to be able to predict the sentiment of out of sample articles.

To begin, we assume that each headline has some sentiment score $p_i \in [0,1]$, with $p_i = 1$ being a headline with maximum positive sentiment, and $p_i = 0$ maximally negative. Our problem is ot model the distribution of returns of a headline y_i given the sentiment of the headline p_i. Thus we have assumed:

$$\mathbb{P}(\text{sgn}(y_i) = 1) = g(p_i), \text{ for a monotone increasing function } g(\cdot) \tag{2.4}$$

where $\text{sgn}(x)$ returns 1 if $x > 0$ and -1 otherwise. This assumption simply states that the higher the sentiment score, the higher the probability the headline has of returning positive returns.

Next, we observe the distribution of word counts in an article. We assume a vocabulary has partition

$$\{1, 2, \ldots, m\} = S \cup N \tag{2.5}$$

where S is the set of sentiment charged words and N is the set of sentiment neutral words. Thus, the distribution of sentiment charged words are the vector $d_{[S],i}$, similarly for sentiment neutral words $d_{[N],i}$, although sentiment neutral words serve as useless noise and remain unmodelled.

Utilising the work of Hoffman, we adapt latent semantic analysis (LSA) which is a technique that maps high-dimensional word count vectors to a lower dimensional representation (in our case, the realised returns) [Hofmann, 2013]. We then assume that the vector of sentiment charged word counts are generated by a mixture multinomial model of form

$$d_{[S],i} \sim \text{Multinomial}(s_i, p_i O_+ + (1 - p_i) O_-) \tag{2.6}$$

where s_i is the total count of sentiment charged words in headline i. O_+ is the probability distribution and simply describes expected word counts in a maximally positive headline (namely $p_i = 1$). Similarly, O_- describes expected word counts in maximally negative headlines ($p_i = 0$). Fundamentally, this model is the probability of counts for each sentiment charged word in our vocabulary, with s_i trials and the outcome is modelled by $p_i O_+ + (1 - p_i) O_-$.

Using these two probability distributions, we can also gain insight into vectors of tone T and frequency F:

$$F = \frac{1}{2}(O_+ + O_-), \quad T = \frac{1}{2}(O_+ - O_-) \tag{2.7}$$

2.2.1 Screening for sentiment charged words

The first step in this algorithm is to screen for sentiment words in a collection of articles, since sentiment neutral words are a nuidance and contribute to noise. This strategy isolates the sentiment charged words and uses these alone to calculate sentiment. This is achieved by using a supervised approach with the realised returns of an associated stock, since if a word appears in headlines that result in positive returns, it is reasonable to assume that the word carries positive sentiment. Some notation will be introduced here to facilitate the discussion and explanation of the algorithm.

- The sample is defined as n articles producing a dictionary of m words.
- The word count of article i is recorded in vector d_i
- $D = [d_1, d_2, \ldots, d_n]$ is an $m \times n$ document term matrix
- The count of sentiment charged words in article i is defined as the submatrix $d_{[S],i}$.

For a word j, we define f_j as the fractional representation of the frequency with which a word appears in a positively tagged article versus the frequency with which a word appears in any article:

$$f_j = \frac{\text{count of word } j \text{ in article with } sgn(y) = +1}{\text{count of word } j \text{ in all articles}} \tag{2.8}$$

Here, sgn is simply the sign of the difference in tagged returns of an article. If an article is released on day t (specifically, between 4pm of day $t-1$ and 4pm of day t), the article is tagged with the returns of the associated firm from day $t-1$ to $t+1$ (specifically market close on day $t-2$ to close on day $t+1$).

Next, f_j is compared against positive and negative threshold. We defined $\hat{\pi}$ to be the fraction of articles that have $sgn(y) = +1$, or articles that are tagged with positive returns. In practice, as the sample size increases, this value tends towards 0.5. For sentiment neutral word, it is expected that $f_j \approx \hat{\pi}$ with some variance either side. Therefore, two thresholds are set α_+ and α_-. These thresholds are set such that any word with $f_j \geq \hat{\pi} + \alpha_+$ is determined to be sentiment positive, and the inverse for words such that $f_j \leq \hat{\pi} - \alpha_-$. This filtering technique accepts words that are further away from the expected average sentiment, meaning only those words with extreme sentiment are included. To ensure that words appearing infrequently do not skew sentiment values significantly (for example a word appearing in exactly one article that is maximally positive ending up with $f_j = 1$), we introduce a third parameter κ, restricting words that appear in fewer than κ headlines. The number of articles in which word j appears is defined as κ_j The set of sentiment charged words is therefore defined by:

$$S = \{j : f_j \geq \hat{\pi} + \alpha_+ \cup f_j \leq \hat{\pi} - \alpha_-\} \cap \{j : k_j \geq \kappa\} \tag{2.9}$$

The three parameters introduced here $(\alpha_+, \alpha_-, \kappa)$ are tuned via cross-validation and the best configuration for each is selected.

2.2.2 Learning Sentiment Topics

With the wordlist S calculated, we can now fit a two-topic model to each of the sentiment-charged counts. We introduce matrix $O = [O_+, O_-]$ that determines the expected probability of sentiment charged words in each article using a supervised learning approach. The teaching signal in this case is the realised three day returns of the associated firm for each headline.

In the model, p_i is the headline's sentiment score, described by:

$$p_i = \frac{\text{rank of } y_i \text{ in } \{y_l\}_{l=1}^n}{n} \tag{2.10}$$

Where the return of a headline is represented by y, and headlines are ranked in ascending order (meaning an article with the highest returns would have p_i of 1). More concretely, the returns are calculated as discussed above, and represented as a percentage. Let the price of the associated firm on day t be y_t, the three day returns would be calculated as $y_{t+1}/y_{t-1} - 1$.

