U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Brett Drury

A Text Mining System for Evaluating the Stock Market's Response To News.

Doctoral Program in Computer Science
of the Universities of Minho, Aveiro and Porto

Universidade do Minho

April 10, 2013

Brett Drury

A Text Mining System for Evaluating the Stock Market's Response To News.

Thesis submitted to Faculty of Sciences of the University of Porto
for the Doctor Degree in Computer Science within the Joint Doctoral Program in
Computer Science of the Universities of Minho, Aveiro and Porto

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
April 10, 2013

I would like to dedicate my thesis to: my wife, Helena Morais, my son Samuel, and my parents, Michael and Sue Drury.

Acknowledgements

This Ph.D. was undertaken in the Department of Computer Science at the University of Porto under the supervision of Luis Torgo and José João Dias de Almeida. I would like, in particular, to thank Luis Torgo for his patience and his continual support for the duration of this Ph.D., especially in the writing up phase.

I would like to thank the support staff in the Department of Computer Science, especially Hugo Ribeiro. I would also like to thank Justino Soares and Carlos Soares who managed to persuade the management at my previous employers to allow me time to work on my Ph.D., without his assistance I would have not been able to continue on the doctoral program.

I would like to thank my co-authors, Gael Dias, José João Dias de Almeida and Helena Morais for their contribution to my work.

Lastly, I would like to thank the organizers and lecturers involved in the MAPi doctoral program.

Preface

This thesis was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science awarded by the MAPi Joint Doctoral Programme. The work presented in this thesis is the culmination of 4 years of work under the supervision of Prof. Luis Torgo from the University of Porto and Jose João Dias de Almeida from the University of Minho.

Abstract

This thesis presents a text mining system which was designed to predict the direction of a share or financial market. The text mining system is a complete pipeline which: 1. scrapes new stories from the Internet, 2. extracts news text from the scraped news stories, 3. identifies relevant news stories for a specific company or financial market, 4. classifies sentences, news stories and quotes and 5. makes a trading inference using these classifications.

The thesis documents advances in 1. ontology construction and maintenance, 2. fine grained event and sentiment extraction and classification at the sentence level, 3. news story classification, 4. direct speech classification and 5. information retrieval. These advances were also contributions to the fields of semi-supervised learning and ontology engineering.

The advances in the news classification at the document, sentence and direct speech level demonstrate measurable advantages in trading experiments on the FTSE 250 financial markets over competing text classification strategies. The complete system, however, did not demonstrate a measurable trading advantage in experiments conducted on the shares of Apple, Google, IBM and Microsoft. The system, however, provides a blueprint for future systems.

Resumo

Nesta tese é apresentado um sistema de" text mining" que foi concebido para prever a direcção de movimento de uma acção ou mercado financeiro. O sistema de "text mining" é uma "pipeline" completa que: 1. extrai noticias da internet; 2. extrai o texto das noticias, 3. identifica noticias relevantes para uma empresa especifica ou mercado financeiro, 4. classifica frases, noticias e citações, 5. faz uma previsão de negociações usando essas classificações.

A tese descreve avanços em 1. construção e manutenção de ontologias, 2. extracção precisa de eventos e sentimento, e classificação ao nível da frase, 4. caracterização de discurso directo, 5. obtenção de informação. Estes avanços são também contribuições nos campos de "semi-supervised learning" e engenharia de ontologias.

Os avanços na classificação de noticias ao nível do documento, frase e discurso directo, demonstram vantagens consideráveis em experiências de negociação no FTSE 250, face a outras estratégias. O sistema completo não demonstra, no entanto, uma vantagem significativa em experiências de negociação de acções da Apple, Google, IBM e Microsoft. O sistema fornece, no entanto, um "blueprint" para sistemas futuros.

Contents

List of Tables

List of Figures

Chapter 1

Overview

1.1 Introduction

This thesis is intended to be a contribution to the increasingly popular area of research of making deductions and inferences about a real world problem from information described in text. The chosen area for this thesis was the estimation of the direction of share price movements. This area was chosen because there are: 1. existing hypothesis that suggests news can influence the stock market and 2. large volumes of freely available news / financial data. Information contained in news which can aid the estimation of a share's price direction is known as "alpha". This thesis is, in essence, an exercise in identifying alpha in news stories.

1.1.1 Problem Definition

A high level problem definition for this thesis can be summarized in a single phrase: the conversion of text into information [Mitra and Mitra, 2011b]. A more concrete definition may be: *Business / financial news stories can contain information (words, sequence of words or overall story opinion) which has an economic value which is not reflected in the current price of a specific share or market index.* Business/ financial news, therefore can be used as a basis of a successful trading strategy. In summary, the underlying hypothesis for the work of this thesis is that it is possible to improve an existing trading strategy by adding information from a financial news text mining system.

There are a number of challenges which are presented by the processing of news. The challenges may be broken down into: 1. lexical, 2. identification, 3. behavioural and 4. data acquisition. The challenges described in this section are specially related to finance or business news.

Lexical challenges are associated with the language and grammar contained in news stories.

Financial / business news contains its own unique lexicon with words which are unique to it, for example: "a bear market" or "double dip recession". This unique lexicon can produce unconventional grammars. An example of atypical grammar can be demonstrated by a negation process. Negation inverts the sentiment of a word / phrase, e.g. "good" (positive sentiment) is inverted by the word "not", consequently, "not good" has the opposite sentiment to "good". In the following quote the *positive* expression "V-shaped recovery" is negated by the expression *far cry*.

> But because these gains follow the massive contraction that occurred during the Great Recession of 2008-2009, they are a *far cry* from the trajectory of a classic *V-shaped recovery*.
> Global economic recovery more superficial than real - Gulf News July 5th 2011

This example demonstrates only one unique feature of news language and it is likely that there other atypical grammars which are common in news stories. This may inhibit strategies which rely upon general linguistic resources, for example, sentiment dictionaries, to extract information from news stories because these resources may not contain specific terms from the finance domain.

Identification Challenges refers to the identification of news stories because a strategy which can accurately classify or extract information from news stories will be reliant upon the discovery of relevant stories. Relevant stories can often be stories which contain a direct reference to a monitored company, but relevant news may be news concerning a competitor or general macro-economic news. The phenomenon of indirectly related news effecting a given stock or share is often referred to as "spill-over" and will be discussed at length later on in this thesis.

Behavioural Challenges refers to the trading behaviour of stock traders. A successful classification of a relevant news story may not be sufficient to trigger a successful trade if the news is expected. If the news is expected then the price of a share in advance of the news will reflect this expectation and will not move when the news is published.

Data Acquisition Challenges refers to the acquisition of raw data and training data. Freely available news is published on the Internet on news sites. News sites don't share the same "templates" and often have advertisements and non-news content on the same page as the news content. This "non-news" content may inhibit news story analysis if its content is included in the analysis process. In addition there is no freely available annotated news stories which can be used as training data.

The themes for this thesis can be simply summarized as: *what news stories to analyse* and *how to analyse the news stories*. The approach proposed by this thesis varies from the published approaches because information retrieval strategies are given as much prominence as the

classification strategies. The classification strategies vary from the typical approaches by combining rules, market alignment and semi-supervised learning to establish robust models. An additional variation from the research literature are the experiments with "direct speech" from economic actors as a basis of a trading strategy. Finally, the thesis is a complete "pipeline" which: 1. gathers news stories from the web, 2. extracts news information from the original HTML, 3. selects the news stories to be classified, 4. classifies elements from the news stories and 5. adds classified elements as features to an existing stock trading system.

1.2 Literature Review

This section describes the literature review conducted to identify relevant stock prediction investigation from the news research community. The literature review was in two parts: economic literature and existing stock prediction systems. The economic literature review was to ensure: 1. markets react to news and 2. a profitable trading strategy can be achieved from trading on news information.

1.2.1 Economic Literature

The earliest evidence in the research literature of the successful use of news information in a real world trading strategy was by Victor Niederhoffer in the early 1970s. Niederhoffer's employees classified stories in the print media into 19 categories, which were a sliding scale of polarity from optimistic to indifferent to pessimistic. Niederhoffer claimed that a polarity of a news story was a good indication of "alpha" and consequently he was able to conduct a successful trading strategy [Niederhoffer, 1971]. De Bondt and Thaler [1985] provided more detail that markets can react to events. Their central claim was: *markets over-react or under-react to an event and then correct.* They made further claims: 1. *Extreme movements in stock prices will be followed by subsequent price movements in the opposite direction* and 2. *The more extreme the initial price movement the greater will be the subsequent price movement.* De Bondt and Thaler claimed that events can move markets, and because they overreact or underreact it may be possible to implement a successful trading strategy. An illustrative example of this effect was the false United Airlines bankruptcy story in September 2008[1]. The share price plunged on the bankruptcy news and then recovered when the story was corrected. The share price is illustrated in Figure 1.1. The graph demonstrates the opening price, the lowest price and the closing price. A trading strategy which used the De Bondt and Thaler hypothesis would have bought at the lowest price because it would have anticipated a later market correction. The false news story also had a depressing effect on other major airlines [Carvalhob, Klaggea, and Moencha, 2011]. This effect is known as "spill-over" and

[1] Details can be found at `http://news.bbc.co.uk/2/hi/7605885.stm`

will be discussed later on in this section. Veronesi [1999] suggested that the overreaction and underreaction to news is a form a hedging by market actors.

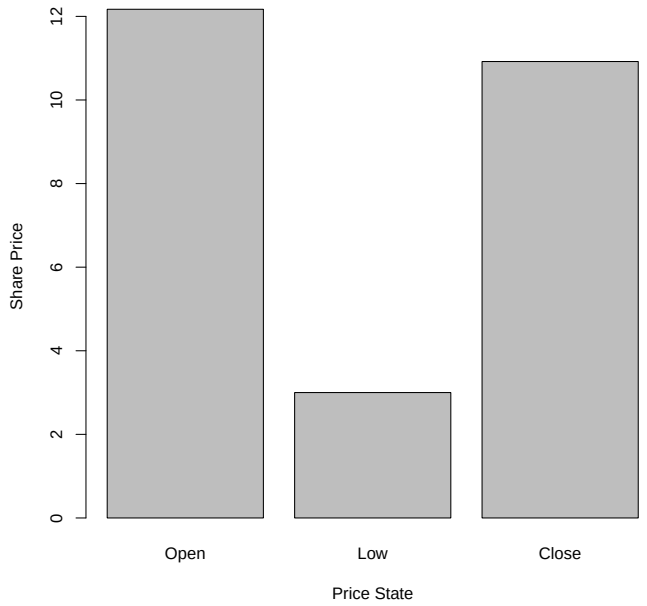

Figure 1.1: United Airlines share price September 8th 2008.

There are further examples where share price has reacted to events, for example the Google+ launch [Schonfeld, 2011] and the free Nokia GPS launch [Hamblen, 2010]. The impact of events upon share prices is documented in Table 1.1

A possible flaw with the De Bondt and Thaler hypothesis is that not all events provoke a reaction [Schuster, 2003][Culter, Poterba, and Summers, 1991][Robbani and Anantharaman, 2004]. Schuster [2003] provides a review of "big events" and the reaction of the S&P market index. The results are described in Figure 1.2. Events which have an economic consequence provoke the expected reaction [De Bondt and Thaler, 1985].

Hypothesis 1: *Events can provoke an immediate reaction in the stock market, but the event must have economic consequences.*

The effect of a relevant event on a given instrument can be amplified if it is unexpected Schuster [2003][Bomfim, 2000]. Expected events tend to provoke no reaction [Schuster, 2003]. The identification of unexpected events in news may be assisted by the economics of news

Company	Date	Event	Open	Low	Close
United Airlines	08-09-2008	Bankruptcy Rumour	12.17	3.00	10.92
Tom Tom	28-10-2009	Launch of Free GPS System by Google	10.60	8.06	8.11
Garmin Ltd	28-10-2009	Launch of Free GPS System by Google	36.02	30.85	31.59
Tom Tom	21-01-2010	Launch of Free GPS System by Nokia	6.50	5.43	5.90
Garmin Ltd	21-01-2010	Launch of Free GPS System by Nokia	34.54	33.95	34.16
Google	27-06-2011	Launch of Google+	474.00	473.50	482.80

Table 1.1: Impact of events on share prices.

Event	S & P Reaction
Bay of Pigs invasion	-0.47
Soviet invasion of Afghanistan	+0.11
Assassination attempt on Ronald Reagan	+0.27
The death of U.S. marines in Lebanon	+0.02
Chernobyl Nuclear Disaster	-1.06
Hurricane Andrew	-0.8

Table 1.2: Market reaction to big events.

publishing. Financial news published in the mass media must have sufficient "utility" to attract readers [McManus, 1988], consequently financial news published in the mass-media are likely to be unexpected or sufficiently interesting to be published. The publication of events in the mass media are likely to intensify market reaction [Schuster, 2003].

Hypothesis 2: *The mass-media filters expected or inconsequential news stories.*

Sentiment in news can be an indication of the future prospects of an economic actor [Davis and Piger, 2006]. This information can be extracted by sentiment analysis systems which seek to extract emotions and feelings expressed about people, organisations, nation states, goods and services, in free natural language texts [Glucksberg, 2008]. Davis hypothesized the effect of language in news and its effect on economic actors. She stated: "We find a

positive (negative) association between optimistic (pessimistic) language usage and future firm performance and a significant incremental market response to optimistic and pessimistic language usage in earnings press releases. Results suggest managers use optimistic and pessimistic language to provide credible information about expected future firm performance to the market, and that the market responds to managers' language usage." [Davis and Piger, 2006]. Davis view is supported by Tetlock [Tetlock, Saar-Tsechansky, and Macskassy, 2008] . He provided a "check list" summary of his findings, which were:

- The fraction of negative words in firm-specific news stories forecasts low firm earnings.
- Firms' stock prices briefly underreact to the information embedded in negative words.
- The earnings and return predictability from negative words is largest for the stories that focus on fundamentals. [Tetlock, Saar-Tsechansky, and Macskassy, 2008]

Henry [2006] analysed the writing style of company reports and found that changes in writing style from optimistic to pessimistic can be a good indicator of a company's futures prospects. She stated: "Results of the study show that inclusion of predictor variables capturing verbal content and writing style of earnings-press releases results in more accurate predictions of market response.".

The view that sentiment in news can effect financial market's value is not limited to academia. Ravenpack, a company which produces a "news analytic" system released a white paper which stated that there were correlations between sentiment in news and two week returns in the Eurstoxx and DJIA (Dow Jones) market indexes [Hafez, 2009].

Hypothesis 3: *The market reaction to sentiment information can be over a longer period than market reaction to events*

News stories may effect other economic actors which are not specifically mentioned in the news story, for example: the downgrade of the credit rating of Portugal directly effected the yields required from government debt for Spain, Italy and Ireland[2]. This effect is known as "spill over" where the prospects of one economic actor effect other related economic actors.

Definition Spill-Over: A specific economic-actor event which also effects the sector index price

[2]More details can be found here http://goo.gl/vtxU5

Hafez [2010] [Mitra and Mitra, 2011c] states that there are a number of studies which demonstrate company specific news events can effect the industry index price. Therefore, an event which notionally concerns a single company can have an effect on many companies in the same sector. This effect is reflected in the industry index price. He does not elaborate on these sources, but provides an example from company research by Cahan et al who suggested that news sentiment for a company can effect the whole industry [Hafez, 2010] [Mitra and Mitra, 2011c]. The evidence that they provided was that *sector excess returns following company specific news events are smaller than market excess returns indicating that the sector also moves on the event* [Hafez, 2010] [Mitra and Mitra, 2011c].

Hafez suggests that there are "spill-over" effects from news releases because they have impact on risk and covariance of stocks [Hafez, 2010] [Mitra and Mitra, 2011c]. In addition he states there is further evidence of information in macroeconomic news effecting company share prices [Panetta, 2002].

Hypothesis 4: *Company share prices can move on news where it is not explicitly mentioned in the news text*

This section provides the economic foundations and justifications for the approach taken in this thesis. This section is not a comprehensive review of the economic literature because this thesis is in the domain of Computer Science rather than Economics. This section, however presents a summary of the current trends of evaluating the effect of news on stock prices from the economics perspective.

1.2.2 Existing Systems

This section describes systems which have been published in peer reviewed journals. There is a caveat. The construction of systems which predict the stock markets has an economic value, and consequently there may have been advances which have not been published. A further caveat is that the published work often omits details which can inhibit an implementation of the published system.

Dictionary Approaches

A dictionary based system typically relies upon a pre-compiled dictionary. The dictionary normally contains a word n-gram ($n \geq 1$) and a value. The function of the value is to infer an influence of the word, for example the unigram "bankrupt" may may have a negative

value. The research literature contains a number of systems which claim to have achieved positive trading returns.

A typical example of a dictionary approach was described by Wuthrich, Cho, and Leung [1998]. Wuthrich's system analysed "overnight" news which is news published when the financial markets were closed. The dictionary constructed by Wuthrich contained tuples of words separated with an "AND" Boolean instruction, for example "Bond" and "Strong". The news stories were categorized based upon the contents of the dictionary. The number of stories in each category would be counted and sell or buy instruction would be generated for the index. They claim a 21% advantage over and above a trader who would trade by guessing based on an uniform distribution [Wuthrich, Cho, and Leung, 1998].

The News Categorization and Trading System *(NewsCATS)* [Mittermayer and Knolmayer, 2006] also relied upon a pre-compiled dictionary. NewsCATS analysed company press releases, and attempted to predict the company's share price upon information contained in the press release. The dictionary was not published, but the authors state that the dictionary was created by hand. The construction methodology was also not published. The function of the dictionary was to assist NewsCATS to categorise press releases into pre-designated categories. These categories were designed to indicate the influence (positive or negative) of the press release upon the share price. The system's authors claim that they significantly out perform a trader who buys on a random basis after press releases are released [Mittermayer and Knolmayer, 2006].

Another system which relied upon a pre-compiled dictionary was developed by Peramunetilleke and Wong [2002]. Their system attempted to predict the movements of targeted currencies. The pre-compiled dictionary was constructed by aligning news stories with movements in the targeted currency. The methodology assumed that news stories which co-occurred with a gain in a certain currency were positive whereas news stories which co-occurred with a drop in a certain currency were negative. The system analysed headlines because they have *a restricted grammar and provide an accurate summary of the following story*. The dictionary contained word unigrams. The dictionary construction methodology extracted unigrams from the headlines of the aligned stories. The words were stemmed and assigned a weight. The unigrams were assigned a polarity (+/-) by the aforementioned alignment strategy. The system classified incoming news headlines with terms in the dictionary. The system used the classified headlines to make a prediction about the direction of a currency. The authors claim a near human performance of being correct nearly 50% of the time [Peramunetilleke and Wong, 2002].

Machine Learning

An alternative to hand-crafted or automatically generated dictionaries is the application of a form of machine learning to classify news stories into predesignated categories.

The *AEnalyst* system was a system designed by Lavrenko, Schmill, Lawrie, Ogilvie, Jensen, and Allan [2000]. The AEnalyst constructed Language Models by aligning news stories with trends in the market. Stories which co-occurred with a negative trend were assumed to be negative whereas stories which co-occurred with a positive trend were assumed to be positive and stories which co-occurred with neither a negative or positive trend were assumed to be neutral. The language models represented probability distributions of unigrams for each category. The AEnalyst system used an activity monitoring strategy to issue alerts to warn of positive or negative trends in streaming news data [Lavrenko, Schmill, Lawrie, Ogilvie, Jensen, and Allan, 2000]. The alerts were issued when the unigrams in the incoming news stories passed a given threshold.

Gidófalvi [2001] used the aforementioned alignment strategy to train a Naive Bayes classifier. The three categories the Naive Bayes classifier was trained for were: 1.up, 2. down or 3. unchanged. Gidófalvi claimed positive returns, but stated the predictive nature of the news stories was limited to 20 minutes from the news stories being published [Gidófalvi, 2001].

Fung, Yu, and Lam [2002] also used a time-series alignment strategy, but used a technique they coined as "guided clustering". The trends detected in time-series data were clustered and news articles were aligned to the clusters. The authors do not state explicitly how the alignment was achieved, but they stated that the stories *would support and account for the happening of the trends.* This quote would suggest that alignment was conducted manually by a domain expert. The features selected from the clusters are then used as features for a Support Vector Machine (SVM). The SVM detected trends which allowed the use of a trading strategy which the authors claim was profitable. [Fung, Yu, and Lam, 2002]

Izumi, Goto, and Matsui [2010] proposed a strategy for predicting bond prices. They extracted feature vectors from the Bank of Japan reports. Trends in the bond market were estimated by regression analysis using the aforementioned feature vectors. The authors claim that they could predict long term bond market trends using this methodology.

AZFinText system [Schumaker and Chen, 2009] used a machine learning approach to predict a future price of a given stock. They tested four news story representation strategies: 1. Bag of Words, 2. Noun Phrases, 3. Proper Nouns and 4. Named Entities. They found that Proper Noun representation provided the best returns.