Once this value is calculated, let $h_i = d_{[S],i}/s_i$ denote the $|S| \times 1$ vector of (sentiment charged) word frequencies for article i. Using model 2.6, we know that the distribution of word frequencies can be modelled as:

$$\mathbb{E}h_i = \mathbb{E}\frac{d_{[S],i}}{s_i} = p_i O_+ + (1 - p_i)O_- \tag{2.11}$$

or in matrix form:

$$\mathbb{E}H = OW, \quad \text{where } W = \begin{bmatrix} p_1 & \cdots & p_n \\ 1-p_1 & \cdots & 1-p_n \end{bmatrix}, \text{ and } H = [h_1, h_2, \ldots, h_n] \tag{2.12}$$

We are now able to estimate O via regression of H on W. H is not directly observed due to S being unobserved, so we estimate H by plugging in S:

$$h_i = \frac{d_{[S],i}}{s_i} \quad \text{where } s_i = \sum_{j \in S} d_{j,i} \tag{2.13}$$

W is estimated using the ranks of returns described in 2.10, leading to the final representation of:

$$O = [h_1, h_2, \ldots, h_n]W'(WW')^{-1}, \quad \text{where } W = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \\ 1-p_1 & 1-p_2 & \cdots & 1-p_n \end{bmatrix} \tag{2.14}$$

Finally, O may have negative entries, so we set these to zero and normalise each column to have ℓ^1-norm. The resulting matrix is referred to as O to simplify notation.

2.2.3 Scoring New Headlines

The previous steps give estimatorss S and O. Using these, we are able to estimate p for a new article that is not in the training sample. Using model 2.6, we estimate p using Maximum Likelihood Estimation. This is simply testing values in some range and determining which value gives the maximum output. We also include penalty term $\lambda \log(p(1-p))$ and finish with the following optimisation:

$$\widehat{p} = \arg\max_{p \in [0,1]} \left\{ \widehat{s}^{-1} \sum_{j \in S} \log(pO_{+,j} + (1-p)O_{-,j}) + \lambda \log(p(1-p)) \right\} \tag{2.15}$$

where $\widehat{s}$ is the total count of word from S in the new article, $d_j, O_{+,j}, O_{-,j}$ are the entries for word j in the corresponding vectors and λ is a tuning parameter that ensures that the majority of headlines are neutral by pushing the estimate towards a neutral sentiment score of 0.5.

2.3 Portfolios and Financial Market Analysis

Evaluating word lists generated by any method for sentiment analysis can be done by creating portfolios based on news with the highest counts of positive or negative words and buying the most positive stocks while selling the most negative. This gives a good indication of the predictive power of a word list.

2.3.1 Portfolios

A portfolio is simply a list of stocks that can be invested in by an individual or firm. The returns from a portfolio is defined as the profit accrued from all stocks over a set time period, usually daily, monthly or annually. Due to the nature of stocks, simply listing the returns as a concrete value does not convey the information required. For example, if stock 'A' were to be invested in at value \$50, and it rose to \$60 the following day, the returns could be said to be \$10. However, if stock 'B' were valued at \$1000, and the following day it rose to \$1010, the monetary value would be equivalent at \$10, but the percentage return is vastly different; 20% returns for stock 'A' and 1% for 'B'. For this reason, returns from portfolios are expressed as a percentage.

Daily returns can be very marginal, as the time period is very small, often being smaller than 1%. In the interest of readability, *basis points* (also known as bps or bips) are used in lieu of a percentage, where 1 bip is equivalent to 0.01%. This makes it much easier to represent very small returns as are common in daily returns.

Creating a portfolio

A portfolio is constructed using a number of stocks and can be either bought (taking the 'long' position) or sold (taking the 'short' position). For stocks that are bought, the returns can be calculated from the difference in price at the time that the stock is sold. More concretely, if a stock has value S_t at time t,

and held for n days before being sold, the long returns in percentage form can be calculated using the following formula:

$$\frac{S_{t+n}}{S_t} - 1 \tag{2.16}$$

Similarly, for short returns, as the stock is being sold, profit is acquired if the stock falls in value, therefore the returns can be calculated using the following formula:

$$\frac{S_t}{S_{t+n}} - 1 \tag{2.17}$$

Of course, these simple formulae negect transaction fees that can apply when constructing real portfolios. However, as we are creating portfolios in a theoretical sense, this suits our needs.

Once the portfolio has been constructed, the weighting for each stock must be considered. Each portfolio will have a value, which is the amount of money invested into it, and each stock will in turn get an investment that is a percentage of this overall value. There are two strategies that we will consider: equal weighted and value weighted strategies. Equal weighted is very simple: if a portfolio is comprised of n stocks and has some investment v, each stock has v/n invested into it. This strategy glosses over differences in stock size or price. On the other hand, value weighted portfolios assign much more money to stocks with higher value associated to them. This can be calculated in a number of ways, but the way we calculate this is if stock s_i in portfolio $P = [s_1, ..., s_n]$ has market value $P_{i,t}$ at time t, the weight of stock s_i would be:

$$w_i = \frac{p_{i,t}}{\sum_n^1 p_{n,t}} \tag{2.18}$$

The amount invested into stock s_i would then be $w_i \times v$.

Equal weighted stocks more closely resemble hedge funds, as well as being the case that smaller companies are able to more quickly encapsulate market share and investor interest. Equal weighted investments ensure a portfolio has a higher representation of smaller stocks, at the higher risk of the stock failing. Conversely, value weighted portfolios tend to be safer, as they prioritise larger companies that are more stable. The downside to utilising this method is that the large percentage increases observed in smaller stocks will have less effect, and therefore some profit can be lost.

2.3.2 Evaluating a portfolio

To successfully determine the success of a portfolio, it is not always as simple as observing the profit alone. While this is a good indicator of the potential returns that could be gleaned from a given portfolio or investment method, it is important to consider external factors and risks that may be involved. The following methods are used to provide more insight into an investment method.

Sharpe Ratio

William Sharpe created the sharpe ratio in 1966 and is one of the most referenced comparison of risk versus return in finance [Sharpe, 1966]. The formula for this ratio is exceedingly simple — one of the key factors in its wide usage — and is as follows:

$$S(x) = \frac{r_x - R_f}{\sigma(r_x)}$$

where x is the investment, r_x is the average rate of return of x, R_f is the risk free rate of return, and $\sigma(r_x)$ is the standard deviation of r_x. The risk free rate of return is simply the theoretical rate of return on an investment with absolutely no risk. Subtracting these risk free returns from the average rate of returns of x yields the true rate of returns.

The value of an investment's Sharpe ratio measures the performance with adjustment for risk: the higher the ratio, the better the performance of the investment when adjusted for risk. As a reference, a ratio of 1 or higher is good, 2 or better is very good, and 3 or better is excellent.

Fama French 3 and 5 Factor Models

Eugene Fama and Kenneth French co-authored a 1992 paper detailing risk factors in returns on both stocks and bonds. This extends the work Sharpe completed on the Sharpe ratio and goes further in exploring risk factors in returns, along with the capital asset pricing model (CAPM) [Fama and French, 1992]. CAPM is used for describing systematic risk and expected return, especially for that in stocks. The equation for this is:

$$ER_i = R_f + \beta_t(ER_m - R_f) \tag{2.19}$$

where ER_i is the expected return of the investment, R_f is the risk free rate, β_i is the beta of the investment and $ER_m - R_f$ is the market risk premium. The beta of an investment is the volatility compared to the rest of the market. It encompases the sensitivity of a stock to changes in the market. In essence, this gives the expected returns of an asset based on systematic risk. Building on this, Fama and French observed two additional risk factors: the size premium of an asset, or small minus big (SMB), and the value premium, or high minus low (HML). SMB is used to account for companies with small value stocks that generate high returns, while HML accommodates for stocks with equity that is valued cheaply compared to its book value that generate higher returns in comparison to the rest of the market. These factors are used in conjunction to provide the following formula for the Fama French 3 factor model (FF3 model):

$$ER_i = R_f + \beta_1(ER_m - R_f) + \beta_2(SMB) + \beta_3(HML) + \alpha$$

The values for SMB and HML are available from French's website [French, 2022], and can be collected for daily, monthly, or yearly returns. Computing this model on a series of returns from a portfolio gives useful information on the nature of the returns, since the model explains part of the returns. The values for the βs detail the exposure to exposure to each of the risk factors while the α, or the intercept, refers to the amount that a portfolio outperformed the expectations of the FF3 model. This alpha is representative of the amount of private or new information that is external from the market and is utilised to construct the portfolio. This means that more simplistic methods of selecting portfolios will not have any private information and therefore the intercept will be much close to zero while a more complicated model will have a larger intercept as simple methods have profits that can be explained by these risk factors.

This model was then revisted by Fama and French in 2015 where they observed two additional factors: robust minus weak (RMW) and conservative minus aggressive [Fama and French, 2015]. RMW corresponds to the profitability of an asset, in that it is the difference between returns of robust or high and weak or low operating profitability. CMA corresponds to the investment factor, and is the difference between returns of conservatively investing firms versus aggressively investing firms, giving the updated formula:

$$ER_i = R_f + \beta_1(ER_m - R_f) + \beta_2(SMB) + \beta_3(HML) + \beta_4(RMW) + \beta_5(CMA) + \alpha \tag{2.20}$$

Chapter 3

Project Execution

In this section, we present the execution of the project, giving an overview of the dataset used, the configurations of hyperparameters and programming completed.

3.1 Dataset and Pre-Processing

The dataset used for training and validation is available from Kaggle[1] and is a collection of around 1.4 million headlines for 6000 stocks from 2009 to 2020. Each headline has the date published, and the ticker that the headline concerns. Some headlines have multiple tickers associated with them, but each ticker-headline combination is another entry in the dataset.

The first challenge is to align these headlines with the relevant three day returns, and this was achieved using the Yahoo Finance python library. For each unique stock in the dataset, market data for the entire 11 year timespan is pulled. Next, a lookup table is computed: for each day t, market close on day $t-1$ and market close $t+1$ is added, facilitating the retrieval of these values. Some headlines are released on non-market days (such as weekends), and for these edge cases, the next available market day is selected as day t, and then the previous market day from this new day t is defined as day $t-1$. Similarly, for market days where $t+1$ would fall on a non-market day, the next available market day is defined as day $t+1$. Finally, each headline is iterated through, assigning the appropriate market close values, and stored in json format for future usage. An example json entry is shown below in listing 3.1 where 'open' refers to market value of a ticker day $t-1$ and 'close' refers to market value on day $t+1$.

Note that some tickers do not have publicly available stock market information for the entire span of the sample, as they are private companies, and therefore headlines aligned with these private tickers are removed from the sample, leaving around 1 million articles.