Other Systems

A hybrid approach was proposed by Luss [2009]. He created a dictionary which contained negative and positive unigrams. The unigrams were used as features for a Support Vector Machine (SVM). The SVM classified press releases (PRNewswire) and made a prediction of the direction of the market. The author claimed that the approach produced abnormal returns [Luss, 2009].

Another hybrid approach was proposed by Tang, Yang, and Zhou [2009]. They used known words to select news stories for training a learner. The approach separated the stories into known categories, and eliminated stories which occurred in more than one category. They extract features from the stories and eliminate words which were "meaningless". The features were ranked and weighted. These features were used in supervised training experiments where the predictive nature of news was compared with a moving average prediction strategy. Thomas proposed a method of detecting trends in prices of stocks and their correlation to postings on financial message boards. His proposed approach used a genetic algorithm to generate predictive models from: 1. patterns and volume of words in messages on finance boards and 2. the volume of messages on finance boards. Thomas claimed that the subsequent models allowed the prediction of the direction of the value of stock market indexes. [Thomas and Sycara, 2000] Another approach was proposed by Lu, Wei, and Chang [2009] who used features from news stories to assist a logistic regression process to detect companies in financial distress.

There have a been a number of papers which have traded on information from social media sites such as Twitter. Activity on social media sites can be can correlated with events on the stock market, and this correlation can be used to successfully trade on the stock market [Ruiz, Hristidis, Castillo, Gionis, and Jaimes, 2012]. The sentiment of posts on Twitter can assist with stock market prediction [Bollen and Mao, 2011]. The media may be new, but the methods for extracting "actionable information" rely upon event or sentiment detection methods.

1.3 Main Contributions

The main contributions of this thesis were achieved to resolve the problems set out in the *problem definition* subsection on page 10. In summary, the main problems were: 1. lexical 2, identification, 3. behavioural and 4. data acquisition.

The claims are stated below with evidence. The evidence is either a: refereed conference article or refereed journal article.

Claim 1:
The domain of a given economic actor can be represented in detail from information contained in news stories and linked data.

Evidence:
1. Drury, B. and Almeida, J.J. Construction of a Local Domain Ontology from News Stories. EPIA, Springer LNCS, pages 400-410, 2009.

2.Drury, B., Almeida, J.J and Morais M.H.M, An Error Correction Methodology for Time Dependent Ontologies.
ONTOSE Workshop, CAiSE, Springer LNBIP, pages 501-512, 2011

Claim 2:
A limited number of changes and errors in economic actor's domain can be resolved automatically.

Evidence:
1.Drury, B., Almeida, J.J and Morais M.H.M, An Error Correction Methodology for Time Dependent Ontologies.
ONTOSE Workshop, CAiSE, Springer LNBIP, pages 501-512, 2011

2. Drury, B., Almeida, J.J and Morais M.H.M, Construction and Maintenance of a Fuzzy Temporal Ontology from News Stories.
International Journal of Metadata, Semantics and Ontologies, pages "in print",2012

Claim 3:
News stories which have a direct or indirect connection to a given domain can be recommended with good precision on the basis of a domain Ontology.

Evidence:
1. Drury, B., Almeida, J.J and Morais M.H.M, Magellan: An Adaptive Ontology Driven "breaking Financial News" Recommender. CISTI, IEEE, pages 99-105, 2011.

Claim 4:
News stories can be weighted and ranked with a temporal weighting scheme which produces a clear advantage in terms of retrieval precision.

Evidence:
1. Drury, B., Almeida, J.J and Morais M.H.M, Magellan: An Adaptive Ontology Driven "breaking Financial News" Recommender. CISTI, IEEE,pages 99-105, 2011.

2. Drury, B., Almeida, J.J and Morais M.H.M, An Error Correction Methodology for Time Dependent Ontologies. ONTOSE Workshop, CAiSE, Springer LNBIP pages 501-512, 2011

Claim 5:
News stories recommended with a dynamic error corrected Ontology have a higher precision and recall than those recommend with a static Ontology or a content based system which were evaluated in this thesis.

Evidence:
1. Drury, B. and Almeida, J.J and Morais M.H.M, Magellan: An Adaptive Ontology Driven "breaking Financial News" Recommender. CISTI, IEEE, pages 99-105,2011.

Claim 6:
Financial news events and sentiment can be extracted with simple grammars.

Evidence:
1. Drury, B. and Almeida, JJ Identification of fine grained feature based event and sentiment phrases from business news stories.
WIMS, ACM, pages NA, 2011.

Claim 7:
Language models and naive bayes classifiers induced through a self-training strategy which is guided by linguistic rules have a higher F-measure and trading profit than a classifier trained on rule selected data on the domains evaluated in this thesis.

Evidence:
1. Drury, B., Torgo L and Almeida, JJ Classifying News Stories to Estimate the Direction of a Stock Market Index.
WISA Workshop, CiSTI, pages 958-962, 2011.

2. Drury, B., Torgo L. and Almeida, JJ. Classifying News Stories with a Constrained Learning Strategy to Estimate the Direction of a Market Index,
International Journal Of Computer Science and Applications, pages 1-22,2012

Claim 8:
Restraining the selection of news stories by aligning news with market movements with rules produces a classifier which has a higher F-measure and trading profit than inducing a classifier from stories selected by aligning news stories with market movements.

Evidence:
1. Drury, B., Torgo L and Almeida, JJ Classifying News Stories to Estimate the Direction of a Stock Market Index
WISA Workshop, CiSTI, IEEE, pages 958-962, 2011.

2. Drury, B., Torgo L. and Almeida, JJ. Classifying News Stories with a Constrained Learning Strategy to Estimate the Direction of a Market Index,
International Journal Of Computer Science and Applications,pages 1-22, 2012

Claim 9:
Splitting a corpus of quotations from financial news into separate groups which are determined by speaker and applying separate semi-supervised strategies to each group generates a higher F-measure than applying a single strategy.

Evidence:
1. Drury, B. Dias, G. and Torgo, L. A Contextual Classification Strategy for Polarity Analysis of Direct Quotations from Financial News. RANLP, pages 434-440, 2011.

Claim 10:
A weak classifier used in a self-training strategy achieves a higher F-measure if high confidence errors are corrected by linguistic rules than modules induced from: voting strategy, rule selected data, self-training and inductive strategy.

Evidence:
1. Drury, B., Torgo, L. and Almeida, JJ. Guided Self Training for Sentiment Classification. ROBUS Workshop, pages 9-16, RANLP, 2011.

Claim 11:
Names of business sectors and their members can be learnt from news text.

Evidence:
1. Drury, B. and Almeida, JJ. Identification, extraction and population of collective named entities from business news.
Resources and Evaluation for Entity Resolution and Entity Management Workshop ,LREC, pages na, 2010

Claim 12:
Connections between business sectors can be learnt from news text.

Evidence:
1. Drury, B. and Almeida, JJ. Identification, extraction and population of collective named entities from business news. Resources and Evaluation for Entity Resolution and Entity Management Workshop ,LREC, 2010

1.4 Organization of Thesis

This thesis is organized into : 1. Overview, 2. News Web Crawler and Text Extraction, 3. Ontology-based Information Retrieval , 4. Text Analysis, 5. System Experiments ,6. Conclusion and 7. Appendix.

Overview : The overview is a general overview of the thesis and contains the economic underpinnings and justification for the thesis. The introduction contains an overview of past systems in this field and the advances that this thesis contains.

News Web Crawler and Text Extraction : This chapter describes the crawler which fetches and extracts text information from RSS feeds. This chapter describes the automatic identification of extraction targets in news web pages. An evaluation of the text extraction scheme is provided. The chapter also describes the addition of meta-data.

Ontology-based Information Retrieval : This chapter describes two information retrieval strategies which rely upon ontologies to select news stories. The chapter describes two types of ontologies: 1. industry ontology and 2. company specific ontology.

The "industry" ontology groups related companies together under industry sectors. The industry ontology uses: 1. related companies to a target company and 2. the target company's industry sector in a entity frequency scoring scheme to select news stories. The "company" specific ontology attempts to map in detail the domain of a company. The domain includes: companies, people, products, etc. The ontology is managed overtime to: 1. remove outdated information and errors and 2. add new information. The company ontology uses all the information in the ontology to select news stories.

Text Analysis : This chapter describes classification of news stories. The section provides a description of an initial grammar based approach as well as a restricted self training strategy. The section provides an evaluation with: a gold standard, cross validation (F-measure) and trading evaluation.

Direct Speech Analysis : This chapter provides a description of two strategies for classifying direct quotations from news stories. The first strategy splits the corpus into separate groups and applies individual semi-supervised techniques to each group. The second strategy is a guided self-training strategy which allows the learner to select high confidence individuals to add as training data. High precision rules correct erroneous selections and consequently boost the F-measure of the induced learner.

System Experiments : This chapter documents the experiments which integrates the work described in the previous chapters. The competing information retrieval strategies select, rank and score news stories for four companies: Apple, Microsoft, IBM and Google. The text and direct speech classification strategies classify the information contained in the selected news stories. The information gathered from the news story is added as extra features to existing technical indicators features which were gathered by an existing stock trading system. The experiments used the existing stock trading system to make predictions of the share price of the four companies. The experiments describe the stock prediction results for the competing information retrieval and classification

strategies for: 1,3,5 and 10 days ahead. The competing strategies are evaluated against a baseline which uses technical indicators only as features.

Conclusion : This chapter discusses the work conducted for the thesis as well the direction of future work.

Appendix : The appendix contains information which was not included in the thesis, but nevertheless may informative to the reader. For example, full definitions of terms and software tools used in this thesis.

Chapter 2

News Web Crawler and Text Extraction

2.1 News Crawler

The "raw material" for this project was news stories. There was no large, freely available set of news stories to use for this project and therefore a large set of documents was required to be scraped from the Internet. This chapter will describe the crawler, text extraction and addition of meta-data parts of the system.

The functional requirements of the crawler were to: 1. scrape news stories from pre defined lists of sites, 2. autonomously extract the story text from the original HTML, 3. assign a published date to the news story, 4. identify the headline, 5. identify duplicate news stories from the same source, 6. be able to run against a pre-defined scheduled and 7. run without human intervention.

The functional requirements required the identification of a published date and headline of the news story. This information was available in Really Simple Syndication (RSS) feeds provided by the news web sites. A RSS feed is a XML dialect which encodes information about a series of news stories. A RSS feed includes: a published date, headline and a uniform resource locator (URL) which provides a link to the news story. The RSS feeds typically have a "linguistic cue" in their URL which indicates the content of the feed, for example a RSS feed URL which contains the term "cricket" would be expected to contain stories about cricket whereas a RSS feed URL which contained the term "finance" would be expected to contain stories about finance.

The process of scraping news stories had two distinct parts: a crawler which fetched news story HTML data as well as associated data from the RSS feed XML and a story text extractor which extracts story text from the news story HTML. The *RSS crawler* relied

upon a list of pre-defined RSS feeds. The RSS feed discovery used aggregator sites, for example, the site `http://dave.org.uk/newsfeeds/` lists a large number of RSS feeds which can be used to gather candidate RSS feeds. The RSS crawler used a manually constructed list of "linguistic cues" to identify relevant RSS crawler feeds from aggregator sites. The cues were: "finance","business" and "company". A RSS crawler feed was deemed relevant if it had one of these cues in its URL. In addition to automatically located RSS feeds, "add hoc" additions of new RSS feeds could be entered manually. A pre-existing API was used to parse the RSS feeds. The API returned: headline, description, news story URL and published date information. This information was stored in a MySQL database. There were often duplicate news stories because: 1. RSS feeds with different URLs from the same web site publish the same news stories and 2. RSS feeds were not updated between crawler runs. Duplicate detection was achieved with: 1. headline of a news story and 2. the RSS feed's domain. A RSS feed's domain was the URL of the site the feed was hosted on, for example all RSS feeds hosted on The Telegraph web site would have a domain of "telegraph.co.uk". The use of RSS feed domain and headline allowed the detection of duplicates across multiple RSS feeds on the same site. If a duplicate news story version was detected then the story with the most recent published date was deleted. If the published dates were identical then the story which was crawled last was deleted. Once a crawl of RSS feeds had been completed and duplicates deleted, the story HTML for each freshly crawled news story was fetched from its URL contained in the news story's RSS feed information.

2.2 Text Extraction

The text extraction stage extracted the news story text from its HTML. It was necessary to have a fast and accurate extraction process because: 1. there were a large number of news stories (400K news stories gathered over 3 years), 2. news story HTML pages can contain non news information, for example advertisements or user comments, which may "contaminate" the news story text with irrelevant text which would introduce errors in an analysis process, and 3. proliferation of web coding standards.

2.2.1 Initial attempts

The first attempt was to hand-code extraction templates for each individual site. The extraction templates relied upon "landmarks" in the HTML to detect outline of the story text. A landmark is a HTML element or section of text which delimits the news story text. The major news web sites often required multiple extraction templates because there were individual coding standards for each separate section of the site. In addition the news web sites changed the coding standards on an irregular basis which necessitated a change in the manual extraction template. If the change in the coding standard went undetected then

incorrect text would be harvested until the manual extraction template was modified. A news story corpus constructed using site specific extraction templates contained documents with: 1. no text, 2. HTML and 3. erroneous text. The corpus had to be actively managed on a daily basis. The active management process failed to remove all erroneous documents. A detailed evaluation of the corpus revealed that: 4% of all documents failed to extract the story text, 2.5% contained only HTML code and 20% contained some HTML code. In summary the manual extraction template approach requires: regular changes to the extraction templates as well as active management of the news story database to eliminate erroneously extracted news story text. The results were unsatisfactory.

2.2.2 Text Extraction Literature Review

The literature review revealed three distinct approaches for extracting text from web pages: a manual approach (method 1), wrapper induction (method 2) and automatic extraction (method 3) [Liu, 2007b]. The manual approach was described in the subsection *Initial attempts* on page 28. The manual approach was unsatisfactory for news text extraction because web site templates changed on an irregular basis.

The wrapper induction strategy is a supervised approach where representative samples of web pages are manually labelled. Extraction rules are then learnt from the labelled data. A weakness of the wrapper induction strategy is the detection of changes in HTML source code for a specific web site. A change in HTML source code may necessitate a change in extraction rules and consequently the manual labelling effort would have to be repeated. This problem is known as *the wrapper verification problem* [Liu, 2007b]. The repetition of generating extraction wrappers was sufficient to discount this method as a candidate solution for the text extraction stage.

An alternative strategy to methods 1 and 2 is an unsupervised method which can learn extraction rules dynamically. The literature review was restricted to method 3 because methods 1 and 2 were sufficiently flawed to be discounted as possible solutions to the text extraction problem. The unsupervised methods are typically predicated upon two methods: string matching or tree matching [Liu, 2007b]. String matching is the comparison of two strings to determine the similarity of the two strings. Typically measures such as Levenshtein distance are used to determine the similarity metric. This string based similarity metric is used to align similar sections of HTML code either: 1. in the same web page or 2, a similar web page from the same web site. An example implementation of this string based matching is the "centre star method" [Liu, 2007b].

The tree matching technique uses a Document Object Model (DOM) representation of a web page to compute a similarity metric with other web pages by comparing matching nodes. Liu [2007b] provides a number of specific tree matching algorithms to identify lists of data

items in web pages. These web sites are typically e-commerce sites which sell products, which presents the user with a large number of products in a form of a list. These methods were unsuitable for the text extraction problem because they were predicated upon repeating HTML elements which may not be available in news story extraction. An unsupervised tree matching approach for identifying items in news text was described by Reis, Golgher, Silva, and Laender [2004]. Their approach was predicated on a "tree edit distance". They define a "tree edit distance" as a minimum mapping cost between two trees. This can be a computational expensive operation, which can be typically $O(n_1 n_2 h_1 h_2)$ [Reis, Golgher, Silva, and Laender, 2004] where n_1 and n_2 are the size of the trees, and h_1 and h_2 are their heights. They used a top down restrictive tree comparison which reduces the complexity to $O(n_1 n_2)$ because the mapping cost is only computed in the leaves. They used tree edit distances to cluster together web pages from a single news site. Their assumption was that separate extraction rules would be needed for each cluster because of differences in DOM structure. The cluster extraction pattern was generated by matching each *DOM Tree* of each web page with the *DOM Trees* of the remaining web pages in the cluster. The cluster extraction patterns were expressed as node extraction expressions. Heuristics such as text length or number of terms in the text were used to identify the relevant text. A possible weakness in this approach is that the clustering process and the matching of DOM trees is a computational expensive task.

2.2.3 Proposed Strategy

The text extraction method used in the extraction process was a variation of a tree edit distance strategy. It used characteristics of news data to select comparison web pages which were used to generate text extraction patterns for a specific candidate web page. The extraction process relied upon the comparison pages having a near identical DOM structure to the candidate web page. The extraction process relied upon DOM text tree nodes from a candidate and comparison web page which matched, i.e. same location in a DOM tree, and had a differing text value. These text candidates were extracted based upon their size and location in the DOM tree.

The comparison web pages were selected by grouping together web pages by their: 1. RSS URL, 2. publication date and 3. web page length which is a count of the characters contained in a web page. The assumption was that pages with the same RSS URL, similar publication dates and page length would have similar DOM trees, and static content. The two web pages which: 1. had the same RSS URL, 2. published within 30 days of the text extraction page and 3. had the closest page length to the text extraction web page were selected as comparison pages. The difference between candidate web page's lengths were calculated by the following equation,

$$d(page1, page2) = |len(page1) - len(page2)| \tag{2.1}$$

where *page*1 and *page*2 represent candidate web pages, *len* represents a count of characters (web page length) in the candidate pages, and d represents the distance between the two pages.

A final comparison page was selected by calculating a tree edit distance for each comparison page with the candidate text extraction page and selecting the comparison page with the lowest tree edit score. This was to ensure that the web page with the most similar DOM tree to the candidate text extraction page was selected for the text extraction phase. The selection strategy was more efficient than the Reis et al clustering strategy [Reis, Golgher, Silva, and Laender, 2004] because it limited the number of tree comparisons to two per candidate page instead of the $n-1$ (n=total of pages per web site), used by a clustering process. An example of a *candidate extraction web page* and a matching web page which was selected through this process is presented in Figure 2.1. The example shows that the layout and non news content of each news story is almost identical. The selection algorithm is described in Algorithm 1.

News text extraction from the candidate web page was achieved by locating matching text tree nodes from the DOM trees of the comparison page and the *candidate extraction web page*. A matching text node was a text node which was located in the same position of both DOM trees. A simple example is demonstrated in Figure 2.2. The text extraction algorithm was a two step process: step one located sub-trees of the "Body" HTML tag from the candidate extraction page and its comparison page and step two match the nodes from the sub trees located in step one. The step one algorithm is described in Algorithm 2 whereas the step two algorithm is described in Algorithm 3.

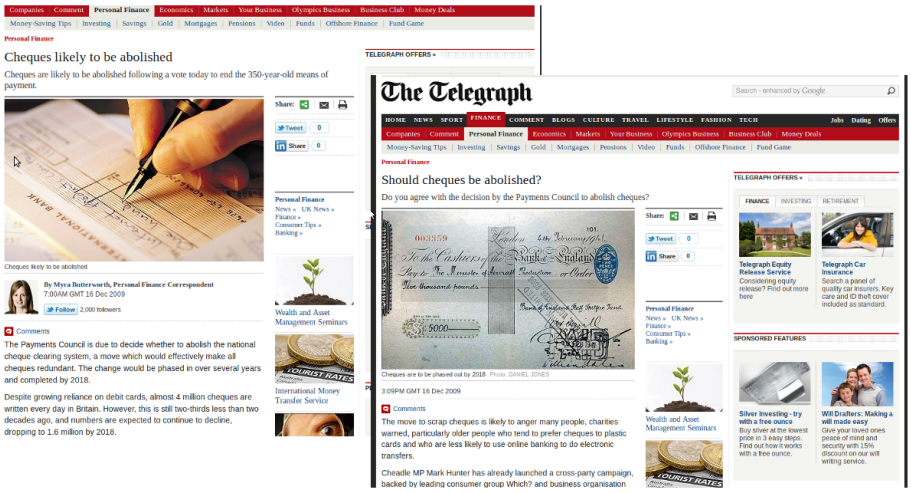

Figure 2.1: Sample Comparison and Target Page.

Algorithm 1: Initial web page Candidate Selection.

Input: candidate – Text Extraction Candidate

Input: documents – News Stories

begin

```
tempstore ← ()
forall the document ∈ documents do
    if (document.RSSURL = candidate.RSSURL and
    |document.pubdate − candidate.pubdate| < 30 days) and
    document.headline ≠ candidate.headline) then
        document.difference ← |document.length − candidate.length|;
        tempstore.add(document);
 //Sort by ascending document length difference;
tempstore ← tempstore.sort();
 //Add "Closest" News Stories;
candidates.add(tempstore[0]);
candidates.add(tempstore[1]);
return (candidates)
```

The matching nodes were disregarded if they contained the same text (static content) as this was non-news text common to large numbers of web pages from the same web site. News story text varies from web page to web page because news story cover individual events therefore matching text nodes from the candidate extraction with different text values instead of the matching nodes from the comparison page, were reserved for extraction.

The text extraction process from the matching nodes relied upon text length to extract text. The matching node with the longest text value was located. The matching nodes which were not on the same DOM tree branch as the longest text node were eliminated. The final text was extracted from the largest text mode, and the remaining text nodes if their text value had a length of at least 30% of the largest text node. This cut off value was derived through simple experiments. A simplified algorithm for extracting text from matching text nodes is described in Algorithm 4.

2.2.4 Evaluation

The evaluation of the proposed text extraction strategy was by comparing the text of a "gold standard" set of news stories with the text extracted by the proposed strategy from the HTML of the "gold standard". The "gold standard" texts had been extracted with the initial

Algorithm 2: Text Node Match Part 1.

Input: Candidate - Candidate Extraction Page
Input: Comparison - Comparison Page
Output: CandNode - Collection of Matching Nodes
begin

```
CandidateDOM ← CreateDOM(Candidate);
ComparisonDOM ← CreateDOM(Comparison);
ChildNodes ← CandidateDOM.getBodyChildNodes();
CompNodes ← ComparisonDOM.getBodyChildNodes();
//variables for use in the second algorithm;
//rootno tracks which branch the algorithm is on;
//depthno tracks how far down the the algorithm is on;
rootno ← 0;
depthno ← 0;
candidates ← ();
for x = 1 → ChildNodes.Length do
    rootno = rootno + 1;
    //Return Matching Node Candidates for text extraction;
    Push(candidates, Algorithm3(ChildNodes[x], CompNodes[x], rootno, depthno));
return(candidates)
```

manual template system and verified by a human annotator. The aim of the experiment was to discover the similarity of the extracted text with the "gold standard" text. The similarity of the "gold standard" texts with the text extracted by the proposed method was calculated by the Simmetrics library [Chapman, 2008] which is a library which implements a number of common similarity measures. The similarity functions in Simmetrics produced a value from 0.0 to 1.0 where 0.0 indicated that there was no similarity and 1.0 indicated perfect similarity. The experiments used three similarity measures which was to ensure that the results could not be skewed by weaknesses in a single measure. The measures were: Qgrams, Jacard Distance and Cosine.

The "gold standard set" contained 2310 manually evaluated news texts. Each text in the: 1. gold standard set and 2. texts extracted by proposed strategy (evaluation set) were: 1. converted to lower case, 2. the spacing between the words was changed to a uniform 1 character space and 3. tabs and newlines were removed and replaced with a single space. This was to ensure that the formatting of the text was uniform and consequently any dissimilarity between text would be due to differences in the text.

The experiment iterated through the evaluation set of documents. The three similarity measures were computed for each text against the same text from the gold standard. As a

Algorithm 3: Text Node Match Part 2.

Input: Cand - SubTree From Candidate Extraction
Input: Comp - SubTree From Comparison Extraction
Input: RootNo - Position number of sub-tree
Input: DepthNo - Depth Number of Parsed Root
begin

```
//Place holder for text candidates;
cands ← ();
forall the child ∈ Cand.getChildNodes() do
    child2 ← Comp.getNextChildNode()
    //Termination because no corresponding node in comparison sub-tree;
    if (child2 = NULL) then
        return(cands)
    //Detect if node is start of a sub tree;
    if (child2.isRoot() ∧ child.isRoot()) then
        //Recursive Call of Text Match Function to add matched candidates;
        push(cands, TextNodematch2(child, child2, RootNo, DepthNo + 1));
    else if (child2.isTextNode() ∧ child.isTextNode()
    ∧child.Text ≠ child2.Text) then
        child.RootNo ← RootNo;
        child.DepthNo ← DepthNo;
        push(cands, child));
return(cands)
```

Algorithm 4: Text Candidate Extraction.

Input: cutoff – minimum length for text extraction
Input: textcandidates – list of text candidates identified through node matching process
begin

```
//Place holder for text from candidate nodes;
text ← "";
forall the candidate ∈ textcandidates do
    //Check length of candidate node text value;
    if (len(candidate.text) < cutoff) then
        return(text)
    text ← append(candidate.text, text)
return(text)
```

comparison four variations of the proposed strategy were used. The variations were in the strategy for selecting comparison documents. The variations selected comparison documents

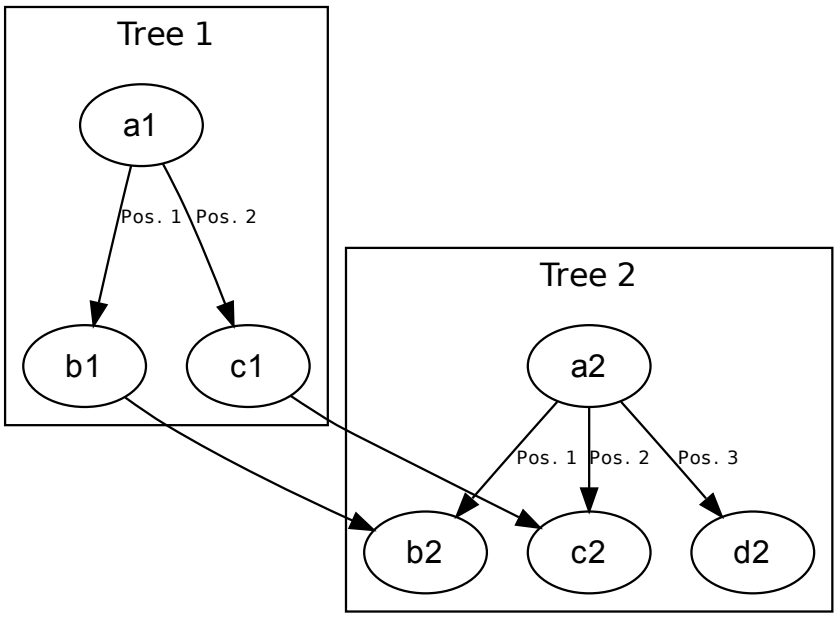

Figure 2.2: Example of matching text nodes.

by: 1. web page length, 2. web page length and RSS feed, 3. web page length and time and 4. random. The results are presented in Figures 2.4, 2.3, 2.6 and 2.5. The results for the proposed strategy were good as the set of extracted texts were almost identical to the gold standard set. There were exceptions where larger advertisement text was not filtered by the extraction rules. The variations of the proposed strategy did not perform as well because the selection of comparison documents allowed large common texts to be present in the extracted text. The common texts were often: time dependent, feed dependent or a combination of the two. This is highlighted by the performance of the time and length strategy which limited the search for documents within a thirty day period which outperformed the proposed strategy contenders on all similarity measures with the exception of Cosine where the Simmetrics library returned an error.

The length of a web page can often indicate a common template was validated because using web page length was often sufficient to extract the news text, but the weakness in using only web page size was that it could not detect changes in common text because text was a relatively small part of a web page. A combination of restrictions by length, RSS feed and time allowed the selection of comparison web pages which enabled near optimal text extraction. The base line comparison of choosing a random page was poor because in the

experiments no text was extracted.

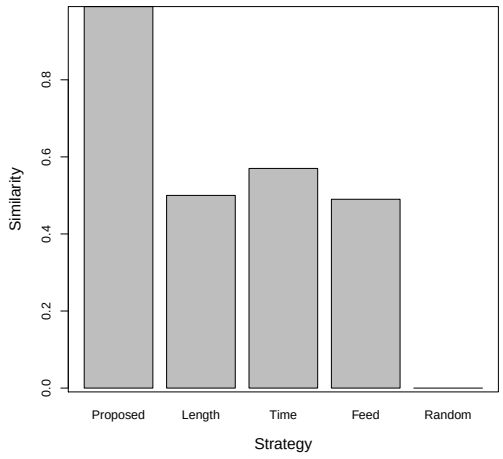

Figure 2.3: Jacard similarity for competing text extraction strategies.

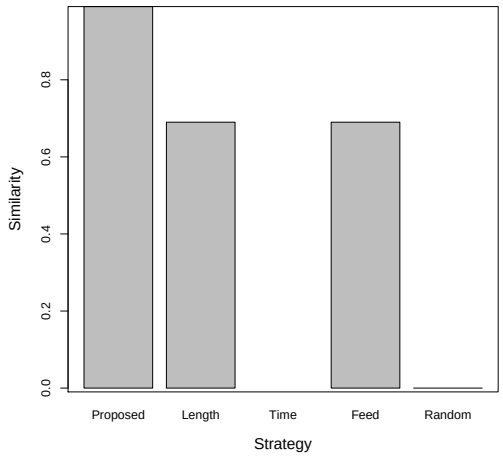

Figure 2.4: Cosine similarity for competing text extraction strategies.

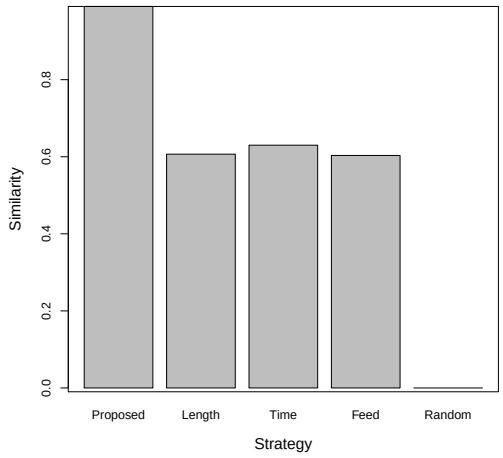

Figure 2.5: Mean similarity for competing text extraction strategies.

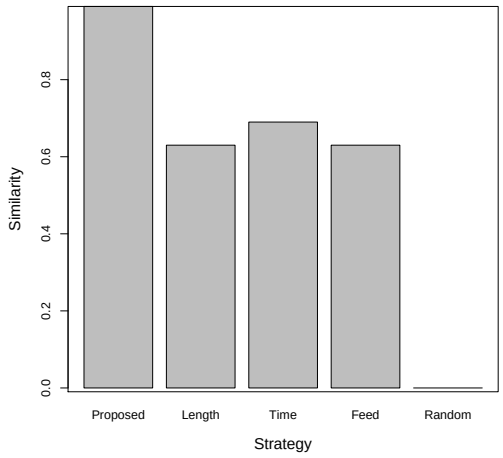

Figure 2.6: QGram similarity for competing text extraction strategies.

The proposed strategy is an autonomous and efficient tree based text extraction strategy. It is more efficient than the strategy proposed by Reis, Golgher, Silva, and Laender [2004] because it performs fewer tree comparisons. The proposed strategy performs a maximum of 2 tree comparisons per candidate page where as Reis, Golgher, Silva, and Laender [2004] performs $n-1$ (n = number of pages on web site) tree comparisons for each target page during the clustering phase and $c-1$ (c = number of pages in cluster) during the node extraction rule calculation stage. The production version of this strategy increased the number of pages

processed per day by using the following strategies: 1. "caching" the newly created web page DOM trees to ensure the same DOM tree was not created twice and 2. storing the tree edit distances between 2 pages if the comparison page's text had not been extracted. Storing of edit distances ensured that the computing of tree edit distances between two pages only occurred once.

2.3 Addition of Meta-Data

Addition of meta-data in this context are the following tasks: 1 named entity identification, 2. text categorization, 3. fact and event identification and 4. identification of relationships between named entities. There are a number of web services which can perform these tasks, for example Zemanta or Open Calais. These services are free. Open Calais was chosen because it was designed specially for news. Open Calais describes the meta data in Resource Description Framework (RDF) language and embedded in the language are links to "Linked Data". Linked data is data which has been recovered from reliable sources such as DbPedia or Reuters databases. The data is described in RDF.

The addition of meta data was achieved by running a separate program on a schedule to process newly extracted text. The text is compressed, sent via a Representational state transfer (REST) API call. The program waits for a reply. The resultant RDF is checked for any error messages. If there is an error then text is resent until the text is processed or it has been resent for a period of 24 hours. If the text has been processed successfully then it is stored, otherwise the text is inspected for the cause of any processing errors.

The following chapters often rely upon information contained in the meta-data. To enable the extraction and processing of information contained in the meta-data a custom library was developed. The rational was to insulate the developer of applications which processes the meta data from writing RDF specific code. The library parsed the code and produced a series of objects which contained attributes and references to other objects.

2.4 Summary

This chapter has described the manner in which stories are crawled and processed. The processing involved the extraction of the story text as well as the addition of meta data. This data is the "building blocks" on which the rest of the thesis is built. The meta data is not complete because Open Calais does not identify "industry sectors". This issue will be addressed in the next chapter. The work produced in relation to this chapter produced a RSS crawler which was fast, robust and extracted text without the need for manual intervention.

Chapter 3

Ontology-based Information Retrieval

This chapter details an information retrieval system which uses ontologies to identify strongly related news to a "monitored company". The information retrieval was necessary because the initial experiments with freely available news stories revealed that there was limited amounts of news for individual companies. There were often large intervals where there was no news available in our corpus for large well known individual companies. For example, in the news story corpus there were only four news stories which directly referred to "Microsoft" published between the 8th of January 2009 and the 16th of January 2009, three of which were published on the 8th and one on the 16th. A trading system which relied on these small volumes of news would only be able to trade infrequently.

A proposed solution to limited volumes of company specific news was to use directly related news stories. Directly related news is news which does not mention a company by name, but has a strong connection with a monitored company. For example, a news story concerning computer technology is directly related to "Microsoft" because Microsoft produces computer software and hardware. The use of related news to estimate the direction of a company share prices utilizes an effect known as "spill-over" [Hafez, 2009] where news from a single company can provoke changes in share prices in a large number of related companies. Related news can be located by identifying terms in news stories which have a identifiable relationship with the monitored company. Terms can be any form of linguistical unit, for example, unigrams, bigrams or multiword expressions. A common strategy for identifying related news is to use ontologies which describe relationships between nouns. The relationships are described with verbs, for example *Bill Gates* (proper noun) *works*(verb) for *Microsoft* (proper noun). In this case, the proper noun *Bill Gates* has an identifiable relationship with Microsoft, consequently news stories which contain the proper noun *Bill Gates* could be used as related news to Microsoft. There are a number of pre-built ontologies which may be used in identifying related news stories. The initial experiments for this chapter used two pre-built ontologies: Cyc [Lenat, 1995] and ConceptNet [Liu and Singh, 2004].

ConceptNet assertions rely upon the *Open Mind Common Sense Project* [Singh, Lin, Mueller, Lim, Perkins, and Zhu, 2002] which used a website to allow people to enter "common sense" assertions about specific subjects. ConceptNet has a large number of assertions, but an initial evaluation in 2009 revealed that there were errors in ConceptNet caused by vandalism and a lack of information about companies. Experiments using ConceptNet to identify related news produced poor results. Cyc is also a large "common sense" ontology which included information about companies. An evaluation in 2009 of the open source version of Cyc (OpenCyc) revealed that the information about companies lacked detail. In addition the update schedules of OpenCyc were not frequent, consequently out of date information in OpenCyc was not corrected rapidly. Experiments in 2009 with OpenCyc to identify related news were not successful. In summary, the available pre built resources were not suitable for the task because of: 1. errors, 2. out of date information and 3. lack of company specific information.

The proposed solutions used ontologies to map relations between nouns (entities) to identify terms in news stories which had a relationship with a monitored company. The first approach created an ontology which mapped relations between industry sectors. Industry sectors were entities which described a collection of companies, for example "car industry" describes a collection of companies which produce cars. Companies can be members of more than one industry sector. Related news for a monitored company was selected by identifying news which contained members of the same industry sectors as the monitored company as well the industry sectors the companies were a member of. The second approach mapped the domain of a monitored company on a day to day basis. Outdated information was removed whilst new information was added. Related news was identified by locating company specific entities in news stories. In both approaches, the related news stories were outputted in XML for further experimentation.

The ontology research field uses various terms which represent a similar notion. This paragraph will define a series of terms and their meaning which will be used in this chapter. An ontology is a computational ontology which formally models a structure of a system [Staab and Studer, 2004b]. An ontology contains: concepts, entities and relations [Staab and Studer, 2004b]. An entity represents the most "general being" and incorporates: subjects, objects and processed items [Staab and Studer, 2004b]. Related entities can be grouped together into concepts such as People and Companies. Entities are linked by relations. The relations describe the nature of relationship between the entities. An assertion is an instruction which creates: relation, concept or entity. A reassertion is a repetition of a previous assertion. An attribute is a value assigned to either: a relation or entity.

The chapter uses terminology to describe the life cycle of an ontology. This paragraph will describe the terminology. The ontology life cycle can involve the following steps: 1. construction, 2. population, 3. maintenance and 4. evaluation. The construction phase

involves creating the structure of an ontology. The creation of a structure of an ontology typically involves the creation of concepts. The population step is the population of concepts with entities and the linking of entities. The maintenance phase involves the addition of new entities and the removing of erroneous entities. The evaluation phase involves the estimation of the fitness of the ontology for its designated purpose.

This chapter will describe the following: 1. related work, 2. industry ontology, 3. company specific ontology and 4. ontology news recall. The related work describes previous work which is related to the work described in this chapter. The industry ontology section describes a method of constructing an ontology based upon the "natural grouping" of companies under related industries. The company specific ontology describes a method of constructing an ontology for a specific company and adapting it over time to allow for changes in the company's domain. The ontology news recall compares the performance of both the industry and company specific ontology for identifying related news in a information retrieval system for a specific company.

3.1 Related Work

The literature search for this chapter encompassed it's main contributions: 1. ontology construction / maintenance, 2. news recommendation and 3. query expansion. This section will describe the relevant state of the art for each of the areas which influenced the work in this chapter.

3.1.1 Ontology Construction

There are a number of generic works which propose general methods for constructing ontologies from text. Lee, Kao, Kuo, and Wang [2011], proposed concept clustering and episodes. An episode was a triple of: term, Part of Speech Label(POS) and location in a sentence. Dahab, Hassan, and Rafea [2008] suggested the use of semantic patterns, for example synsets from WordNet [Fellbaum, 1998], to construct an ontology. These strategies may be effective as general methodologies, however there was no evaluation of these techniques on financial news. Financial news domain has specific problems which these strategies do not address, for example the identification of similar products or financial instruments.

There have been some attempts to model information gathered from the news [Newman, Chemudugunta, Smyth, and Steyvers, 2006][Vargas-Vera and Celjuska, 2004] [Lloyd, Kechagias, and Skiena, 2005], but the knowledge captured was for a demonstration of a technique, rather than to produce a rich representation of background knowledge. Newman, Chemudugunta, Smyth, and Steyvers [2006] constructed topic maps using co-occurrence of named entities in related topics from stories published in the "The New York Times". The

topic maps were a representation of New York Times stories rather than a detailed illustration of the financial / business domain. Vargas-Vera and Celjuska [2004] populated a hand-crafted ontology with news information, but there was no error control. Finally, Borsje, Levering, and Frasincar [2008] used a hand-crafted ontology to represent a general financial domain which was used to select news.

There are a number of semi-supervised / unsupervised strategies which use a seed set of concepts to construct an ontology. These concepts are expanded and consequently the ontology is enriched with new information. Levering, Frasincar, and Borsje [2009] demonstrated a strategy for constructing a domain ontology from news stories. The strategy followed was: 1. user selects seed set of news stories, 2. extract concepts from seed set of news stories and 3. populate the ontology with extracted concepts. The concepts in the domain ontology were used in a news recommender to select news stories. There are some drawbacks to this process: 1. humans may select a small or inadequate number of stories to generate an adequate ontology, 2. no adaptation of ontology to accept new information or remove outdated information.

It is possible to expand a small manually selected seed set. The literature review revealed some relevant approaches. For example, concepts can be expanded with synonyms from WordNet [Fellbaum, 1998], i.e, the concept "Car" can be expanded to auto, automobile, machine, motorcar. There have been attempts to measure the relevance of synonyms to seed concepts which can assist in the expansion of a domain ontology [Lee, Kao, Kuo, and Wang, 2011]. General linguistic resources were not suitable for the target domain because they do not contain domain specific language, for example, WordNet does not contain synonyms for common business terms such as companies, products, etc. This ensured that the aforementioned strategies could not be adapted for the problem domain.

3.1.2 Ontology Maintenance

The research literature revealed a number of approaches for ontology maintenance. The target domain is dynamic and a derived ontology would require the addition of new information and deletion of errors and outdated information. The literature review discovered semi-automatic methods [Gargouri, Lefebvre, and guy Meunier, 2003] and strategies for collaborative environments [Noy, Chugh, Liu, and Musen, 2006] for ontology management. These techniques were used to expand ontologies rather than to provide active error control.