With the headlines aligned to the appropriate returns, the text data must be preprocessed to allow for successful and efficient semantic analysis. Taking the text content of each headline, the following transformations are applied:

- Convert the headline to lower case
- Remove non alphabetic characters
  - Spaces are retained
  - Convert new line characters into spaces
- Remove non-English words [2]
- Remove stop words [3]
- Lemmatise each word (for example converting 'likes' to 'like' or 'trying' to 'try') [4]
- Tokenise the headline (convert to list of words)

[1] https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests
[2] The list of English words is available from item 106 from https://www.nltk.org/nltk_data/
[3] The list of stopwords used is from item 86 from https://www.nltk.org/nltk_data/
[4] The lemmatisation process uses WordNet (item 106) from https://www.nltk.org/nltk_data/

```
{
        "headline": Barclays Maintains Equal-Weight on Agilent
        Technologies, Lowers Price Target to $76
        "date": 2020-03-26
        "ticker": A
        "mrkt_info": {
                "open": 67.0
                "close": 70.9100036621
        }
}
```

Listing 3.1: Example JSON headline entry

- Convert to bag of words (BOW) representation (list of unique words with associated word counts for a given headline)

The decision was made to not include stemming, which converts English word to the stem, as the stem can sometimes be confusing. For example, the stem of the word 'rates' is 'rat'. Thus lemmatisation was deemed sufficient for preparation.

3.2 Training the Model

Using the BOW representation of each of the headlines, the data is able to be processed according to the algorithm outlined by Ke et al. In the spirit of the original paper, the dataset is divided up into 17 three year training and validation windows, where two years are used for training a model, and the final year is used for validation purposes. More concretely, the training sample begins in 2010-01-01 and ends on 2018-12-31, the validation sample begins in 2012-01-01 and ends in 2018-12-31, leaving articles between 2019-01-01 and 2020-06-08 as an out of sample dataset used for testing the robustness of the model. All the computation is completed on this window, and then it is moved forward four months and repeated in a rolling window method.

The training section employs the screening (2.2.1) and learning (2.2.2) steps, while the validation is the application of the scoring new headlines (2.2.3) step. The validation section is used to consider the hyperparameters $(\alpha_+, \alpha_-, \kappa, \lambda)$, and these are evaluated according to a fixed number of possibilities. $\alpha_\pm$ is calculated such that S has either 25, 50, or 100 words of each sentiment (i.e. for 25, $|S| = 50$). κ is selected to be the 86, 88, 90, 92 or 94th percentiles of word counts. Note that the κ restraint is applied first such that a word is not selected via the α constraint that must then be removed due to the κ constraint, leaving S with fewer words than desired. Finally, λ is selected to be either 1, 5, or 10, for a total of 45 configurations. Each of these 45 configurations is iterated through for each window, and the ℓ^1 error is calculated for each, before selecting the setup with minimum error (as this is our loss function). ℓ^1 error in this case is simply:

$$\sum_{i=1}^{n} |\widehat{p_i} - p_i| \tag{3.1}$$

where $\widehat{p_i}$ is the estimated sentiment and p_i is the standardised return rank of article i in the validation set. The loss function of ℓ^1-norm error was selected for its robustness. The entire process of training and validation takes a considerable length of time and therefore some time was spent optimising the code. A complete list of optimisations can be found in appendix (REFERENCE)

Table 3.1 details the results of the completed rolling window training. Due to the nature of news, some validation sets are larger than others, leading to skewed summed ℓ^1-norm error. To accommodate for this variation is sample size, the error is taken as an average over all headlines in the sample. Window 2011-1-1 has the smallest minimum error, but also has the smallest validation sample size and after controlling for this factor, window 2013-5-1 is has slightly lower error, meaning this is the optimum window.

3.3 Out of Sample Testing

Using the optimally trained model (shown in table 3.1), the articles not used in either training or validation samples are then used to determine the strength of the model. Each market day t, the headlines released

Window start date	$\|S\|/2$	α_+	α_-	κ	λ	Minimum error	Avg Min Error
2010-1-1	100	0.0757	0.0902	88	5	20407.82	0.24733
2010-5-1	100	0.0613	0.0716	90	10	20724.68	0.24939
2010-9-1	25	0.0888	0.0981	94	5	20489.27	0.24750
2011-1-1	25	0.1023	0.1114	92	5	20054.78	0.24686
2011-5-1	25	0.1052	0.1133	92	5	19171.12	0.24722
2011-9-1	100	0.0456	0.3831	94	5	19300.80	0.24711
2012-1-1	100	0.0796	0.0765	88	5	20512.66	0.24790
2012-5-1	100	0.0572	0.0468	92	10	21856.76	0.24980
2012-9-1	100	0.0557	0.0431	92	10	22273.73	0.24965
2013-1-1	50	0.0641	0.0651	94	5	21631.78	0.24799
***2013-5-1**	**50**	**0.0996**	**0.1172**	**88**	**5**	**22211.64**	**0.24681**
2013-9-1	100	0.0709	0.0812	88	10	23407.62	0.24877
2014-1-1	100	0.0405	0.0385	94	5	24860.11	0.24859
2014-5-1	100	0.0873	0.1004	86	5	24194.88	0.24789
2014-9-1	100	0.0412	0.0412	94	5	23539.46	0.24907
2015-1-1	100	0.0499	0.0579	92	5	22821.52	0.24827
2015-5-1	100	0.0491	0.0591	92	5	23627.83	0.24855
2015-9-1	100	0.0613	0.0774	90	5	26349.57	0.24761
2016-1-1	100	0.0498	0.0630	92	5	29938.69	0.24809

Table 3.1: Best configuration and error for each window. Smallest error window highlighted in **bold**

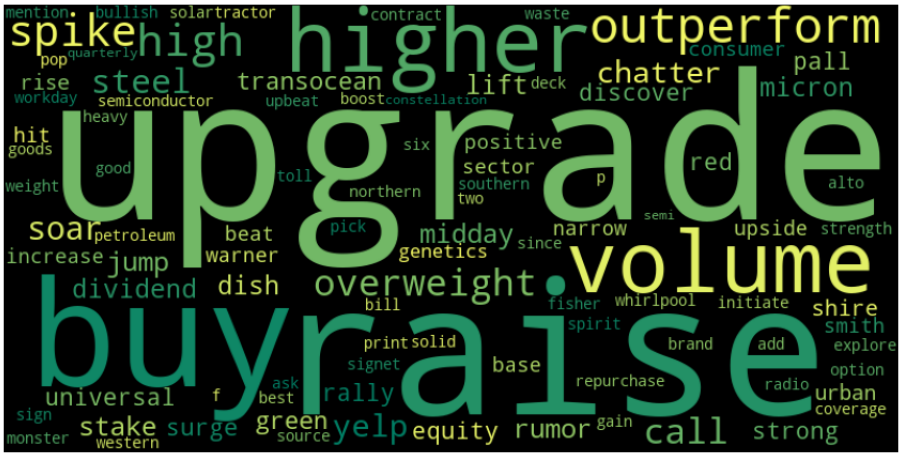
from 9 a.m. on day $t-1$ to 9 a.m. on day t are selected and ranked p value calculated from the scoring step (2.2.3). Each ticker in the sample is then ranked according to sentiment of related headlines for that day. If a ticker has multiple headlines, the average sentiment from all related headlines is taken for the firm. From this, a portfolio is created, where the top 50 sentiment stocks are bought, and the lowest 50 sentiment stocks are shorted.

Some constraints are placed on the stocks that can be chosen, to ensure that stocks are not bought when they have negative sentiment. For a stock to be bought, it must have $\widehat{p}_i < 0.5$, and the inverse for a stock to be sold. On the occasion where there are not 50 stocks with positive sentiment, this is to avoid the portfolio purchasing slightly negative stocks, and therefore less than 50 stocks are used in this instance.

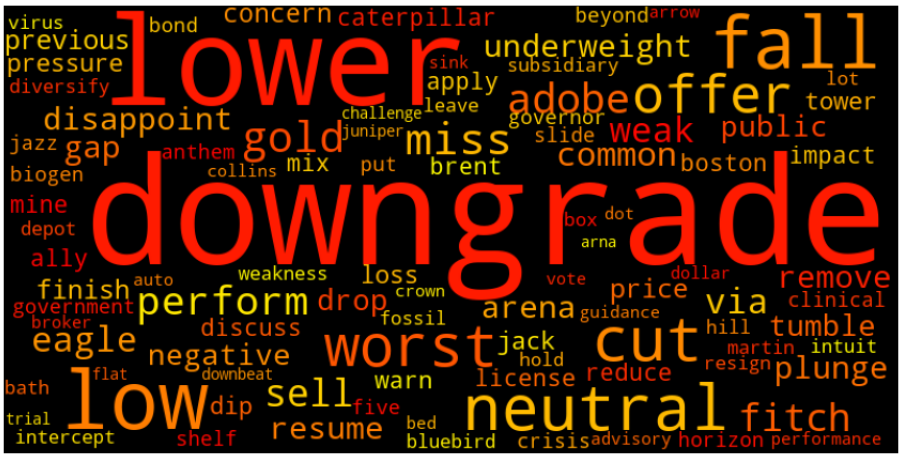
Chapter 4

Critical Evaluation

4.1 Analysis of Word Lists



(a) Positive words



(b) Negative words

Figure 4.1: Word clouds demonstrating sentiment charged words. Font size corresponds to average tone across all training samples

Following the construction of matrix O, figure 4.1 demonstrates the list of sentiment charged words on average over all training 19 windows. At each training and validation window, the sentiment lists are generated completely from scratch, and while there is some overlap, each list can vary significantly. The font size corresponds to the average tone (calculated by $\frac{1}{2}(O_+ - O_-)$) of the words across all windows. Of the top 50 positive sentiment words, the following appeared in at least 75% of windows, with words

highlighted in **bold** appearing in all windows:

rumor, outperform, repurchase, spike, volume, **raise, high, upgrade**

The following words appear in at least 85% of windows with respect to top 50 negative sentiment words with words highlighted in **bold** appearing in all windows:

plunge, remove, lower, public, fall, **miss, lower, downgrade, cut, underweight**

Simply by inspection, each group appears reasonable, in the sense that many of the words with high values in either sentiment could be assumed. However, some words are somewhat surprising and this may offer an insight into subconsious bias that exists in writing headlines as opposed to article bodies. For example, the word *volume* is, under normal circumstances, a sentiment neutral word, but according to the model generated by SESTM, is a highly positive word. Examples of headlines including this word include:

- *Agilent spikes to high of $60.40 on Volume*
- *Markets gather some momentum as volume remains light, geopolitical tension improving*
- *Tuesday's Mid-day options for volume leaders*

Observing these headlines, it is clear that the words are being used in a positive context, and this could be due to subconscious usage of the word when constructing such headlines. However, another explanation could be overfitting. Included in the sample are headlines from 'Benzinga', which is a company that offers realtime news articles, and has a significant quantity of headlines of the form *Benzinga's top upgrades* (around 16,000 headlines from the entire dataset) and *Benzinga's volume* (with around 2,000). This could be seen as an issue of overfitting and may skew these words' sentiment value meaning that it is not reflective of the true sentiment of the word when not used in the context of Benzinga. However, both 'upgrade' and 'volume' appear multiple times in the word lists for bigrams [1] with different combinations of words, meaning that there is positive sentiment attached to these words without the context of Benzinga.