Facts in business news can be volatile because of the dynamic nature of commerce. A source of errors in a company specific ontology could be outdated relations which represent an assertion which was once true. A determination of the life of a relation and its removal from the ontology when the relation's life has expired may reduce this type of error. The literature review yielded an approach which assumed a relation to be true until a contravening

event [McCarthy, 1986]. A contravening event renders the relation false. For example, a person is employed by a company until the employee is: 1. dismissed, 2. resigns, 3. retires or 4. dies. The contravening event is any item of the previous list. A problem with the mass media is, it does not publish all available news [McManus, 1988] and consequently a contravening event may not be published. A maintenance strategy must be able to infer a contravening event. A possible solution was proposed by Dean and Dean and Kanazawa [1988] who used temporal reasoning to infer a contravening event by using a lattice of discrete previous events partially ordered in time. The lattice was used to compute a probability of a contravening event. A high probability of a contravening event would invalidate a relation. This approach relied upon previous events being known. This approach would be unsuitable because: 1. economics of news publishing and 2. companies are not compelled to make all their data public.

An ontology which changes overtime can be often referred to as a "dynamic ontology". An ontology which has been favourably evaluated at the end of construction phase may at other specific points in time not accurately represent the target domain. An evaluation of the evolution of the ontology over time would be required. There have been a number of attempts to evaluate dynamic ontologies over time. Murdock, Buckner, and Allen [2010] proposed two metrics: 1. volatility score which measures structural stability over time and 2. violation score which measures the semantic fit between an ontology's taxonomic structure and the distribution of terms in an underlying text corpus. These measures make an explicit assumption that the ontology should be stable overtime. Facts in news are inherently unstable and therefore any ontology generated from it would be volatile. This feature of news ensures that the evaluation scheme proposed by Murdock, Buckner, and Allen [2010] would be unsuitable for this domain.

An alternative to evaluating dynamic ontologies for errors is to control the evolution of the ontology overtime to minimise the addition of errors to the ontology. One approach is to use "design patterns" to identify errors and propose resolutions [Djedidi and Aufaure, 2010][Djedidi and Aufaure, 2009]. Djedidi and Aufaure [2010] used three design patterns in a pipeline. The patterns were: change, inconsistency and alternative. The change pattern is a process by which a change in an ontology is specified. The inconsistency pattern identifies possible logical inconsistencies in the ontology created by the proposed change. The alternative pattern provides resolutions to the logical inconsistencies in the ontology identified by the inconsistency pattern. A variation of this approach was proposed by Scharrenbach, d'Amato, Fanizzi, Groutter, Waldvogel, and Bernstein3 [2010]. They used description logics, which is a form of schema for ontologies, to control ontology evolution. This approach also concentrated on resolving logical errors in an ontology. These techniques did not consider resolving errors which were due to false information because there was no ground truth to compare the ontology against.

3.1.3 Query Expansion

Query expansion is relevant for the target task of retrieving news that is relevant to a target company because it is not possible to manually construct a definitive query to locate all possible relevant news stories in a corpus. An automatic process of identifying relevant terms is required.

Query expansion is *the process of reformulating a seed query to improve retrieval performance in information retrieval operations.* For example, original query keywords can be expanded with synonyms from lexical resources [Voorhees, 1994] such as WordNet [Fellbaum, 1998]. The advantage of this approach is that it was automatic and consequently the term expansion could be large. Large "common sense" ontologies such as Cyc [Lenat, 1995] or ConceptNet [Liu and Singh, 2004] can be used to expand initial queries with similar concepts which may not have a lexical relationship, but a semantic relationship [Akrivas, Wallace, Andreou, Stamou, and Kollias, 2002]. A "semantic query" may use named entities [Akrivas, Wallace, Andreou, Stamou, and Kollias, 2002] as keywords in the query, for example, company names. A hypothetical query expansion could be to expand the original company name with names of companies which manufacture similar products. There is one further alternative to the lexical and semantic query expansion, corpus based query expansion. Corpus based methods rely upon statistical association between terms in a corpus [Bhogal, Macfarlane, and Smith, 2007]. A typical use of corpus methods is "term clustering". In the example described by Bhogal, Macfarlane, and Smith [2007], the document collection was clustered, and if the terms in the query appeared in n documents (n being a pre-set constant) in the cluster then the whole cluster was returned. The assumption was that documents in a cluster contained language related to the initial query term.

To summarize, there are three main approaches to query expansion: 1. lexical, which is dependent on language specific relations between keywords, 2. semantic, which is dependent on semantic relations between keywords and 3. corpus, which is dependent upon statistical associations between keywords.

3.1.4 News Recommendation

News recommendation is a specific type of information retrieval which uses latent relationships in news to "personalize" news content for a specific reader. There are arguably two types of news recommendation: content based and semantic based [IJntema, Goossen, Frasincar, and Hogenboom, 2010]. The content based systems use statistical measures to recommend news. For example: the "NewsDude" and "YourNews" systems use a Term Frequency Inverse Document Frequency (TF-IDF) which is a weighting scheme which computes a term's weight in a specific document by its frequency in the document compared its frequency in the whole document collection [Liu, 2007c]. Content based systems are not limited to TF-IDF

indexing, but can use "article similarity" in a collaborative filtering environment [Kompman and Bielikova, 2010] to recommend the *most similar articles* to the last read news story. The inherent weakness of the content based systems is that they ignore the latent relations contained in news stories.

A competitor to the content based systems are the "semantic" news recommender systems (SNRS) [IJntema, Goossen, Frasincar, and Hogenboom, 2010][Cantador and Castells, 2009]. The SNRS systems often use a domain ontology to represent latent relations in a specific domain [IJntema, Goossen, Frasincar, and Hogenboom, 2010]. The domain ontology can store "concepts". A concept is an instantiation of a class which holds values and relations to other classes. A class is a collection of concepts which can be contained in a pre-determined hierarchy. For example Apple could be of class "company". The SRNS systems have the advantage of knowing the class of the news terms they are evaluating. In the previously mentioned "Apple" example a content based system may return stories about: 1. fruit , 2. The Beatles (Apple was a company founded by The Beatles in the 1960s) and 3. Apple - the computer company. The SRNS systems will able to evaluate the "class" of the concept and return stories from the target domain - Apple - the computer company. The SRNS systems can use a variation of statistical measures to compute the "importance" of the concept to the total document collection, for example: Goossen, IJntema, Frasincar, Hogenboom, and Kaymak [2011] proposed the Concept Frequency Inverse Document Frequency (CF-IDF) which is a variant of TF-IDF. CF-IDF computes the relevancy of a news story by: 1. identifying concepts which co-occur in a news story, 2. calculating the frequency of the concepts in the news story, 3 . computing the frequency of the concepts in the whole document collection and 4. normalizing the news story score by the number of times the concept appears in the collection. In addition to knowing the class of a concept, SNRS systems can recommend similar news stories by identify articles which contain related concepts, for example news stories about competitors [Goossen, IJntema, Frasincar, Hogenboom, and Kaymak, 2011].

The SRNS are not limited to using one general domain ontology to identify latent relations in news. The news story itself can be represented as a specialized ontology, a taxonomy. Mannens, Coppens, Pessemier, Dacquin, Deursen, and de Walle [2010] used a variety of specialized web services (Open Calais, Zemanta, etc.) to extract named entities and affix an unique resource identifier (URI) to the extracted relation. The relations between each named entity were identified and incorporated into a *global knowledge base*. The *global knowledge base* was used in conjunction with user profiles to recommend news. The user profiles contained the "habits" of the user, consequently information in the *global knowledge base* could be used to identify related stories which had a relationship with the concepts stored in the user profile of a specific user. An alternative approach proposed by Lašek [2011] was to use a pre-computed general ontology, DbPedia [Auer, Bizer, Kobilarov, Lehmann, Cyganiak, and Ives, 2007], to identify relations between named entities in news stories. DbPedia is a project to extract structured information from Wikipedia and represent it as a knowledge base. The

resource is described in RDF and therefore entries are linked to each other with a description of the relation. Lašek [2011] used DbPedia to identify relations between named entities in news stories to recommend news.

3.2 Industry Ontology

The aim of the work described in this chapter is to locate relevant news to a target company through a query expansion process which uses a lexical resource. The first proposed method is to construct an Industry Ontology for use in a future query expansion process.

Related companies can be grouped together under a single moniker which describes the function of all of the related companies, for example *The Car Manufacturing Sector* where each company grouped under this title will be involved in the construction of vehicles. Phrases or terms which describe a related group of companies are often referred to as business or industry sectors because they refer to a subset of activity of the general economy. The identification of industry sectors and their related companies can assist in the determination of the economic prospects of an individual company by propagating the effects of industry sector level news to the related companies. In this context news which is relevant to an industry sector should be relevant to its member companies.

A company can be a member of more than one industry sector. For example, Microsoft produces an operating system for mobile devices as well as for personal computers, consequently Microsoft can be a member of an industry sector which represents: mobile computing, software development, computer technology and telecommunications. This basic example demonstrates that the manual assignment of companies to industry sectors would have taken a significant amount of time.

The work undertaken had two aims: 1. identify industry sectors in news stories and 2. assign companies to each industry sector. The first step is a named entity extraction phase which extracts industry sectors and adds them to the ANNIE Gazetteer. The second step uses company names and industry sectors from the ANNIE Gazetteer as well as predefined rules to assign companies to industry sectors. The hierarchical relationship between industry sectors and companies was represented as an ontology.

3.2.1 Industry Sectors Identification

The named entity extraction phase used linguistic patterns to identify industry sectors. This technique was preferred to supervised machine learning techniques [Nadeau and Sekine, 2007] because a supervised strategy may require large amounts of labelled data which would have required a significant manual resource which was not available. The initial experiments required a series of seed extraction patterns (regular expressions) to extract an initial sample

of industry sector titles. A possible hypothesis was a Noun Phrase (NP) which contained the word "industry" would be an indicator of an industry sector. Evidence was required to support the hypothesis and therefore a list of words which were part of known titles was created. The list included words: "car", "banking", "finance", etc. This list will be referred to as *The Known Industry Sector Variable List*. A regular expression was created where one word either side of the words from the Known Industry Sector Variable List was extracted. For example, the NP *the car industry* would be extracted as "the ... industry". The extracted phrases were counted. A sample of the words returned by the regular expression can be found in Table 3.1. The most frequent NP extracted by the regular expression contained the word: *industry*. This simple experiment provided partial evidence to support the initial hypothesis.

Frequency	Pattern
1046	the ... industry
952	GAAP ... measures
906	the ... sector
815	the ... crisis

Table 3.1: Extracted phrase delimiters for known collective entities.

The industry section extraction task was not limited to NPs containing the word, "industry" because it would limit industry sector extraction to a subset of possible industry sector titles. It was possible to increase the number of industry sectors detected by expanding the anchor term, "industry" with synonyms. The synonyms were taken from The Oxford Thesaurus Of English [of English, 2010]. This list will be referred to as *The Industry Synonym List*. A new regular expression was created to extract three word phrases where list items from The Industry Synonym List constituted the second word. A manual inspection indicated that a large number of phrases were not industry sector titles. These phrases contained one or more of the following: stop words, numbers, continuations, adjectives, verbs, comparisons, and quantifications. A number of new regular expressions were created, which were variations of the previously described pattern. The variant patterns contained one or more refinements. Table 3.2 describes the regular expression refinements as well as its subsequent accuracy and number of phrases returned.

Pattern	Accuracy	No. Phrases Returned
the [Word] [Industry Synonyms]	0.33	23344
the [Any Word] [Industry Synonyms (-Industry)]	0.36	5033
[Word] [Word] Industry	0.72	3753
the [Word] Industry	0.98	1450

Table 3.2: No. = Number.

Each run was measured with the following criteria: accuracy and number of phrases returned.

It was not possible to verify all of the extracted industry sector titles because of the large number of candidate phrases, consequently the accuracy of each term was calculated by the verification of the 100 most frequent phrases. The candidate industry sector titles' frequency in the corpus followed a Zipf's distribution [Zipf, 1949] and therefore a low accuracy amongst the most frequent candidate titles would impair greatly the overall accuracy of the developed resource. It was not possible to calculate a recall figure because there was no exhaustive list to compare the titles against, consequently a "raw number of phrases returned" value was calculated.

A new regular expression was created for each item in The Industry Synonym List. The regular expression used the most effective refinement described in Table 3.2, which was the word "the" as the first word and the list item as the third word in the three word pattern. The 100 most frequent candidate industry sector titles were extracted. The second word of each correct candidate (industry sector variable) phrase was compared to the industry sector variable of candidate phrases generated by the industry pattern. For example, in the candidate phrase *The Finance Sector*, the word "Finance" would be the industry sector variable. Each correct industry sector variable identified by patterns without the word "industry" was also contained in the list of industry sector variables generated by the industry pattern. The final list was calculated by extracting the correct terms from the industry list and duplicating the phrase for each term. The duplication consisted of replacing the term "industry" with the new term.

The above experiments were with single term patterns. A further number of experiments were made with multi-word patterns. The patterns identified the word "the" and up to three words and then a word from The Industry Synonym List ("industry", "business", etc). There were substantially less extracted terms, and these candidate terms were hand checked. The terms were expanded in the same manner as the single term candidate phrases.

3.2.2 Assigning Companies to Industry Sectors

The ANNIE Gazetteer holds a small list of companies. It was necessary to expand the company list to ensure that there were a sufficient number of companies to assign to each industry sector. The expansion of the company list was achieved by extracting companies from the Open Calais meta-data. The Open Calais [Reuters, 2010a] web service uses a series of rules to identify named entities in text rather than extracting terms from a comprehensive dictionary, consequently the Open Calais erroneously identified terms as companies. A list constructed with all the "companies" from the Open Calais meta-data had numerous errors. A solution was to record the frequency of individual companies in the Open Calais meta-data. A "cut-off" figure could be computed and companies which had a frequency of less than the cut-off would be removed. The rational behind this strategy was erroneous company names appeared infrequently in the meta-data. The "cut-off" figure was calculated by setting an

initial cut-off figure and taking a sample of the company names with the lowest frequency. If there was an error then the cut-off figure would be increased until there were no errors in the sample. At the end of the exercise 42,823 company names were extracted.

The strategy for learning members of industry sectors relied upon the co-occurrence of a company with an industry sector. The text held in the corpus was parsed with the ANNIE Gazetteer. The location of each industry sector and company in each separate text was recorded. There were two alternate methods for computing the members of an industry sector:

- Company Proximity to the industry sector, i.e. industry sector and company co-occurred in the same sentence
- Headline proximity, i.e. industry sector was present in the headline and company was present in the text.

A manual inspection of sentences where industry sector and companies co-occurred revealed that there were a number of occasions where the company was not a member of the industry sector. The cause of the lack of relationship between the company and the industry sector was that the company was a 3rd party commenting on the status of the industry sector. For example, an analyst from financial services could be commenting on the economic prospects of the industry sector "Car Industry".

To assist the co-occurrence process two JAPE rules were constructed. The JAPE rules identified companies which had no direct relationship with the co-occurring industry sector. The rules were:

- Rule 1: Pattern = Consultant synonyms | of\|from\| | Company → exclude company
- Rule 2: Pattern = Possessive form of a company name | Person's Name → exclude company

An industry sector contained in a headline of a news story may indicate that the following news text has a strong relation with the industry sector [Andrew, 2007]. The proposed hypothesis was because there is a commonly accepted relationship between text and headlines, consequently there may be a relationship between companies in the news text and industry sectors in the headline. For example a news story published by Reuters had the headline: *Signs point to U.S. car industry's resurgence* [1]. The industry sector in the headline was the "Car Industry". The text made references to "Nissan Motor Co.", "Hyundai Motor Co." and "Ford Motor Co." who are car manufacturers and are part of the "Car Industry". This

[1] `http://goo.gl/QmgAc`

co-occurrence experiment populated the industry sector in the headline text with all the companies discovered in the news text.

The evaluation of the co-occurrence experiments was by manual inspection. It was not possible to validate every industry sector and their associated company members. The evaluation was by selecting a number of industry sectors and validating their members. There was a single evaluator. An industry sector member was deemed to be correct if: 1. it was known to the evaluator or 2. the membership could be confirmed through legitimate resources on the Internet, for example company websites.

There were two evaluation measures: accuracy and number of industry sectors returned. The accuracy figure referred to the percentage of companies which were correctly assigned to an industry sector. The range was from 0 (none correct) to 1.0 (all correct). The returned figure referred to the total number of companies assigned to an industry sector in the extracted sample. The results are displayed in Table 3.3.

Experiment Type	Accuracy	No. Ind. Sectors Returned
Headlines	0.66	630
Same Sentence	0.77	3535

Table 3.3: No = number and Ind. = Industry.

The co-occurrence strategy which assigned companies to industry sectors when they appeared in the same sentence was clearly the superior strategy. The number of companies assigned was greater than the headline strategy and it was more accurate. This strategy was used to assign the companies to the industry sector which was represented as an ontology.

3.2.3 Ontology Refinements

The industry sector member companies can be known by a variety of names, for example: IBM may be known as International Business Machines, IBM or "Big Blue". A standardization of the names was required because not all variants of the company may be contained in all relevant industry sectors because the ontology would become too large to manage with the available software tools. The name standardization was achieved through information in the Open Calais meta-data. The Open Calais web service affixes a Uniform Resource Identifier (URI) to each named entity it detects. The URI points to a "landing page". The landing page has a standard name for the named entity as well as a further URI which points to a "linked data page" which contains further information. The first step in the standardization step was to assign the same standard names to companies which pointed to the same "landing page". The second step was to use the "SameAs Relation" in the linked data page. The "SameAs" relation pointed to companies which were the same as the company described in the linked data page. The "SameAs" relations typically pointed to acquisitions or subsidiaries

of a specific company.

In common with company names, industry sector names may have many variants, for example *the car manufacturing business* may be referred to as the *the automotive construction sector* or *the car construction business*. The standardization removed the "determiner" (typically "the") and the last word which would be an entry from the industry synonym list, consequently *The Finance Industry* and *The Finance Business* was represented as a single entry in the ontology with the name "Finance". The company members of each industry sector were added to the single entry in the ontology. Tables 3.4 and 3.5 provide examples of 1. industry sectors and their members and 2. companies and their encapsulating industry sectors from the industry ontology.

Sector	Members
Web	AOL ,CFC , Ebay, Google, Heroku, Microsoft Nature, News Corp, Shopify, Sun, Times, Virgin Warner, Yahoo, iPharro, The Times
Utilities	AXON, American Electric Power ,BTS Group AB, Centrica, Drax, ETFs, Electric, Energy, HCL, IDC, LNG, Severn Trent, There
Video-Conferencing	Cisco, Bell Laboratories
Television	ABC, CNN, CSI, Carlton, Dish Network, Ensequence, FA, General Electric, Lifetime, News Corp, Next, Nielsen, Nintendo, One, Operating, PRS, SONY

Table 3.4: Examples of sectors and their members.

This method produced a detailed ontology which grouped together related companies under a common industry. The grouping of companies revealed expected areas of well known companies, for example, Google and search as well as some unexpected areas, for example, Microsoft and cars, which on further investigation revealed a direct link between the company and sector[2].

3.3 Company Specific Ontology

The second strategy for constructing a lexical resource for query expansion is to construct a company specific ontology. The *company specific ontology* is a strategy which constructs an ontology which models the company's system on a day to day basis. A company's system represents entities which have an indirect or direct connection to a company. These items may include: people, companies, products, etc. The ontology changes to reflect changes in

[2]Microsoft produces software for cars,http://goo.gl/sn9Ye

Microsoft	advertising, auto, automobile, automotive, banking, chip, communications, computer,computing, consumer, content, design, device, digital, electronics, energy, entertainment, equipment, game, games, gaming, global, handset, hospitality, hosting, infrastructure, insurance, internet, mainframe, media, mobile, networking, porn,search, securities, semiconductor, server, services, smartphone, software, software and internet, tech, technology, telecom, telecoms,web, wireless
Apple	advertising, book, car, cellphone, chip, communications, computer, computing, consumer, content, education, electronics, energy, entertainment, games, global, graphics, handset, hardware, hosting, laptop, media, mobile, music, netbook, phone, publishing, radio, record, recording, smartphone, software, tech, technology, telecommunications, telecoms, video, wireless
IBM	airline, analytics, automotive, banking, chip, computer, consulting, content, database, hardware,internet, laptop, mainframe, media, networking, pharmaceutical, semiconductor, server, service, services, software, storage, tech, technology, telecom, telecoms
Google	advertising, book, computer, consumer, content, digital, display, electronics, handset, hardware, internet, media, media and telecom, mobile, music, networking, news, newspaper, online, phone, publishing, search, services, smartphone, software, software and internet, tech, technology, telecom, telecommunications, telephone, translation, video, web

Table 3.5: Examples of companies and their industry sector memberships in alphabetical order.

the company's system such as management succession. The ontology was used to score and rank news stories for a chosen company.

The *company specific ontology* strategy is a two part process which has: a construction phase and a maintenance phase. The construction step is an iterative process which initially selects a small highly relevant set of news stories with "headline keywords" (name of company). Entities and relations were extracted from the news story text. The extracted entities and relations were used to construct a base ontology. The keywords were expanded with entities from the base ontology. The new keywords were used to identify new stories where entities and relations are extracted to enrich the ontology. The process was repeated until there was no more news stories to process. The process produced a detailed company specific ontology. The maintenance phase is predicated upon the redundancy of "facts" in news stories. The news publication process ensures that facts are repeated in multiple information sources and at regular intervals over time. The maintenance process assigns a value to a relation and

reduces the value until a predetermined value when the relation is deleted. The relation's value is refreshed when its assertion is repeated in a news story. An entity is deleted when it has no relations. "Valid facts" are retained because they are repeated regularly in news stories. Errors or outdated information is expelled because the error or outdated information is not repeated in additional news stories.

3.3.1 Ontology Construction

The ontology construction strategy relies upon entities and their relations being explicitly stated in news text. The detection of entities and their relations was with the Open Calais Web Service. News text was sent to Open Calais which returned the news story's meta-data in Resource Description Framework (RDF) language. The meta-data contained: named entities, named entity's relations with other named entities and a news story categorization. A named entity is a section of text which has been classified into a pre-existing category such as company or person. A named entity is synonymous with an entity in an ontology. The meta-data provided the basics of an ontology: named-entities (entities), relations (relations) and named entity category (concepts). The meta-data provided a Uniform Resource Identifier (URI) for each named entity. The URI acts as an unique identifier for each named entity and on occasion acted as a link to external information. This external information is known as "linked data". This external information was also represented in RDF and contained the units to construct an ontology.

The company specific ontology was constructed from information in news stories and its associated linked data. The Open Calais' named entities and their categories were directly used to represent entities and their concepts, but the Open Calais relations required processing. The relations in the meta-data connected entities with more than one entity. The relation extraction split the relations into separate binary relations. A binary relation connects one entity with another entity. A binary relation is invertible which allows the inference of new relations. For example, the relation, *Bill Gates worked for Microsoft* can be inverted to read *Microsoft employed Bill Gates*.

The relation description supplied by Open Calais was "verbose" and inconsistent. A name normalization process was used to provide a descriptive and consistent relation name. The name normalization process relied upon a normalization table which held pairs of concepts (class names) and their relation name. The relation names table was hand constructed and represented all possible concept combinations available in the Open Calais meta-data. The normalization table was used to return the normalized relation name. Other tables held pairs of incompatible concepts, which identified incorrect relations in the Open Calais meta-data.

A simple ontology could now be extracted from a single news story and therefore a large number of news stories could be used to create a detailed ontology. The company specific strategy

required the construction of an ontology which represented a single company, consequently only "relevant" news stories to the chosen company could be used as ontology building blocks.

The process of constructing the ontology was in two steps. The first step had the goal of constructing a base ontology using news stories that we were highly confident to be related to the target company. This base ontology was then to be improved in the second step of the process.

News stories which contained the company name were selected to build the base ontology because there is a known relationship between headlines and news text [Andrew, 2007]. An additional set of similar stories were added. The criteria for a story to be added were: 1. linked by a hyperlink in the story's html to one or more of the stories initially selected and 2. parent html element of the hyperlink had a "linguistic clue" in its text property. The linguistic cues used in this process are in Table 3.6. The relevance of the linking provided by the cues was confirmed by a manual inspection. In summary, an initial set of seed stories was build by checking the presence of the company in the news headlines and by also selecting stories referred as related to these by means of linguist cues. Relations and entities were extracted from this seed set of stories to construct the base ontology.

Linguistic Cues
Related Stories
Related Articles
More on this Story
We Recommend
From around the web

Table 3.6: Examples of linguistic cues.

The next step of the ontology construction is strongly based on a notion of similarity between other potentially useful news stories and the base ontology constructed in the first step. This notion of similarity was defined as the intersection of the entities in a news story and the entities in the base ontology, as defined in Eq. 3.1,

$$Sim(N, O) = \frac{|Ent(O) \cap Ent(N)|}{|Ent(N)|} \tag{3.1}$$

where $Ent(N)$ is the set of entities in the news story N, and $Ent(O)$ is the set of entities in the base ontology O.

This notion of similarity was used to improve the base ontology as follows. For all remaining stories not belonging to the initial set used to obtain the base ontology, if their similarity score was higher than a threshold their relations and entities were extracted and added to the ontology. This process of improving the base ontology is detailed in Algorithm 5. This

algorithm includes as input parameter the similarity threshold. In this study the value of this threshold was selected for each company using the following heuristic process. The remaining stories mentioned above were split into 5 different subsets according to their similarity score. Each subset contained stories whose similarity score was in one of the following intervals: $[0, 0.2[$, $[0.2, 0.4[$, $[0.4, 0.6[$, $[0.6, 0.8[$, $[0.8, 1]$. From each of these subsets of stories a random sample of stories was selected. The resulting set of stories contained stories with a reasonable diversity of similarity scores. These stories were ordered by similarity score and we have manually determined the value of the threshold as the score above which all stories were related to the target company.

Algorithm 5: Ontology Enrichment.

Input: candStories - set of stories not used for building the base ontology
Input: Input: O - the base ontology
Input: Input: simThr - the similarity threshold
forall the ($sincandStories$) **do**
  if ($sim(s, O) > simThr$) **then**
    $O < -AddConceptsToOntology(O, s)$
return (O)

3.3.2 General Ontology Adaptation Strategy

A baseline evaluation was made with ontologies built with the ontology construction strategy. The ontology construction experiments had been conducted with news stories crawled from January 2009 to August 2010. The baseline evaluation was conducted on data which was not used in the construction experiments to ensure that the proposed ontology construction methodology was not biased to the data in the initial experiments. At the time the evaluation was conducted there were 53 days of news story data which had not been used in the initial construction experiments. The target company selected for the evaluation was Microsoft because: 1. it is a large company which often features in the news and 2. the company is well known. A series of ontologies were generated for "Microsoft", one for each day. The generation of an ontology was from news stories available before the ontology was generated, i.e. an ontology generated on day 1 would have access to one day of news stories and an ontology generated on day 53 would have access to 53 days of news stories.

An ontology was chosen at random (day 18) for evaluation. The evaluation of the ontology was with Vrandecic's [Staab and Studer, 2004a] tests which were: Accuracy, Clarity, Completeness, Computational efficiency, Conciseness, Consistency. The evaluation was conducted manually by a domain expert. The ontology was too large to evaluate in its entirety, consequently the entity with the most relations was chosen for the validation tasks. The evaluation methodology obliged the evaluator to positively validate errors, consequently a

Test Name	Evaluator Comments
Accuracy	The entity with the most relations was the Microsoft entity. The Microsoft entity contained 436 relations. 61 relations were positively verified as incorrect
Clarity	The number of assertions were large, in-excess of 11,000. There were 10 class types and 20 relation types. Each class and relation type conformed to Vrandecic's [Staab and Studer, 2004a] tests of understandability and transparency.
Completeness	The Microsoft entity's relations were verified for completeness. There were a number of unverifiable relations, for example *Microsoft employs the Governor of California*, however the verifiable relations provided a rich description of the domain. The assertions exceeded the experts knowledge and ability to locate all valid relations.
Conciseness	The ontology contained a number of irrelevant axioms, for example *Microsoft employs current CEO*
Consistency	There were a number of axioms which were inconsistent, for example the ontology asserted that both Microsoft and Adobe produced Acrobat Reader.

Table 3.7: Baseline ontology evaluation comments.

relation was assumed to be "correct" if it could not be verified as incorrect. There was limited time to verify relations and consequently it was necessary to restrict the effort spent on verification. A positive error identification ensured that the manual verification task could be completed in the allocated time. The annotators comments are in Table 3.7.

The initial ontology failed three of Vrandecic's [Staab and Studer, 2004a] validation tasks: conciseness, consistency and accuracy. It may have been possible that the selected ontology was atypical, and therefore an ontology which was generated at day 1 with 1 day's worth of stories was compared with an ontology generated at day 53 with 53 day's worth of news stories. The evaluator reported that the day 53 ontology had more assertions, but also more errors than the one generated on day 1. The apparent increase in errors over time necessitated the use of an error correcting strategy.

Correction Strategy

The related work section described some error correction strategies which relied upon finding logical inconstancies or measuring volatility of an ontology against a gold standard. These approaches often rely upon manual tuning or construction of a gold standard. These approaches are not possible in this domain because of the lack of: 1. a comprehensive gold

standard and 2. an authoritative logical model of relations found in news. The proposed error correction strategy seeks to avoid the restrictions of the strategies described in the related work. The proposed method was predicated upon "fact redundancy" in news stories, i.e. facts which were true were repeated in multiple locations and at regular intervals. The redundancy of information is created by the dynamics of news publication. The initial story is published by an agency [Bell, 1991](Reuters, AP, UPI, AFP) and then is repeated by mass media news publishers. An example of this phenomenon was a story which concerned the appointment of a Microsoft executive to Nokia. It was repeated in the corpus by several information sources with slight differences in the headlines. The "central fact" of the story was that *Stephen Elop is the new CEO of Nokia.* This "fact" was repeated multiple times in each story and separately by multiple information sources [BBC, 2010][Yahoo, 2010]. The proposed strategy assigns a numerical value (weight) to a relation. The relation weight is reduced until a pre-determined value when the relation is removed. The relation weight is reinitialized when the relation is asserted in a news story. An entity is removed when it has no relations. The following paragraph will describe the intuitions and definitions which will be used in the strategy description.

The weight of a relation r is a time-dependent property which is updated according to the following criteria:

- The maximum weight is a user-defined constant that we will denote as $W_0(r)$
- The weight of the relation should be maintained if the relation was re-asserted in todays' news stories
- If the relation was not re-asserted its weight should be decreased at some rate to be defined below
- If the updated weight of the relation reaches a certain user-defined threshold, the relation should be deleted from the ontology.

A number of concepts will be defined to assist the process of implementing these intuitions. Let $S_t^w(r)$ be the set of new stories appearing in the time interval $[t-w+1, t]$ that assert the relation r. In this context, $S_t^1(r)$ will be the set of stories in day t. Let G be a user-defined mutually exclusive grouping (partitioning) of the set of relations R of an ontology O. This grouping should be formed with the goal of expressing the domain knowledge that relations on the same group are expected to have a similar assertion frequency. The grouping is necessary because it is expected that not all relations in an ontology have the same frequency of assertion. Let g_r be the group to which r belongs, i.e. $g_r = g \in G : r \in g$.

The updated weight of a relation r, $W_{t+1}^w(r)$, is given by the following equation,

$$W_{t+1}^w(r) = \begin{cases} W_0(r) & \text{if } |S_t^1(r)| > 0 \\ W_t^w(r) \times (1 - DR_t^w(r)) & \text{otherwise} \end{cases} \tag{3.2}$$

where w is a time window size; $S_t^1(r)$ is the set of news appearing today that assert relation r; and $DR_t^w(r)$ is the decay rate of relation r to be defined below.

As previously mentioned , after a new weight is calculated using the above equation, the system decides whether to maintain or not the relation in the ontology based on a user-defined minimum weight threshold.

The definition of the weight updating rule show in Eq. 3.2 means that every time a relation is re-asserted its weight goes back to its default user-defined value, $W_0(r)$. Otherwise, its weight will be decreased at some rate that varies with time.

The weight decay rate was designed to implement the following intuitive ideas:

- Different groups (types) of relations have different expected assertion frequencies, which also vary with the company (ontology) being monitored
- Relations whose assertion frequency is lower than what is expected should see their decay rate increase, while the opposite should also occur.

The key factors behind these ideas are: 1. how to calculate the assertion frequency of r; and 2. how to calculate the expected frequency of assertion of r. Given the dynamic nature of news, these notions are defined as a function of the time window w.

The assertion frequency of a relation r is the number of stories asserting r per day, calculated on a past window w, i.e.

$$AF_t^w(r) = \frac{|S_t^w(r)|}{w} \tag{3.3}$$

As mentioned before, the relations in an ontology were grouped into a set of sub-sets, G. The motivation is the empirical observation that not all types of relations are expected to be asserted with the same frequency. In this context, the expected assertion frequency of a relation r will be defined as a function of the assertion frequencies of the members of the group of r, i.e. g_r. Namely, we are searching for relations whose assertion frequency is much lower than expected, which provides a clear indication that they may be errors and thus should be deleted from the ontology. In this context, we are looking for unusually low values of the AF score with respect to the relation group. From a statistical point of view we are looking for extreme low outliers. A frequent and robust method to determine if a value is an outlier with respect to a given sample of values is the *box plot rule*. This non-parametric rule says that a value is an outlier if it is outside of the interval $[Q_1(x) - 1.5 \times IQR(x), Q_3(x) + 1.5 \times IQR(x)$, where $Q_1(Q_3)$ is the 1st (3rd) quartile of the sample and $IQR(x)$ is the respective interquartile range $(= Q_3(x) - Q_1(x))$. In this case, we are only interested in AF values unusually low, i.e. below the first limit of the above interval. Using this statistical notion we can state that if a relation has a value of AF that is unusually low with respect to the AF scores of its group, then its weight decay rate should be set to ensure that the relation is deleted from

the ontology, i.e. $DR_t^w(r) = 1$. For all remaining relations, their weight decay rate will be set as a function of their position regards the distribution of AF scores in their respective groups. The motivation is that the higher the AF score the lower should be the decay rate, so that the weight of the relation is not decreased too much.

Let $\mathcal{AF}$ be the set of AF scores of the relations in the same group as r, i.e. $\{AF_t^w(s) : s \in g_r\}$. Let $k = Q_1(\mathcal{AF}) - 1.5 \times IQR(\mathcal{AF})$ be the threshold below which an AF score is considered unusually low, in accordance to the *box plot rule*. The weight decay rate of a relation $DR_t^w(r)$ is defined as,

$$DR_t^w(r) = \begin{cases} 1 & \text{if } AF_t^w(r) < k \\ 0.5 + \frac{0.5}{mx-k} \times (k - AF_t^w(r)) & \text{otherwise} \end{cases} \tag{3.4}$$

where $mx = \arg\max_{s \in g_r} AF_t^w(s)$ is the maximum assertion frequency of the relations in the same group as r.

This formulation gives a decay rate of 1 to the unusually low AF scores and a rate decreasing linearly in the interval $[0.5, 0]$ for the relations with AF scores above the threshold k. Figure 3.1 exemplifies the behaviour of the proposed formulation.

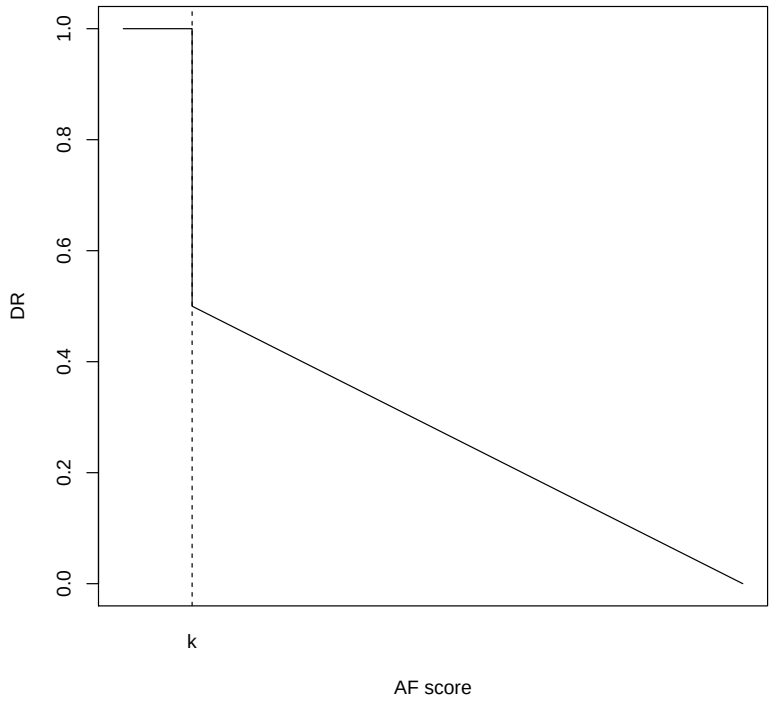

Figure 3.1: The proposed weight decay rate function.

In summary, the proposed relation weight updating strategy can be used to adapt the

ontology dynamically by deleting relations which are deemed to be invalid. The proposed strategy consists simply starting from the base ontology, with the relation weights set to their default user-defined values, and then at the end of each new day updating the weights using Eq. 3.2. If the new weight is below a user-defined value the relation is deleted from the ontology, thus dynamically correcting errors.

3.3.3 Evaluation of General Ontology Adaptation Strategy

The evaluation of the proposed error correction strategy was by manual evaluation against a baseline which was no error correction strategy. The evaluation set-up was the same as described on page 54. Two sets of ontologies were maintained over a 53 day period. One set of ontologies used the proposed correction strategy whereas the other set did not use any form of error correction. The ontologies used news up to the day they were created, i.e. an ontology generated on day 1 had access to one day of news whilst an ontology generated on day 53 had access to 53 days of news. The evaluation process chose 3 days at random, as well as the ontologies which were generated on days 1 and 53. The evaluation was limited to ontologies generated on 5 selected days because manual evaluation is an expensive process, and consequently it was necessary to limit the number of ontologies evaluated so that the evaluation could be achieved in a reasonable period of time. The selected days were days: 1, 4, 19, 35 and 53. The rules for the manual inspection were the same as described on page 54, i.e. 1. the entity with most relations was evaluated, 2. errors had to be positively identified and 3. relations which were not positively validated as false were assumed to be correct. The number of errors and relations for the entity with most relations was recorded.

The results for the number of errors evaluation is presented in Figure 3.2. The number of errors in the uncorrected ontology increases when the ontology construction process had access to an increasing number of news stories. For example, on day 1 the entity which has most relations had 18 erroneous relations, and on day 53 it has 63 erroneous relations. The ontologies which used the correction scheme did not have the same increase in the number of erroneous relations. On day 1 the ontology which used the correction scheme the entity which has most relations had 15 erroneous relations, and on day 53 it had 12 erroneous relations. There was a decrease in the number of errors as the ontology construction process had access to more news stories.

It is possible that the decrease in erroneous relations could be accounted for by a general reduction in all relations. If this was the case then the error rate for both relations should be similar. The error rates for both sets of ontologies is in Figure 3.3. The error rate demonstrates that the ontologies which had the correction strategy applied to them had a lower error rate than the ontologies which did not. It is reasonable to assume that if the error correcting strategy deleted correct relations then it deleted them at a lower rate then it deleted erroneous relations.

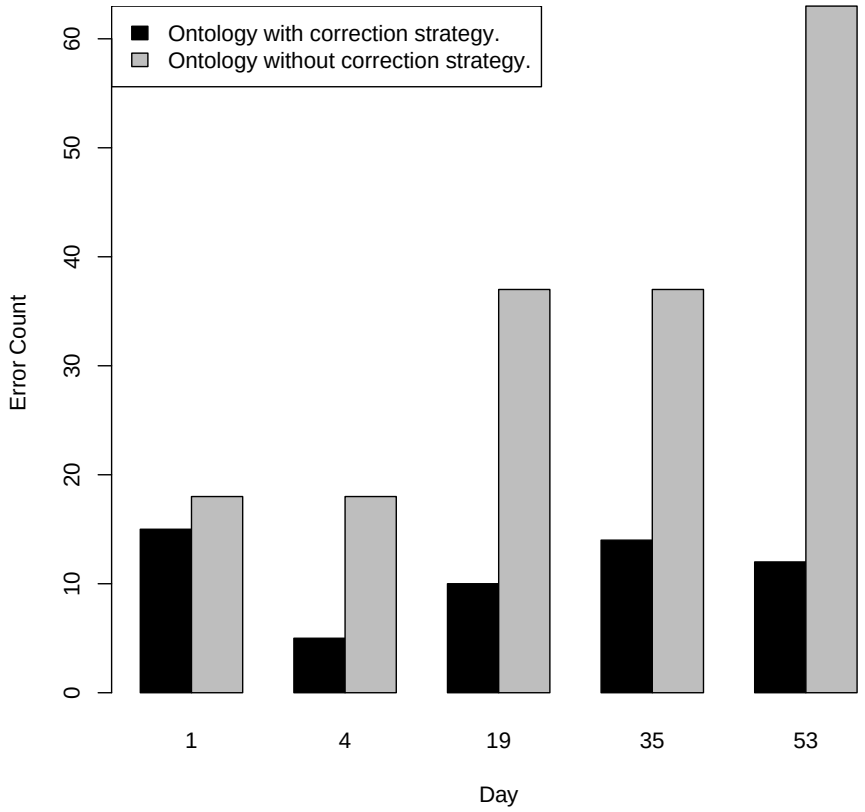

Figure 3.2: umber of errors for entity with most relations for: 1. ontologies with correction scheme and 2. ontologies without correction scheme.