When compared to the Harvard IV and Loughran McDonald dictionaries, we find that the majority of words labelled with sentiment according to our model are not in either dictionary. The negative sentiment words have much higher overlap, with 13 of the top 50 words appearing in the LM dictionary, while only 3 appear in the H4. Conversely, only 6 words overlap LM in the positive tone, while 5 words overlap the H4 dictionary. Furthermore, many words that are included in either dictionary are determined to be sentiment neutral by the model. This is because the model is trained on a sample of headlines and the vocabulary used in headlines is vastly different to that in everyday use or 10-k filings in the case of LM. Headline vocabulary often contains much more impactful words, as it is intended to be a punchy, attention grabbing piece of text. Often, words that are typically used in headlines are rarely found outside of the context of headlines [Reah, 2002]. For this reason, the lexicons of the model, and that of H4 and LM differ.

4.2 Daily returns

	Sharpe	Average	Daily	FF3		FF5	
Formation	Ratio	Return	Turnover	α	R^2	α	R^2
EW L-S	0.77	7.54	90.3%	5.18	2.83%	5.67	3.50%
EW L	1.05	12.59	91.83%	10.47	37.5%	10.37	38.50%
EW S	-0.43	-5.05	88.70%	-5.29	21.08%	-4.70	21.34%
VW L-S	0.26	2.65	90.09%	0.93	2.32%	1.59	3.75%
VW L	0.35	4.42	90.38%	2.65	25.75%	2.72	27.65%
VW S	-0.21	-1.77	89.80%	-1.72	24.20%	-1.13	24.75%

Table 4.1: Performance of Daily News Sentiment Portfolios

[1] Information on bigram computation further on

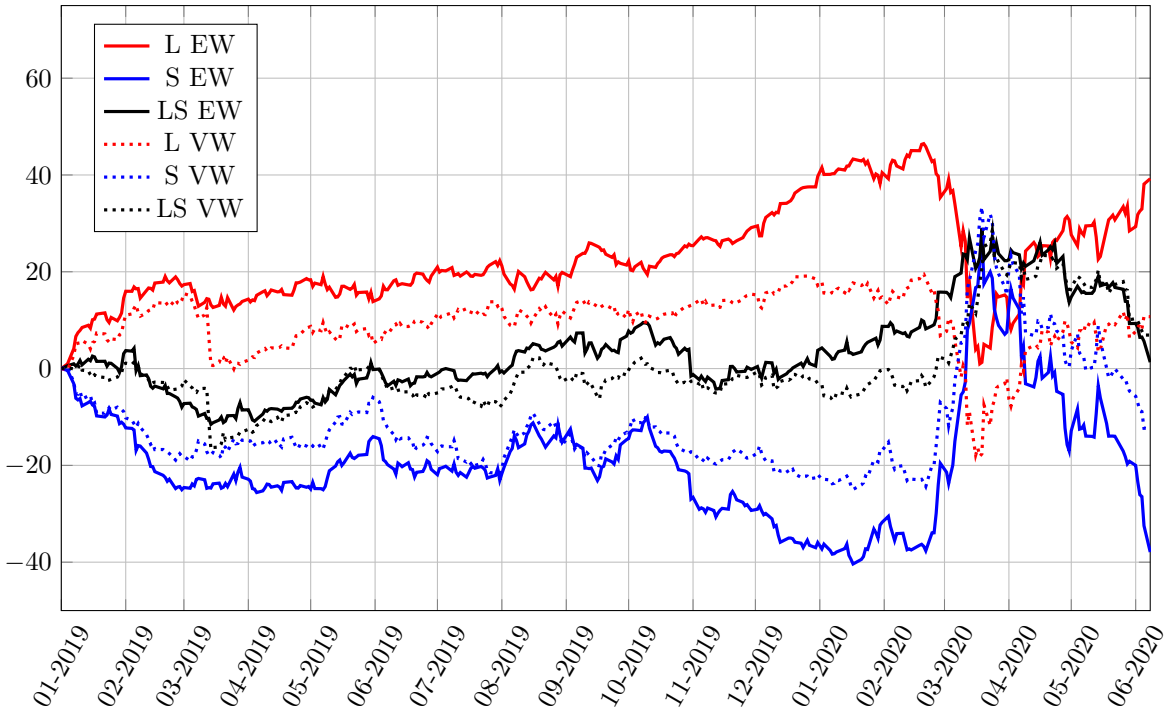


Figure 4.2: Cumulative log returns for each formation over the out of sample headlines

Using the headlines that were saved for out of sample testing, a portfolio is constructed for each day. On average, 353 firms have articles linked to them on a given day, and of these, almost half of these headlines contain one or more sentiment words (are not marked neutral by the model). According to the constraints ($\widehat{p}_i < 0.5$ for a stock to be bought, and vice versa for a stock to be sold), there are some days where less than 100 stocks form the portfolio, in which case we trade with the largest value possible. On average, the long side of the portfolio has 40 stocks, while the short side has 48, therefore the average number of stocks in the portfolio is 88.

Table 4.1 describes the performance of the constructed portfolios. The two investment methods (equal and value weighted) are split up into the Long-Short combined portfolio (L-S), and the long (L) and short (S) legs are also displayed separately for comparison purposes. The daily turnover section displays the average turnover each day, which would be 100% as the profit is liquidated at the end of each day, but some stocks are retained the following day. A turnover of 90% (as in VW L) implies that on average 1 in 10 stocks are retained the following day. This could be due to headlines or news articles that are concerning the same events (stale news), or repeat sentiment headlines as a story unfolds over a number of days.

Unfortunately, neither of these formations are very profitable, with the only formation that is profitable being the equal weighted long strategy with a Sharpe ratio of 1.07, indicating that the profit versus risk ratio is beneficial.

The FF3 and FF5 sections refer to Fama French 3 and 5 factor regression respectively, while the α concerns the intercept. The higher percentage of the average returns that the α value is refers to the amount of private information held in the investment. In other words, if the α is a very small percentage of the generated returns, then the returns that an investment has generated can be explained by regular movement in the markets, and there is no private information that is being used to generate profit.

4.2.1 Speed of information Assimilation

4.2.2 Comparison to other methods

4.3 What to do

A topic-specific chapter

	Sharpe	Average	Daily	FF3		FF5	
Formation	Ratio	Return	Turnover	α	R^2	α	R^2
			Day $t-1$				
EW	18.80	267.01	90.3%	5.18	2.83%	5.67	3.50%
VW	13.32	168.69	90.09%	0.93	2.32%	1.59	3.75%
			Day $t+0$				
EW	13.24	152.74	90.3%	5.18	2.83%	5.67	3.50%
VW	10.70	101.40	90.09%	0.93	2.32%	1.59	3.75%
			Day $t+1$				
EW	0.77	7.54	90.3%	5.18	2.83%	5.67	3.50%
VW	0.26	2.65	90.09%	0.93	2.32%	1.59	3.75%

Table 4.2: Performance of Daily News Sentiment Portfolios day $t-1$ to day $t+1$

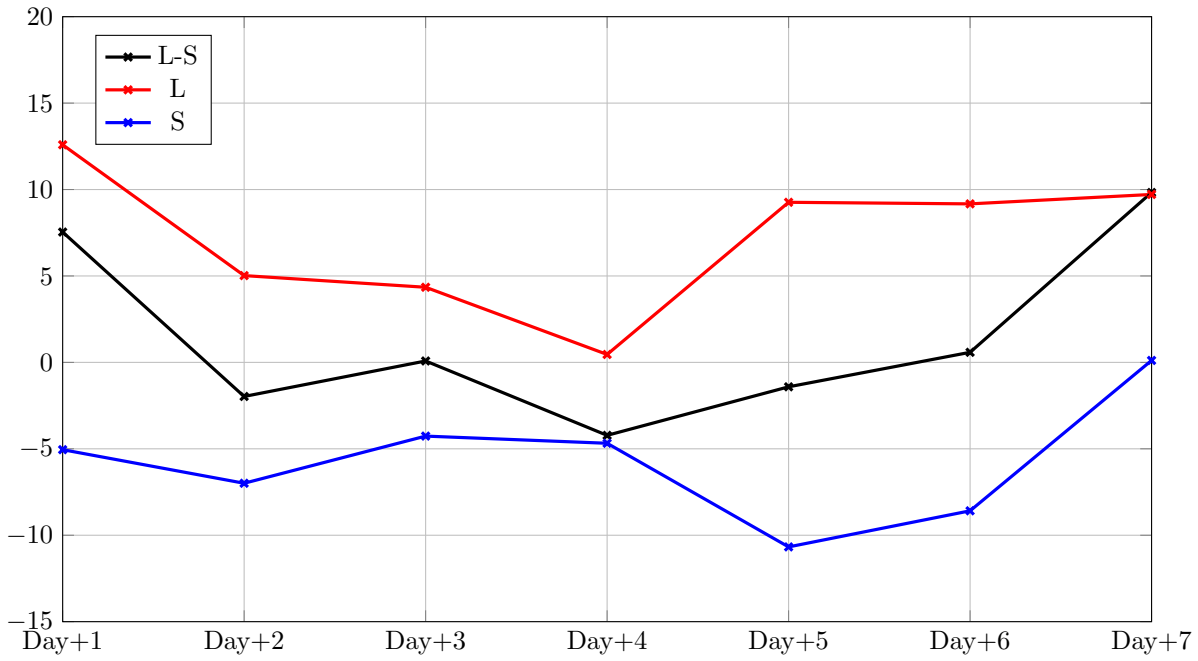


Figure 4.3: Cumulative log returns for each formation over the out of sample headlines when headlines are considered at different delays. These formations are impossible in practice, but show the successful sentiment capture of the model

This chapter is intended to evaluate what you did. The content is highly topic-specific, but for many projects will have flavours of the following:

1. functional testing, including analysis and explanation of failure cases,
2. behavioural testing, often including analysis of any results that draw some form of conclusion wrt. the aims and objectives, and
3. evaluation of options and decisions within the project, and/or a comparison with alternatives.

This chapter often acts to differentiate project quality: even if the work completed is of a high technical quality, critical yet objective evaluation and comparison of the outcomes is crucial. In essence, the reader wants to learn something, so the worst examples amount to simple statements of fact (e.g., "graph X shows the result is Y"); the best examples are analytical and exploratory (e.g., "graph X shows the result is Y, which means Z; this contradicts [1], which may be because I use a different assumption"). As such, both positive *and* negative outcomes are valid *if* presented in a suitable manner.

Chapter 5

Conclusion

The concluding chapter of a dissertation is often underutilised because it is too often left too close to the deadline: it is important to allocate enough attention to it. Ideally, the chapter will consist of three parts:

1. (Re)summarise the main contributions and achievements, in essence summing up the content.
2. Clearly state the current project status (e.g., "X is working, Y is not") and evaluate what has been achieved with respect to the initial aims and objectives (e.g., "I completed aim X outlined previously, the evidence for this is within Chapter Y"). There is no problem including aims which were not completed, but it is important to evaluate and/or justify why this is the case.
3. Outline any open problems or future plans. Rather than treat this only as an exercise in what you *could* have done given more time, try to focus on any unexplored options or interesting outcomes (e.g., "my experiment for X gave counter-intuitive results, this could be because Y and would form an interesting area for further study" or "users found feature Z of my software difficult to use, which is obvious in hindsight but not during at design stage; to resolve this, I could clearly apply the technique of Smith [7]").