The growth of the number of relations for the entity with the most relations is demonstrated in Figure 3.4. The ontologies which were not subject to the error correction strategy had more relations than the the ontologies which were subject to the error correction scheme. The difference may be explained by two factors: 1. higher number of errors and 2. higher number of unverified relations.

In summary the evaluation criteria underplayed the effectiveness of the proposed methodology because: 1. errors had to be “positively” identified, 2. the "quality" of the relations was not considered. There was a larger number of “unverifiable relations” in the uncorrected ontology than the ontology which had been corrected by the proposed strategy. The “unverifiable relations” were not “obviously wrong”, however it was not clear if they were part of the domain. The “unverifiable relations” accounted for the significant part of the difference in total relations between the uncorrected and corrected ontologies.

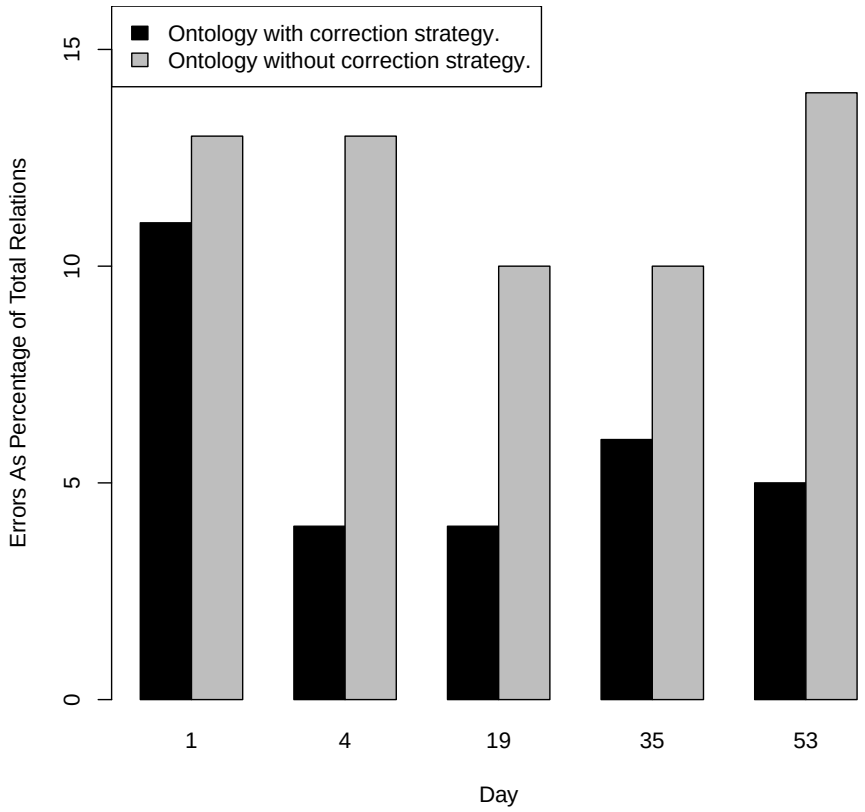

Figure 3.3: Error rate for entity with most relations.for: 1. ontologies with correction scheme and 2. ontologies without correction scheme.

The evaluation criteria did not consider the ability of the ontology which had been corrected to resolve ambiguous relations. For example at day 19 both Ontologies asserted that Microsoft and Adobe produced Acrobat Reader. In the ontology which had been corrected there was a measure of "confidence" for both relations in a form of a "relation weight". The "relation weight" for the assertion "Adobe produces Acrobat Reader" was higher than the weight for the assertion "Microsoft produces Acrobat Reader". It was possible to identify the "true" relation by accepting the relation with the highest weight.

It was not possible to evaluate the "completeness" of the ontology as it was not feasible to generate a "Gold Standard" of all of the relations and entities in the domain, however the Modified Ontology contained a number of stable relations which described the commonly known "facts" of the domain. The stable relations did not exit the ontology at the points the intermediate ontologies were manually evaluated. Although this was a limited empirical validation of the proposed strategy, the evidence provides a good indication of the effectiveness

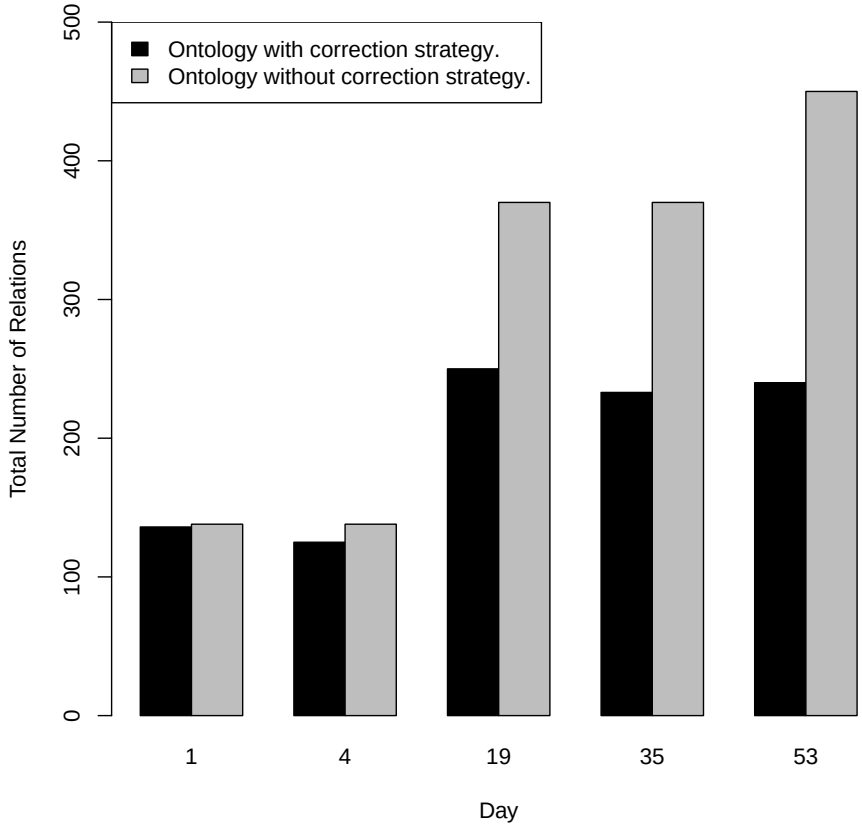

Figure 3.4: Number of relations for entity with most relations for: 1. ontologies with correction scheme and 2. ontologies without correction scheme.

of the proposed error control strategy.

3.4 News Retrieval using Ontologies

The aim of the construction of both the company specific and the industry ontologies was to use the information contained in the ontologies to retrieve relevant news stories. This section will describe two alternative ways of retrieving these stories: one based on industry ontologies and the other using the company specific ontologies. We will describe methods for using the information contained on these ontologies to calculate a score for each candidate news story. These scores will be used to select the relevant news stories. Finally, we will compare these two alternative proposals for news retrieval based on ontologies.

3.4.1 News Retrieval using the Industry Ontology

Given the information contained on the industry ontology it is proposed to calculate a relevance score for each news story. This score will be obtained by counting how many of the entities appearing on the story are relevant for the company we are interested in. In this context, it is first defined the set of entities in an industry ontology that are relevant for a given company C, which we will denote as $Ent_C(O)$, where O is an industry ontology. This set is formed by: 1. entities which share the same industry sector as C, 2. entities which are encapsulating the industry sector of C and 3. the entity representing the company C. The set of entities that are mentioned in a news story d are denoted as $ent(d)$ and the relevance score of the news story is defined as the cardinality of the intersection of these two sets of entities,

$$rel_{IO}(d, C) = \mid ent(d) \cap Ent_C(O) \mid \tag{3.5}$$

In summary, given the set of stories D_t appearing on day t our proposed retrieval method consists of selecting the ones with a relevance score greater than zero, i.e. $\{d \in D_t : rel_{IO}(d, C) > 0\}$.

3.4.2 News Retrieval using Company Ontologies

The second alternative method we are proposing will use the information of the company specific ontology to retrieve the news stories that are relevant for the company. Again, we are proposing to qualify the candidate news by a relevance score that will be used to select the ones we judge relevant for the company. However, given the different nature of the company specific ontology, the way we obtain this score is different. Given the set of entities appearing in a news story, $ent(d)$, two aspects of these entities will be considered to calculate the news story relevance score: 1. the importance of the entity within the company specific ontology and 2. the strength of the connection between each entity and the target company C. The intuition is that we want to penalize both entities that are not very strongly connected to the company (though being a part of its ontology), and also entities that though sufficiently connected are not very relevant within the ontology.

To calculate the importance of an entity within the ontology the notion of centrality from social networks has been used. Centrality measures calculate an *importance* of an entity by calculating the number of relations it has with other entities[Liu, 2007d]. Given the temporal nature of the company specific ontology we have used the notion of assertion frequency to qualify each relation of an entity and thus obtaining the centrality score of an entity as described in Equation 3.6.

$$EC_t(e) = \sum_{r \in R(e)} AF_t^w(r) \tag{3.6}$$

where $R(e)$ is the set of relations in the ontology involving entity e and $AF_t^w(r)$ is the assertion frequency of a relation (c.f. Equation 3.3 on page 57).

The strength of the connection between an entity e and the target company C is going to be estimated by means of a Proximity Prestige (PP) score. This score will be calculated as a function of the weights of the relations in the shortest path from e to the company entity. The shortest path between these two entities was calculated with the Floyd's Algorithm [Floyd, 1962]. Note that this algorithm can return more than one path in case several paths have the shortest length. The Proximity Prestige (PP) of entity e with respect to the company entity e_C is given by the following equation,

$$PP_t(e, e_C) = \underset{p \in SP(e,e_C)}{\arg\max} \frac{1}{|p|} \sum_{r \in p} W_t(r) \tag{3.7}$$

where p is a path formed by a set of relations, $SP(e, e_C)$ is the set of shortest paths from entity e to e_C and $W_t(r)$ is the relation weight (c.f. Equation 3.2 on page 56) at time t.

Having described these two notions we can now define the relevance score of a news story d using the information in a company specific ontology as,

$$rel_{CO}(d, C) = \sum_{e \in ent(d)} PP_t(e, e_C) \times \frac{EC_t(e)}{\arg\max_{s \in ent(O)} EC_t(s)} \tag{3.8}$$

where O is the company specific ontology of C, and $\arg\max_{s \in ent(O)} EC_t(s)$ is the maximum centrality score of the set of entities in the ontology ($ent(O)$).

As with the previous relevance score we will retrieve the news stories from D_t that have $rel_{CO}(d, C)$ greater than zero, i.e. $\{d \in D_t : rel_{CO}(d, C) > 0\}$.

3.4.3 Comparative Evaluation of Ontology Information Retrieval Strategies

This subsection will describe an evaluation of a news retrieval system which uses either the: 1. Industry Ontology or 2. Company Specific Ontology which was corrected with the proposed adaptation scheme to score news stories. The strategies will be compared against two baselines: 1. Lucene [Gospodnetic and McCandless, 2009] which uses the name of a monitored company to retrieve news stories and 2. a Company Specific Ontology which was not corrected. Lucene is a document based inverted index, and in the experiments each individual news story was assigned to a single document. Lucene ranks a news story

(document) by applying a score to a document. Lucene applies a score to a document in response to a query. The technical documentation for Lucene is not clear, but it states that a document score is calculated with a combination of boolean logic and the Vector Space Model (VSM). The boolean part of the query excludes documents which do not contain any of the query terms, and the VSM computes a document score for the document[3]. The query issued in this experiment was the date of the required news experiment with a start and end date time, e.g 00.00.01 - 23.59.59, and the name of the target company, e.g. "Microsoft". The document indexing and scoring were the default settings. The stories were ranked in descending order of document score. The Company Specific Ontology which was not corrected ranks news stories in the same manner as the corrected Company Specific Ontology.

The evaluation was a manual evaluation against a set of general guidelines for the evaluator. It was not possible to supply prescriptive guidelines because a number of the judgements would be subjective and therefore left to the discretion of the evaluator. The ranking rules are described in Tables 3.8 and 3.9.

Rank	Rule
1	Domain company name in text no other company preceding it.
2	Prominent domain company employee in text no other company or employee preceding it.
3	Domain company product in text no other company, product or employee preceding it.
4	Prominent domain company's competitor in text no other company, employee or product preceding it.
5	Prominent domain company competitor's employee story in text and no other company, employee or product preceding it.
6	Prominent domain company competitor's product story in text and no other company, employee or product preceding it.
7	Domain company general market area, i.e. if company makes phones then a general story about telecoms is acceptable.

Table 3.8: Ranking guidance for manual evaluator.

The evaluation used news stories published between the 1st September 2010 - 1st January 2011. Earlier news stories were not used because they had been used in the construction experiments for the Company Specific Ontology. The Industry Ontology was generated from news stories published between October 2008 - August 31st 2010. The Industry Ontology was designed to be static and therefore did not use news stories from the evaluation period in its construction.

[3]More information about document scoring can be found at the Lucene website at: `http://lucene.apache.org/core/3_6_0/scoring.html`

Rule	Reason
Macro Economic News	This news applies to all companies and should not have a high rank.
News which refers to markets which the company does not operate in.	This news is unrelated to domain
News which has preceding mentions of unrelated companies, products before any relevant terms (described in Ranking Rules Table)	In general stories which mention large amounts of companies are typically stock recommendation stories and have no relevance to the domain.

Table 3.9: Exclusion guidance for manual evaluator.

The Company Specific Ontology was generated on a daily basis. It used news stories from the beginning of the evaluation period to the previous day an ontology was generated, i.e. an ontology which was generated for 2nd September 2010 would have access only to news stories published on 1st September 2010 whereas an ontology generated for 30th September would have access to news stories published between 1st September 2010 - 29th September 2010. This process ensured that there was no company specific ontology for September 1st 2010 and consequently September 1st was excluded from the evaluation process because at least two of the strategies would not be able to score news stories for that day.

The experiments were conducted in January 2011. The evaluation used Microsoft as the target company because: 1. there are a significant number of news stories published for Microsoft and 2. the domain is well known. In the evaluation each strategy would score all news stories available in the news story corpus on daily basis. News stories which scored 0 were excluded. The remaining news stories were ranked in descending order of their assigned score on daily basis. The evaluation therefore had for each day in the evaluation period: 1. list of all news stories published and 2. four individual ranked lists (one for each competing strategy) of news stories sorted in descending order of news story scores.

The evaluation followed a suggested test by Manning and Schütze [1999] which appraised the accuracies of each system at 1-5, 1-10 and 1-20 document ranges. A document range in this case is a set of documents with a specific rank, i.e. an evaluation range 1-5 would evaluate news stories which had been assigned a rank between 1 and 5 by a specific strategy. A random selection range of 5 news stories with a rank greater than 20 was selected for evaluation. The rational to this evaluation was to ensure that there was an evaluation measure for news stories with a rank greater than 20. In addition a count of the number of stories each strategy selected was recorded.

Manuel evaluation was a labour intensive process and consequently it was not possible to evaluate all of the days in the evaluation period. A large random sample was taken. The

random sample encompassed 25 days of news stories. The results were an average of the individual evaluations for each of the 25 days. The results are documented in Tables 3.10 and 3.11. The accuracy figures are expressed as a percentage and the documents returned figure is a raw count.

Strategy	Accuracy(1-5 Documents)	Accuracy(1-10 Documents)	Accuracy(1-20 Documents)	Accuracy Random Selection
Industry	80%±5	90%±8	90%±7	55%±29
Company Specific	100%±0	93%±5	90%±5	73%±31
Lucene	70%±16	80%±0	N/A	N/A
Uncorrected Company Specific	80%±16	73%±22	63%±18	20%±28

Table 3.10: Evaluation results for ontology recall strategies.

Strategy	Documents Returned
Industry	72 ±45
Company Specific	65±60
Lucene	12±10
Uncorrected Company Specific	173±113

Table 3.11: Average number of documents returned.

The results demonstrate that the proposed correction strategy on the Company Specific Ontology had a direct impact on the quality of the results. The accuracy of the Company Specific Ontology when used to rank news stories was higher when the correction strategy was applied than when it wasn't. An uncorrected Company Specific Ontology when used to rank news returned many news stories, but was inaccurate.

The Company Specific Ontology based news story recall strategy arguably outperformed the Industry Ontology news story recall strategy because it made less "serious" mistakes. The Industry Ontology news story recall strategy often made mistakes with stories ranked with a position of 5 or higher. In addition, the Company Specific Ontology based news story recall strategy accuracy declined slower with the random selection of news stories than the Industry Ontology strategy. The Lucene baseline was less accurate than the competing strategies at rank 1-5 and had a higher accuracy than the uncorrected Company Specific Ontology news recall strategy at rank: 1-10. Lucene did not return enough news stories for either: the random selection and the 1-20 document selection evaluation to be attempted. Company Specific Ontology news recall strategy was unique because it had a high accuracy measure and a relatively high average number of returned documents.

3.5 Summary

This chapter presented strategies for identifying related news to a target company. The strategies relied upon ontologies to represent relationships between target companies and other entities such as industry sectors or products. The strategies necessitated advances in two areas: 1. ontology construction and maintenance, and 2. news retrieval with ontologies.

3.5.1 Ontology construction and maintenance

There were two separate ontology construction strategies presented in this chapter. The first strategy (industry ontology) used patterns to identify industry sectors. Industry sectors are entities which represent groups of companies. The construction process used news stories to build the ontology. The news stories were divided into sentences and the meta-entities and companies were located in each sentence. The identified companies were assigned to a meta entity by their co-occurrence in the same sentence as a specific meta-entity. The final industry ontology contained industry sectors and their company members. A company could be a member of 1 or more industry sector if they conduced business in more than one area.

The second strategy (company ontology) attempted to represent the *domain* of a company. The domain of a company can include a larger variety of entities than the industry ontology because the proposed strategy attempted to provide more detail than the industry ontology. The company ontology was constructed from the Open Calais meta data which was added to each news story as well from external resources the Open Calais meta data provided links to. The construction methodology used a recursive strategy to initially build an ontology from *directly relevant* news stories which contained the target company's name in the headline. The ontology was used to select related stories which did not contain the company's name in the headline to enrich to the ontology.

The increased detail presented problems because these details changed overtime, for example employees changing employer or companies launching new products. The ontology had to be adapted over time to match the changes in the company's domain. There was no gold standard against which to evaluate the ontology, consequently a decay system based upon characteristic's of news where facts are repeated over time was developed. This decay scheme assigned an initial "weight" to a relation of an entity. The weight was reduced until a pre-defined value where it was removed. An entity with no relations was deleted. The relation weight was reinitialized if the relation was reasserted in a news story. In addition a relation weight's decay was influenced by the "domain volatility". A domain where there was frequent publication of news stories would increase a relation's decay rate, but a domain where news stories were published infrequently reduced a relation's decay rate.

The ontology construction methods were advances to the field of ontology engineering because

there was no similar work published at the time the work was constructed. The concept of building time dependent ontologies from volatile resources such as news stories also seems to be an original concept. Time dependent ontologies and their management are not new concepts because there are many published approaches, but these approaches rely upon either manual management or a *ground truth* against which to evaluate an ontology at a given point in time. There was a lack of published work where there was no ground truth or resources to do active management. The decay scheme seems to be an unique contribution to the field of ontology engineering.

3.5.2 News retrieval with ontologies

The notion of using ontologies to select news is not a new concept, but the manner in which the ontologies were used to select news is a contribution to the research literature. The notion of using industry sectors and their members to recommend news was not identified during the research literature search. This approach allowed the identification of news which was directly relevant to a target company.

The concept of using social network measures in an ontology to rank and score news stories is a novel contribution. The use of social network measures allows the use of the structure of the ontology to rank the entities contained within it. These entities are then used to score a news story. This approach differs to to traditional information retrieval strategies which use ontologies to expand queries, but rely upon a separate indexing engine to rank the documents. The addition of a ontology correction strategy may change the structure of the ontology and therefore the information retrieval may alter over time, i.e. a series of stories scored and ranked by ontology created on day x may have a different rank or score when evaluated by a separate ontology on day y.

In summary, this chapter has presented two novel strategies for ontology construction and maintenance, and ontology based news recall. The strategies allowed for an accurate identification of related news to a target company which outperformed two competing strategies.

Chapter 4

Text Analysis

The hypothesis of this thesis is news can contain information which is not reflected in the price of a share or market index at the time of the news story's publication [Mitra and Mitra, 2011a]. This information can allow a trade to be made where the risk is disproportionately lower than the potential returns. This type of information is known as "alpha". News can be a common source of "alpha" because it contains timely information [Mitra and Mitra, 2011a] which describes either: the immediate future or present whereas numerical data is typically concerned with the past [Kloptchenko, Eklund, Back, and Karlsson, 2004].

The previous chapter described strategies for collecting relevant news stories for specific companies. This chapter will describe strategies for analysing the news story text returned by an information retrieval strategy. The analysis strategies calculate a value for the text which reflects the information it contains. The aggregated value of all the stories analysed on a specific day is passed onto a stock trading system.