Bibliography

[Fama and French, 1992] Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, 47(2):427–465.

[Fama and French, 2015] Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of financial economics*, 116(1):1–22.

[French, 2022] French, K. (2022). Kenneth r. french - data library.

[Hofmann, 2013] Hofmann, T. (2013). Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*.

[Ke et al., 2019] Ke, Z. T., Kelly, B. T., and Xiu, D. (2019). Predicting returns with text data. Technical report, National Bureau of Economic Research.

[LOUGHRAN and MCDONALD, 2011] LOUGHRAN, T. and MCDONALD, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65.

[Reah, 2002] Reah, D. (2002). *The language of newspapers*. Psychology Press.

[Sharpe, 1966] Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business*, 39(1):119–138.

Appendix A

An Example Appendix

Content which is not central to, but may enhance the dissertation can be included in one or more appendices; examples include, but are not limited to

- lengthy mathematical proofs, numerical or graphical results which are summarised in the main body,
- sample or example calculations, and
- results of user studies or questionnaires.

Note that in line with most research conferences, the marking panel is not obliged to read such appendices. The point of including them is to serve as an additional reference if and only if the marker needs it in order to check something in the main text. For example, the marker might check a program listing in an appendix if they think the description in the main dissertation is ambiguous.

Positive					Negative				
Word	Sentiment	Count	LM	H4	Word	Sentiment	Count	LM	H4
upgrade	0.020817	20	0	1	downgrade	-0.028087	20	1	0
raise	0.017047	20	0	0	lower	-0.018566	20	0	0
high	0.002975	20	0	0	cut	-0.00346	20	1	0
volume	0.006617	18	0	0	miss	-0.001796	20	1	0
outperform	0.003602	16	1	0	underweight	-0.000875	20	0	0
spike	0.002856	16	0	0	fall	-0.004667	19	0	0
repurchase	0.000383	16	0	0	weak	-0.001026	19	1	0
rumor	0.000896	15	0	0	low	-0.006097	18	0	0
buy	0.010115	14	0	0	public	-0.000672	18	0	0
higher	0.006746	14	0	0	plunge	-0.000851	17	0	0
overweight	0.001977	14	0	0	remove	-0.000822	17	0	0
green	0.000745	14	0	0	offer	-0.002654	16	0	0
lift	0.000786	13	0	0	common	-0.00099	16	0	0
solid	0.000382	13	0	0	disappoint	-0.000845	16	1	0
soar	0.001384	12	0	0	negative	-0.000697	16	1	1
strength	0.000265	12	1	0	neutral	-0.003533	15	0	0
mention	0.000183	12	0	0	concern	-0.000515	15	1	0
special	0.000131	12	0	1	impact	-0.000425	15	0	0
chatter	0.001139	11	0	0	shelf	-0.00031	15	0	0
strong	0.000701	11	1	0	weakness	-0.000259	15	1	0
quarterly	0.000179	11	0	0	worst	-0.003078	14	1	1
dynamics	0.00013	11	0	0	pressure	-0.000469	14	0	0
jump	0.000898	10	0	0	resume	-0.000858	13	0	0
upside	0.000418	10	0	1	tumble	-0.000674	13	0	0
boost	0.000333	10	1	0	dip	-0.000481	13	0	0
steel	0.001527	9	0	0	perform	-0.001468	12	0	0
stake	0.000901	9	0	0	sell	-0.001288	12	0	0
micron	0.000816	9	0	0	loss	-0.00048	12	1	1
rally	0.000676	9	0	1	adobe	-0.001511	11	0	0
narrow	0.000459	9	0	0	drop	-0.00069	11	0	0
f	0.000373	9	0	0	prelim	-0.000171	11	0	0
weigh	9e-06	9	0	0	fitch	-0.001301	10	0	0
call	0.001808	8	0	0	beyond	-0.000341	10	0	0
yelp	0.001253	8	0	0	challenge	-0.000199	10	1	0
dish	0.000812	8	0	0	reduce	-0.000396	9	0	0
dividend	0.000809	8	0	0	warn	-0.000392	9	1	0
surge	0.000692	8	0	1	secondary	-0.000175	9	0	0
base	0.000419	8	0	0	downside	-0.000169	9	0	0
pop	0.000353	8	0	0	halt	-0.000165	9	1	0
monster	0.000286	8	0	0	propose	-9.7e-05	9	0	0
expansion	0.000162	8	0	0	community	-3.9e-05	9	0	0
attribute	0.000142	8	0	0	four	-3.3e-05	9	0	0
rebound	0.000138	8	1	0	gold	-0.001621	8	0	0
unconfirmed	0.000114	8	0	0	finish	-0.000573	8	0	0
announcement	0.000102	8	0	0	price	-0.000542	8	0	0
declare	6.5e-05	8	0	0	mix	-0.000368	8	0	0
test	3.2e-05	8	0	0	license	-0.00036	8	0	0
improve	3e-05	8	1	0	slide	-0.000283	8	0	0
urban	0.000528	7	0	0	leave	-0.00023	8	0	0
beat	0.000459	7	0	0	bed	-0.000217	8	0	0

Table A.1: Top sentiment words for each polarity, along with appearance in either Loughran McDonald dictionary (LM) or Harvard IV psychological dictionary (H4). Note sentiment in this case refers to average *tone* over all 20 training windows. Words are first sorted via count of training windows appeared in, and then by sentiment