This chapter presents two types of text analysis strategies: sentence level and document level. Sentence level strategies classify information at the sentence level, and therefore preserve linguistic information between words in the same sentence. The sentence level strategy and its application to classifying a day's worth of news can be summarized as the following:

- Split news stories published on a single day into sentences.
- Score each sentence.
- Aggregate the score of all sentences into a single value.

The document level strategies classify individual news stories. The strategies represent each news story as a "bag of words". A bag of words representation does not preserve linguistic relationships between words. The document level strategies and their application to classifying a day's worth of news can be summarized as the following:

- Represent each news story as a bag of words.
- Classify news story into one of three categories: neutral, positive or negative.
- Replace the categories with a value, for example 1 for positive and -1 for negative.
- Aggregate the values of the news classified on a single day.

The development of each of the strategies had three restrictions: 1. there was no resource to label a large amount of documents, 2. there was no resource available with specific knowledge about financial markets and 3. there was no readily available set of labelled documents. The implication of the restrictions was that a significant amount of time was devoted to labelling data for each strategy. The labelling strategies relied upon: 1. insights about the financial news domain, 2. insights about financial markets or 3. general linguistic characteristics of the problem domain. The lack of labelled data impacted the validation of the techniques which had to rely upon small hand labelled sets of data. The evaluation presented in this chapter is an initial evaluation which provides some justification for the proposed strategies. The strategies will be subjected to a comprehensive trading evaluation which is described in Chapter 6.

4.1 Sentence Level Strategies

The sentence level strategies were designed to detect two specific types of information in a sentence: event or sentiment information. The justification for this approach is found in the economic literature review on page 12. The economic literature review discovered previous work that suggested that financial markets reacted to event or sentiment information in news text, and therefore a text analysis strategy which identifies this information in news stories could be successful in predicting the direction of a financial market. The sentence level strategies are rule based. A rule based strategy was selected because of the lack of labelled data which would be required for a machine learning based approach. The rule based approach relies upon intuitions about the problem domain as well as linguistic knowledge of the language used in news stories.

The sentence level strategies had two steps: 1. phrase annotation (labelling) at the sentence level and 2. phrase scoring. The phrase annotation was achieved with rules written in Java Annotation Patterns Engine (JAPE) format for the GATE platform. The annotated phrases are extracted and scored by a separate process which had different methods for scoring event or sentiment phrases. An event phrase is scored by estimating the relationship between the verb and the object. The sentiment phrases are scored by using a modified AVAC algorithm [Subrahmanian and Reforgiato, 2008]. A sentence is scored by combining the scores of the annotated: 1. sentiment phrases, 2. event phrases or 3. event and sentiment phrases

it contains. The combination approach was selected upon due to an intuition about the text. The intuition suggested that sentences which contained more than one event or sentiment phrases could be broken down into smaller sentences due to the presence of conjunctions or semi-colons. For example, the hypothetical sentence, "Shares in BAE Systems, Europe's biggest defence contractor, have fallen 33 per cent over the past 12 months, although they still have outperformed the FTSE All-Share index by 5 per cent.", contains two events, which are joined by the conjunction "although". It is arguable that this sentence could be split into two sentences and therefore a strategy which did not combine the phrase scores of larger sentences would reward shorter sentences at the expense of longer sentences. In the complete system a summing strategy was used to combine all the sentence scores to produce a daily score, but in this chapter a document equivalence was calculated by summing all the sentences in a given news story and dividing it by the total number of sentences in the story. This document equivalence score was used to compare the sentence and document level strategies against a manually labelled set of stories.

The annotation step annotates phrases using extraction patterns: 1. <subject> <verb> <object> for event phrases and 2. <subject> <adjective> <object> for sentiment phrases. In both patterns, the subject was either: an economic actor or a property of an economic actor. The verb in the event pattern was a business event verb and the adjective in the sentiment pattern was a sentiment adjective. The annotation step relies upon the resources available in GATE, for example sentence splitting, part of speech (POS) tagging and named entity extraction. The resources in GATE were not sufficient, and a number of additional resources were required. The next subsection will describe the resources construction step.

4.1.1 Lexical Resource Construction

The lexical resource construction was predicated on the following definitions:

- Named Entity: a proper noun which is assigned to a pre-defined class.
- *Economic Actor*: a named entity which is assigned to one of the following pre-defined classes: 1. Company, 2. Public Organization (government, central bank, etc.), 3. Business Leader (CEO, Managing Directory,etc.), 4. Market Index (NASDAQ, FTSE, etc.) and 5. Industry Sectors
- *Property of Economic Actor*: a noun which has an affiliation with an economic actor, for example profits, costs, etc.
- *Business Event Verbs*: a verb which connects a property of an economic actor to an economic actor.
- *Sentiment Adjective*: an adjective which connects a property of an economic actor to an economic actor.

- *Negators*: an adverb which inverts the sentiment orientation of an adjective.
- *Sentiment Modifiers*: an adverb which either: intensifies or minimizes the sentiment orientation of an adjective.

In addition the lexical resource construction relied upon the notion of Pointwise Mutual Information (PMI). PMI is an association measure which was introduced into linguistics by [Church and Hanks, 1990]. In the lexical resource construction PMI is used to determine the association between words.

There are two types of lexical resources constructed for the sentence level strategy: 1. resources for the GATE platform and 2. resources for a scoring process which is external to GATE. The resources for GATE platform are either: 1. lists with an entry and a class name or 2. a Jape Rule which is a form of regular expression for named entities. The resources for the external scoring process are lists which contain an entry and a numerical value.

The resources which were constructed for the GATE platform were the following: 1. economic actor named entity lists, 2. list of business event verbs, 3. list of sentiment adjectives, 4. list of adverbs and 5. list of property of an economic actor. The resources which were constructed for the external scoring process were: 1. list of business event verbs and their associated scores, 2. list of sentiment adjectives and their associated scores, 3. list of property of an economic actor and their associated score. 4. list of adverbs and their associated scores.

GATE contains a named entity recognizer which is a series of lists with an entry and a class name, for example, the entry Microsoft has a class name "company". The lists which are used in a named entity recognition process is controlled by a single file which is a central named entity list register. The named entity lists supplied with GATE did not contain a sufficient number of economic actors (companies, organizations, etc.). Additional economic actor named entities were extracted from the Open Calais meta-data in a process described on page 47.

The lists of: 1. business event verbs, 2. sentiment adjectives and 3. adverbs for the GATE platform were constructed from news stories which were harvested between October 2008 and September 2010. The first two construction phases for each of the lexical resources for GATE are identical: split news stories into sentences, add POS and named entity information to a sentence. At this stage there is a list of sentences whose words have a POS tag (Noun, Verb, Adverb, Adjective, etc.) attached, and a named entity tag. The description of the construction method for each lexical resource will assume that these common steps have been completed, and therefore are not required to be stated explicitly.

The event extraction pattern relied upon the discovery of a group of verbs known as business event verbs. Business event verbs were a hypothesized subset of event verbs[Levin, 1993] which describe actions of an economic actor (EA). The discovery method of these verbs

assumed that these types of verbs would co-occur in the same sentence as an EA and therefore identifying verbs which have a "strong connection" with a class of economic actor would be a good indication of a business event verb. The discovery process can be summarized as the following: 1. extract verbs which co-occur in the same sentence as an EA, 2. calculate Pointwise Mutual Information score for verbs and their co-occurrence with EA categories, 3. delete verbs with PMI score of 0 or less, 4. expand verbs with synonyms from WordNet and 5. expand verbs with members of verb's Levin class from VerbNet.

The list of sentences created by the common steps were filtered for sentences which did not contain: 1. an EA or 2. a verb. The remaining sentences contained at least one verb and EA. This group of sentences will be known as the "business event group of sentences". Each sentence in 'business event group of sentences" was labelled with the category of the EA it contained, for example a sentence which contained the named entity "Microsoft" would be labelled as "company". A sentence which contained more than one class of named entity would have multiple labels. The verbs were extracted from these labelled sentences. This set of verbs will be known as the "base set". The affinity of a verb contained in the base set with a class of named entity was calculated with a Pointwise Mutual Information score (PMI). The PMI score of verb is described in the following equation,

$$PMI(Category, Verb) = \log_2 \frac{Pr(Category, Verb)}{Pr(Category)Pr(Verb)} \tag{4.1}$$

where $Category$ represents a label of a sentence in which an EA and verb co-occur, Pr represents a probability and $Verb$ represents an entry from the base set.

The $Pr(Category, Verb)$ is the probability of a verb co-occurring with a specific category label. This probability is calculated by the following equation,

$$Pr(Category, Verb) = \frac{Nu(Category, Verb)}{Nu(Verb)} \tag{4.2}$$

where $Nu(Category, Verb)$ is the frequency a verb appears in a sentence from the "business event group of sentences" which has been assigned a specific category and $Nu(Verb)$ is the frequency of the verb in all sentences in the news story corpus.

All frequencies in the previous and forthcoming equations were calculated from a set of news which was crawled from 1st October 2008 - 30th September 2010.

The probability of a verb is determined by the following,

$$Pr(Verb) = \frac{Nu(Verb)}{Nu(AllVerbs)} \tag{4.3}$$

where $Nu(Verb)$ is the frequency of a specific verb in the whole news corpus and $Nu(AllVerbs)$ is the frequency of all verbs in the whole news corpus.

The frequency of $AllVerbs$ is a count of words which comply to the requirements of one of the following POS tags: VB, VBD, VBG, VBN, VBP and VBZ. For example, if there were two verbs in the whole news story corpus, each occurring once each then $Nu(AllVerbs)$ would equal two.

The probability of a sentence category is determined by the following,

$$Pr(Category) = \frac{Nu(Category)}{Nu(Sentences)} \tag{4.4}$$

where $Nu(Category)$ is a count of the sentences which have been labelled with a specific category and $Nu(Sentences)$ is a count of the total number of sentences in the news story corpus.

Verbs which had a PMI score above 0 were assumed to have an affiliation with a category (class name) of EA. The list of verbs were relatively small, and consequently they were expanded. The expansion process used two linguistic resources: VerbNet [Karin Kipper and Palmer, 2006] and WordNet [Fellbaum, 1998]. VerbNet is a dictionary which groups verbs into Levin classes [Levin, 1993]. A Levin class is a group of verbs which share common characteristics which were determined by linguistic analysis by Beth Levin. For example, the verb *bounce* could be expanded with members of the *Roll Class* which were: *drift*, *drop*, *float*, *glide*, *move*, *roll*, *slide*, *swing*. The base set of verbs and their Levin Class members were expanded with synonyms from WordNet. This final list of expanded verbs were manually verified to ensure that there were no obvious errors. An example of the "business event verbs" can be found in Table 4.1 where an artificial verb category has been added for clarification purposes. The expanded verb lists were added as separate resources to GATE named entity recognizer.

Verb Category	Examples
Obtained	gain, add, forge, win, attract
Lost	fire, cut, cancel
Direction	climb, fall, boost, down
Behaviour	storm, unravel
Influence	hurt,hit push, suffer

Table 4.1: A Sample of business event verbs.

The identification of adjectives for use in the sentiment extraction patterns were discovered in a similar manner to the event verbs. A list of adjectives was created by extracting adjectives from sentences which contained an EA and adjective. A PMI score was calculated for each extracted adjective with an EA class. Adjectives which had a PMI greater than 0 with a specific class of EA were assumed to have an affiliation with that type of EA.

At this point the adjective discovery process diverges from the verb discovery process. The extracted adjectives were to be increased in number (expanded) because the initial extracted

adjectives were relatively few in number. The expansion process uses "adjective values" and conjunctions. Adjectives can be assigned a value which indicates the sentiment orientation of an adjective, for example a value of 1 may indicate a positive orientation whereas -1 may indicate a negative orientation. This value will be used in a linguistic resource external to GATE to score the extracted phrases, however this value can assist in an expansion of the initially selected adjectives. Hatzivassiloglou and McKeown [1997] claimed that conjunctions such as "and" and "or" can join together adjectives of similar orientation. They claimed it was possible to propagate a label or value from an adjective with a known orientation to an adjective with an unknown orientation. This property of sentiment adjectives allows the expansion of sentiment adjectives for the GATE linguistic resource as well scoring the adjectives for use in a linguistic resource external to GATE which will be used to score extracted sentiment phrases.

The first step in the expansion process was to score each of the adjectives extracted in the initial process. A value for each adjective was gathered from Sentiwordnet [Esuli and Sebastiani, 2006] which is a pre-compiled sentiment dictionary. Sentiwordnet contains an entry (a unigram) with an associated value. The list of adjectives were manually verified for errors in the assigned values. The next step was to expand the adjectives with synonyms from WordNet. The synonyms were assigned the same value as the original adjective as input to WordNet. The set of adjectives which were used in the expansion process contained an entry and a value. The sentiment propagation algorithm used in the expansion process was a modified version of the algorithm proposed by Hatzivassiloglou and McKeown [1997], and used conjunctions to identify and predict new sentiment words and their orientation. Candidate words were identified by their "connection" by a conjunction ("and" or "or") to a word of known sentiment orientation. The following sequence was extracted in the experiments for this chapter: 1. good and cost-efficient, 2. cost-efficient and fair, 3. fair and transparent. The word "good" was in the original adjective list and therefore had a known sentiment orientation. It's value was propagated from "good" to cost-effective and then to "fair" and finally to "transparent". Each iteration of the algorithm produced new adjectives which were not in the input list of words. The new words were expanded with semantic equivalents from WordNet. A new input list which consisted of the newly expanded words was created and used as a seed list for the new iteration of the algorithm. This process was continued until no more new words were produced. The adjectives were manually verified at the end of the process. The process produced two types of resources: 1. a list of adjectives with a class name, which was a resource for the GATE Platform and 2. a list of adjectives with a value, which was a resource for an external scoring process.

The next lexical resource constructed for the GATE platform was a set of sentiment modifiers and negators. Sentiment modifiers are typically adverbs which either increase or reduce the sentiment strength of an adjective, whereas a negator inverts the sentiment of an adjective [Benamara, Cesarano, Picariello, Reforgiato, and Subrahmanian, 2007]. An example

of a sentiment modifier is the adverb "very" which increases the sentiment strength of an adjective, i.e. the bigram "very good" has an arguably stronger sentiment strength than the unigram "good". An example of a negator is the adverb "not" which inverts the sentiment direction of an adjective, i.e. "not good" has the opposite sentiment direction of "good".

The discovery of adverbs used the steps in the adjective discovery process which produced a list of sentences which contained an adjective and an economic actor named entity. These sentences were required because it was necessary to locate adverbs which had an identifiable influence upon an adjective. This process is discussed on page 75. This group of sentences will be known as "adjective group of sentences". The next step is to calculate a PointWise Mutual Information (PMI) score for adverbs and their co-occurrence with a sentiment adjective. The PMI calculation is described in the following equation,

$$PMI(sa, ad) = \log_2 \frac{Pr(sa, ad)}{Pr(sa)Pr(ad)} \tag{4.5}$$

where sa which represents all the adjectives, ad is an individual adverb and Pr represents a probability.

The probability of an adverb and sentiment adjective ($Pr(sa, ad)$) is calculated by the following equation:

$$Pr(sa, ad) = \frac{Nu(sa, ad)}{Nu(ad)} \tag{4.6}$$

where $Nu(sa, ad)$ is the frequency an individual adjective appear co-occurs with a sentiment adjective in the same sentence. $Nu(ad)$ is the frequency of the adverb in the whole news corpus.

The probability of an individual adverb was calculated as per the following equation,

$$Pr(ad) = \frac{Nu(ad)}{Nu(AllAdverbs)} \tag{4.7}$$

where $Nu(ad)$ is the frequency of an individual adverb, and $Nu(AllAdverbs)$ is the frequency of all adverbs in the news story corpus.

The frequency of all adverbs refers to the words which meets the criteria of the following tags:RN and WRB. For example, if there was two adverbs in the whole corpus each appearing the corpus once then $Nu(AllAdverbs)$ would be two.

The probability of an individual adjective was calculated by the following equation,

$$Pr(sa) = \frac{Nu(sa)}{Nu(AllAdjectives)} \tag{4.8}$$

where $Nu(sa)$ is the frequency of an individual adjective, and $Nu(AllAdjectives)$ is the frequency of all adjectives in the news story corpus.

The frequency of all adjectives is a count of all of the adjectives in the news corpus. For example, if there were two adjectives in the news corpus, each occurring once then $Nu(AllAdjectives)$ is equal to two.

Adverbs which scored a PMI of 0 or less were removed. An example of the adverbs discovered are in Table 4.2. This process produced a list of adverbs which contained: an adverb and a class name. These lists were added to GATE. An example of the discovered adverbs is described in Table 4.2 with a manually added categorization label. The manual categorization relied upon a native speaker's understanding of adverbs and their sentiment modification properties[1].

Sentiment modifier categorization	Examples
Maximization	sharply, super, perfectly
Minimization	rickety, piffling, just
Negation	not, none, never

Table 4.2: Selection of sample of sentiment modification adverbs.

The last lexical resource produced was the "property of economic actor" (PEA) list. A PEA is a noun, which has a relationship with an economic actor (EA) which is described by either an event verb or adjective. For example: Microsoft's profits fell, the verb "fell" describes the relationship of the noun "profits" to the company "Microsoft".

At this stage the previously mentioned lexical resources (event verbs, sentiment adjectives and adverbs) developed for GATE have been added, and available to use in the PEA construction process, therefore the PEA is dependent upon the previous construction steps.

The first step of the PEA construction process was to execute the common stages described on page 73. In addition to the identification of named entities and lexical information, event verbs, sentiment adjectives and adverbs were annotated in the sentences returned by the common stages. The next step was to identify sentences which contained nouns which were not proper nouns, and contained one of the following: 1. business event verb or 2. sentiment adjective. Proper nouns were excluded because they are typically named entities, and therefore could not be a property of another named entity. The nouns from these sentences were extracted. The next step was to identify an affinity between an extracted noun and: 1. sentiment adjective or 2. business event verb. The method of detecting this affinity was to calculate a Pointwise Mutual Information (PMI) score for each noun and all

[1] A discussion of sentiment negators and sentiment modifiers can be found at [Hogenboom, van Iterson, Heerschop, Frasincar, and Kaymak, 2011] and [Benamara, Cesarano, Picariello, Reforgiato, and Subrahmanian, 2007]

of the business event verbs and sentiment adjectives. The PMI score is described in the following equation,

$$PMI(n, sabv) = \log_2 \frac{Pr(n, sabv)}{Pr(n)Pr(sabv)} \tag{4.9}$$

where *sabv* represents all business event verbs or sentiment adjectives, Pr a probability and n an extracted noun.

The $Pr(n, sabv)$ was calculated with the following equation,

$$Pr(n, sabv) = \frac{Nu(n, sabv)}{Nu(n)} \tag{4.10}$$

where $Nu(n, sabv)$ is the frequency of a noun co-occurring with a sentiment adjective or business event verb in the same sentence and $Nu(n)$ is the frequency of the noun in the whole news corpus.

The probability of n is calculated by computing the frequency of the noun in the whole corpus and diving the number by the total number of nouns in the whole corpus. This calculation is represented in the following equation,

$$Pr(n) = \frac{Nu(n)}{Nu(AllNouns)} \tag{4.11}$$

where $Nu(n)$ is a frequency of a noun, and $Nu(AllNouns)$ of all nouns in the corpus.

The probability of the *sabv* was calculated by computing the frequency of all sentiment adjectives and business event verbs and dividing the result by the total frequency of all the business event verbs and sentiment adjectives in the corpus. The calculation is represented by the following equation,

$$Pr(sabv) = \frac{Nu(sabv)}{Nu(AllSABV)} \tag{4.12}$$

where $Nu(sabv)$ is the frequency of the event verb or sentiment adjective, and $Nu(AllSABV)$ is the frequency of all the business event verbs and sentiment adjectives.

The nouns which had gained a score of 0 or less were removed. The remaining nouns were assumed to be properties of named entities, and were added to GATE as a list with the class name "PEA". An example of the extracted nouns can be found in Table 4.3. The small sample has been hand classified under an artificial noun category for clarification purposes.

At this stage all the GATE lexical resources which contained an entry and a class name, had been created. The remaining steps were to create separate resources for an external scoring process. These new resources were created from the GATE resources, and contained an entry and a value rather than an entry and a class name which were present in the original GATE resources.

Noun Categorization	Examples
Success Measures	footfall, sales, profits, demand
Time Periods	Monday, Tuesday, January, month, year, period
Third Parties	investors, analysts, investors, economists, regulators, consumers
Miscellaneous	transactions, finance, bankruptcy

Table 4.3: Sample selection of Properties of Economic Actors (PEA).

Lexical Resources for Scoring Process

The lexical resources for the scoring process were a subset of the ones developed for GATE. The unique difference was that the class label was replaced with a value, therefore the scoring process resources had an entry and a value. The resources to be scored were: business event verbs, PEA and adverbs. Each of the scoring processes was conducted manually and relied upon the intuition of a native speaker. It is accepted that the scoring process could be seen as "ad-hoc", however there was no authoritative alternate scoring processes.

The sentiment adjectives have been already scored in a process described on page 75. The scores of the business event verbs and sentiment adjectives indicated the polarity of a verb or adjective, i.e. a positive score indicates a positive polarity for a verb or adjective, and a negative score indicates a negative polarity. The scores for adverbs indicate their influencing role on an adjective, i.e. an adverb with a score of less than 1 reduces the score of an adjective where as a score of greater than 1 increases the positive score and reduces the negative score of an adjective. An adverb which has a score -1 inverts the score of an adjective. The score of a PEA indicates the relationship between a business event verb and a PEA. A score of 1 indicates that the PEA has a "typical" relationship with a business event verb and does not alter the value assigned to a verb, and -1 indicates that the PEA has a "atypical" relationship with a business event verb which alters the value of a business event verb.

The first step is to score the business event verbs. The verb scoring process used one annotator who estimated the relationship of an individual business event verb with the majority of the PEA nouns. A verb which in the opinion of the annotator had a "positive" relationship with the majority of the PEA nouns was assigned a score of 1 whereas a verb with a "negative" relationship with the majority of PEA nouns was assigned a score of -1. A positive relationship for a verb indicates that the verb forms part of a "positive phrase" with the majority of PEAs. For example, the "positive" phrase, "Microsoft profits rose in 2011", "rose" and "profits" are the verb and PEA respectively, consequently the verb "rose" would be assigned a score of 1. A negative relationship for a verb indicates that the verb forms part of a "negative phrase" with the majority of PEAs. For example, the "negative

phrase", "Microsoft profits dropped in 2011","dropped" and "profits" are the verb and PEA respectively, consequently the verb "dropped" would be assigned the score -1. A sample of the values of assigned business event verbs' values are in Table 4.4 with an artificial verb category.

Verb Category	Examples
Obtained	gain(1), add(1), forge(1), win(1), attract(1)
Lost	fire(-1), cut(-1), cancel(-1)
Direction	climb(1), fall(-1), boost(1), down(-1)
Behaviour	storm(1), unravel(-1)
Influence	hurt(-1),hit(-1), push(1), suffer(-1)

Table 4.4: Sample Verb Values.

The next step was to score the "Property of Economic Actor (PEA)" nouns. The scoring process was predicated upon two concepts: 1. "typical relationship" and 2. "atypical relationship". A typical relationship is in the opinion of the annotator the PEA does not effect the scores of the business event verbs. For example, the phase, "Microsoft profits rose in 2011", the PEA "profits" does not alter the score (1) of the verb "rose". An atypical relationship of a PEA is in the opinion of the annotator the PEA does effect the scores of the business event verbs. For example, the phase, "Microsoft costs rose in 2011", the PEA "costs" does alter the score (1) of the verb "rose" to a negative score because "rising costs" are normally seen to be negative. The scoring process for the PEA nouns was a manual process. It is accepted that this may be an "ad-hoc" process, but it was only undertaken because of the lack of an authoritative alternate process. A PEA noun was assigned a score by a single annotator. A PEA was assigned 1 if it had a majority positive relationship with the business event verbs otherwise it was assigned a score of -1. The -1 score was designed to "invert" the score of the event verb, for example the PEA noun "costs" was assigned a score of -1, and in the hypothetical phrase "Premier Foods cut costs" would invert the score of the business verb "cut". A sample of PEA nouns and their relationships with business event verbs can be found in Table 4.5.

PEA Category	Examples
Typical	profits, footfall, sales, demand
Atypical	costs, debt, unemployment, arrears

Table 4.5: A sample of Properties of Economic Actors and their relationship with business event verbs

The final step was to score the adverbs. The annotator assigned a score which reflected the role of an adverb with a sentiment adjective. The range was arbitrary, but reflected the

annotator intuition about the adverbs.

A sentiment maximizer (adverb) seeks to increase the pre-assigned value of an adjective, and consequently it was scored on the scale $1.1 <= am < 2.0$, where am is the sentiment maximizer value. A sentiment minimizer (adverb) seeks to reduce the pre-assigned value of an adjective, consequently it was scored on the scale $0.9 <= ami < 0.1$, where ami is the sentiment minimizer value. The negator value is -1. The scoring of the sentiment minimizers and maximizers was a manual process which reflected the annotators opinion of the "effect" of the adverb on a sentiment adjective. It is accepted that this may be an "ad-hoc" process, but was used in the absence of an authoritative alternative. The negator value is a commonly used value[Hogenboom, van Iterson, Heerschop, Frasincar, and Kaymak, 2011].

4.1.2 Jape Rule Construction for Phrase Annotation

The resources added to GATE were used to annotate sentiment or event phrases in text. In this context "annotate" refers to the process of delimiting a phrase and attaching a label, for example the hypothetical phrase, "Virgin Trains loses West Coast Mainline franchise." could be labelled as an "event phrase" because it obeys the <subject> (Virgin Trains) <verb> (loses) <object> (franchise) pattern. The sentences which contain the annotated phrases are extracted and sent to a separate scoring process.

The annotation step is conducted within GATE. GATE uses inbuilt resources and the newly constructed lexical resources to: 1. Delimit sentences and 2. label: economic actors, business event verbs, sentiment adjectives, adverbs and properties of economic actors within sentences. The annotation of the phrases within sentences is with JAPE (Java Annotation Patterns Engine) rules which is a form of regular expression for annotations applied by tools in the GATE platform [Cunningham and Tablan, 2002].

JAPE rules have their own syntax, an example of which can be found in the appendix on page 156. The simplified anatomy of an event JAPE Rule is the following:

- Rule accepts as arguments: a sentence text and named entities in the text.
- If sentence does not contain any named entities, return sentence with no annotations
- If sentence does not contain: economic actor, event verb or property of economic actor, return sentence with no annotations
- Find position of economic actor, event verb or property of economic actor in sentence
- If event verb occurs earlier (left of sentence) than both economic actor and property, return sentence with no annotations

- If event verb occurs later (right of sentence) than both economic actor and property, return sentence with no annotations
- Locate pairs of economic actor and property of economic actor.
- Annotate phrase with text between the economic actor and property of economic actor.
- Add economic actor and property of economic actor to the phrase.

The sentence is annotated with an annotation label which captures the phrase using the location of the: Economic Actor (EA), Business Event Verb and Property of Economic Actor (PEA) in the sentence. This annotation label will be used in the scoring process to locate the phrase. There was one JAPE rule written for each type of economic actor for each phrase type (event or sentiment), consequently there were 10 individual JAPE rules written. Although the rules are used in conjunction they were separated in case it was required to analyse each economic actor separately.

4.1.3 Scoring Annotated Phrases

The scoring process had access to the GATE APIs and therefore was able to extract sentences which had been labelled by the JAPE rules as either containing a sentiment or event phrase. The scoring application had access to the sentence level annotations made by GATE, e.g. sentiment adjectives, event verbs, etc.

There were two separate scoring strategies, one for event phrases and another for sentiment phrases. The event phrases are scored by assigning the value applied to the business event verbs (see page 81) to the whole phrase unless the PEA noun (see page 81) has a score of -1 where the business event verb score is inverted.

An example of a "positive" event phrase is: "Take-Two posts profit" where Take-Two is an economic actor (company), "posts" is an business event verb and "profit" is a property of an economic actor. The scoring process assigned a value of "1" to both the business event verb and the PEA, and therefore the "score" of the phrase is 1. It is possible to alter the phrase from positive to negative by changing the noun "profit" to "loss". The PEA "loss" had been assigned a value of -1 in the scoring process, and consequently the scoring strategy would invert the score of "post (1)" and score the strategy as -1. The annotated phrase is short, but clearly shows the anatomy of an event phrase and how it is scored.

Sentences typically contained one event phrase and therefore a sentence score would be the phrase score it contains. In the event of a sentence containing two event phrases or more then the event phrases would be summed to produce a final event score for a sentence.

Sentiment phrases were scored with a variation of pre-existing algorithm developed by Subrahmanian known as AVAC [Subrahmanian and Reforgiato, 2008]. The AVAC algorithm first

creates a *dependency tree* of unigrams in the candidate phrase. For example, the candidate phrase "Beni Prasad not 'happy' with rising prices." is represented as a dependency tree by the Stanford Parser [de Marneffe, MacCartney, and Manning, 2006] as the following.

```
(ROOT
  (S
    (NP (NNP Beni))
    (VP (VBP Prasad)
      (S
        (ADJP (RB not) (`` `) (JJ happy)))
      (PP (`` `) (IN with)
        (S
          (VP (VBG rising)
            (NP (NNS prices))))))
    (. .)))
```

The dependency tree allows the identification of dependencies between adverbs and adjectives. These dependencies can typically be retrieved from the dependency parser itself. The Stanford Parser returns the following dependencies for the example phrase ("Beni Prasad not 'happy' with rising prices.").

```
nsubj(Prasad-2, Beni-1)
root(ROOT-0, Prasad-2)
neg(happy-5, not-3)
acomp(Prasad-2, happy-5)
prep(Prasad-2, with-7)
pcomp(with-7, rising-8)
dobj(rising-8, prices-9)
```

The parser has identified a negation dependency relationship between "not" and the sentiment adjective "happy".

If a phrase has a dependency between an adjective and an adverb, a new score is computed for the adjective using the following equation,

$$ps = smn \cdot adj \tag{4.13}$$

where ps is a phrase score, smn is a sentiment modifier or negator score and adj is a sentiment adjective score.

In the example phrase, "Beni Prasad not 'happy' with rising prices." the adverb "not" was assigned a score of -1 and the adjective was assigned a score of 0.875. The new score of

the adjective was calculated using the following values, 0.875×-1. The new score for the adjective "happy" was -0.875. This score would be assigned to the phrase. Typically, there was one sentiment adjective per sentiment phrase. In the event of two or more sentiment adjectives the values of the adjectives and adverbs were summed to produce a phrase score.

Sentences typically contained one sentiment phrase and therefore a sentence score would be the phrase score it contained. In the event of a sentence containing two sentiment phrases or more then the sentiment phrases would be summed to produce a final sentiment score for a sentence.

A final combination possibility is the combination of sentiment and event phrases. Sentences which contained either a sentiment phrase or an event phrase were assigned the respective phrase score. Sentences which contained: multiple sentiment phrases, multiple event phrases or multiple sentiment and event phrases, were assigned the sum of the phrase score.

In the full system, which is described in Chapter 6, a single sentence level score was required for each trading day. The calculation of the single score was computed by the following steps:

- Split news stories published in a single day into sentences.
- Score each sentence with either: sentiment, event or sentiment and event rules.
- Sum the scores assigned to each sentence.

The final score is added as a feature to an existing stock trading system.

4.1.4 Evaluation

The evaluation tested the ability of the sentence level strategy to label and score sentiment and events phrases correctly. The evaluation for the sentence level strategies was against a small manually annotated gold standard. The gold standard was a collection of 770 randomly selected sentences from the news corpus. The sentences were manually classified into the following categories.

- Event
- Sentiment
- Objective
- Sentiment or Event

The annotator was allowed to mark a sentence as "Event or Sentiment" if, 1. a sentence was not objective and 2. it was not clear if the sentence was a distinct member of either the Event

or Sentiment categories. Each sentence was assigned an additional label which indicated the direction of the sentence. The values were: "N " indicates a "negative sentence" and "P" indicates a "positive sentence". An example of an annotated sentence is the following:

"UK investment trusts have suffered greater losses from the market turmoil than any other fund type in Europe, according to data from the European Fund and Asset Management Association, with overall assets sliding by 30.8 per cent to £55bn (£47.5bn, $69.7bn) in the first nine months of 2008."

This sentence was manually labelled as an "Event" with an additional label of "N" which indicates it is a negative sentence.

The event and sentiment rules were evaluated separately. These rules were evaluated on a binary classification problem. Event rules were evaluated on their ability to discriminate among sentences that were hand classified as either "Event" or "Sentiment or Event" from the remaining sentences, whilst the sentiment rules were evaluated on their ability to discriminate among sentences that were hand classified as either "Sentiment" or "Sentiment or Event" from the others. The goal of the each of the binary classification problems is to identify the target classes, often referred to as the "positive class". In the case of Event rules the "positive class" is constructed from sentences tagged as "Event" or "Sentiment or Event" while for the sentiment rule the "positive class" is constructed from sentences tagged as "Sentiment" or "Sentiment or Event". For this type of binary classification the correct evaluation framework is to use Precision and Recall which are predicated on the following notions:

- False positive (FP) a sentence which was incorrectly identified as belonging to the positive class.
- False negative (FN) a sentence which was incorrectly identified as belonging to the negative class.
- True positive (TP) a sentence which was correctly identified as belonging to the positive class.
- True negative (TN) a sentence which was correctly identified as belonging to the positive class.

The above notions can be used to calculate: Precision, Recall and F-measure which is their harmonic mean [Voorhees and Harman, 2003].

The Precision calculation for each task was achieved with the following equation,

$$Precision = \frac{TP}{TP + FP} \tag{4.14}$$

where TP is the number of true positives and FP is the number of false positives.

The Recall calculation for each task was achieved with the following equation,

$$Recall = \frac{TP}{TP + FN} \tag{4.15}$$

where TP is the number of true positives and FN is the number of false negatives.

The F-measure calculation was achieved with the following equation,

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.16}$$

These three metrics were used to evaluate the sentiment and event rules. In addition, the same metrics were used to evaluate the ability of the rules to estimate an orientation of the phrases against the manually obtained labels P (positive) and N (negative). More specifically, the result of the scoring process (a number) was transformed into a predicted direction, with positive scores corresponding to the P class and negative numbers corresponding to the N class.

The results for the sentence level strategies are documented in Table 4.6.

Evaluation Item	Precision	Recall	F-Measure
Sentiment phrase annotation	0.84	0.72	0.78
Event phrase annotation	0.64	0.78	0.70
Sentiment phrase orientation	0.76	0.61	0.68
Event phrase orientation	0.53	0.72	0.61

Table 4.6: Recall and Precision for Phrase Extraction.

The results of the strategy were encouraging, however the gold standard represented a small selection of the possible sentences from the available news corpus. It was not feasible to annotate a substantially larger gold standard because it would have taken a substantial period of time. The purpose of this evaluation was to act as a baseline to ensure that the proposed strategy was not obviously flawed. The sentence strategies were designed to be part of a trading strategy, and consequently its classification robustness, although important, was not its ultimate evaluation. The final justification for these strategies will be a trading evaluation which will be described in Chapter 6.

4.2 Document Level Strategies

The document level strategies differed from the sentence level strategy because they represented a day's news as a collection of news stories rather than a collection of sentences. The document level strategies represent a news story as a *bag of words*. A bag of words

representation of a news story does not preserve linguistic information between words such as negation, sentiment maximization and minimization.

The document level strategies use classifiers which induce models from labelled data. As stated at the beginning of this chapter there was no resource available to label large amounts of data, consequently the labelled data in the document level strategies used various automated labelling strategies. The labelling strategies used: 1. sentence level rules, 2. fluctuations in a market index, 3. a combination of sentence level rules and fluctuations in a market index to label data for a classifier and 4. a combination of sentence level rules and fluctuations in a market index to label data for a classifier with self-training.

The document level strategies are predicated upon several notions from the machine learning field. The notions are: balancing, self-training and co-training which will be explained in the following text.

Balancing is a method of equalizing the number of training instances in each class of training data. Training sets which contain classes with differing numbers of texts may "confuse a classifier" induced from it [Batista, Prati, and Monard, 2004]. This "confusion" often leads to unsatisfactory classifiers induced from unbalanced data [Provost, 2000]. An alternative to unbalanced training data is to balance the training data. There are a number of methods of balancing data, for example increasing the number of instances in the minority class with synthetic data [Chawla, Bowyer, Hall, and Kegelmeyer, 2002]. A sample of competing balancing techniques can be found in a paper by Japkowicz [2000].

Self-training is a process where a base learner is induced from initial labelled data. The base learner then classifies the unlabelled data. Unlabelled data which has been classified with a sufficiently high confidence is added to the labelled data, a new model is induced and the process continues until a specific stopping condition is met[Abney, 2007].

Co-training is a form of "hard constraint" which limits the the news stories accepted as new training data. Co-training induces separate models for each "view" of a document and the separate models are used in conjunction to classify the document [Abney, 2007].

4.2.1 Labelling Strategies

The first strategy which will be known as the "rule based labelling" strategy used the rules developed for the sentence level strategies to label news stories. The sentence rules label headlines of news stories, and assigns it to one of three categories: neutral, positive and negative. Headlines were chosen as the labelling criteria because there is a known existing relationship between headlines and the story text [Andrew, 2007].

Headlines are typically constructed from a single sentence which summarizes the following news story, for example "Retailers face 'perfect storm' of rent day". The headlines often

convey sentiment and event information which can be classified by the sentence level rules. In this case the rules label and score the headlines, sentences which gain a score of above 0 are labelled as "positive", sentences which gain a score of below 0 are labelled as "negative", whilst the remaining headlines are labelled as neutral.

As discussed earlier the sentence level rules can label and score either: sentiment or event phrases. It is possible to combine the rules into the following labelling strategies: 1. sentiment rules only, 2. event rules only and 3. sentiment and event rules. It is possible in the sentiment and event rules both sets of rules label and score the same sentence, in that circumstance the two scores are added together. The combined score is used to assign a news story to a category.

The rule labelling strategy can produce categories which had unequal number of documents, and consequently a balancing processing was used to ensure that each category had an equal number of documents. The chosen balancing strategy is known as "undersampling" [Kubat and Matwin, 1997]. The balancing strategy balanced the classes by using the following steps:

- Count number of instances in smallest class.
- Randomly sample the same number of instances from each of the larger classes.

This process ensures that each class has the same number of training instances.

The second labelling strategy used fluctuations in a market index to label news stories. The strategy used single day movements in a financial market to infer a category for the day's news. The strategy does not prescribe a choice of financial index, but the FTSE250 was used in these experiments because at the time of the experiments the majority of the news corpus consisted of stories from British media sources. All news stories were labelled by the following criteria: 1. all stories were labelled positive if they were published on a day where the market value rose above a given threshold, 2. all stories were labelled negative if they were published on a day where the market value fell below a given threshold and 3. all stories were labelled neutral if a story was published on a day where the market value moved less than the thresholds. There are no prescriptive values for the aforementioned thresholds. The values used in the final trading experiments in Chapter 6 were discovered through experimentation.

The justification of using single day movements to label stories was that single day movements in a financial market often have an underlying cause. Examples of single day movements and their underlying causes are described in Table 4.7.

The previous two strategies arguably had weaknesses. The rule based strategy can make errors when labelling the data and it did not consider the impact of the news story on the market. The market alignment based strategy may erroneously label news stories because

Date	FTSE (+/-)	Reason
8th August 2011	-3.39%	Falls in US and Asian Markets
10th/12th Sept 2011	-2.73%	Terrorist Attacks
7th Sept 2008	-1.93%	Financial Crisis

Table 4.7: Large Single Day Fluctuations in the FTSE.

they were published on the same day as a market movement by chance. A possible solution is to combine the strategies to limit the number of errors in the labelling process. This combined strategy labelled a story with a category if both the rule and market alignment strategies agreed on the same label, i.e. a story would be labelled positive if it was published on a day where a market value rose above a given threshold and its headline was classified as positive by the rule classifier. The stories when the classifiers disagreed were not assigned a label.

The combined strategy had three variants which were dictated by the three rule labelling possibilities described on page 89. The possible variants were:

- Sentiment rules only and market alignment.
- Event rules only and market alignment.
- Sentiment and event rules with market alignment.

The quantity of data produced by the each variant of the combined strategy was limited when compared to the rule labelling and market alignment strategies, therefore a form of self-training was used to increase the number of labelled news stories. Self-training selects "high-confidence" classifications and incorporates them in an iterative training process [Abney, 2007]. Although there are a number of semi-supervised strategies available self-training was selected as it was a mature strategy which is relatively simple to implement. An initial attempt with classical self-training was unsuccessful because the learner misclassified an unacceptably number of news stories, therefore a decision was made to "constrain" the self-training process to the news stories whose headlines had been labelled by the sentence rules. A second decision was made to use co-training [Abney, 2007]. Co-training uses separate "views" of a document to induce separate classifiers. A "view" of a document is a characteristic of document which provides a set of features which describes the document, for example the original work on co-training used hyperlinks and hyperlink text were used to describe web pages [Blum and Mitchell, 1998].

There were a number of possible "views" which could have been used to describe a news story. Three were selected: 1. news story headline, 2. RSS description text[2] and 3. news story text.

[2]RSS description field provides a short description of the news story.

These "views" were selected because: 1. each "view" provided a "rich" description of a news story and 2. each "view" did not repeat the features of the other[3]. Each "view" of a news story was represented as a "bag of words". Although there are a number of alternate representation methods a "bag of words" was selected because of the limitations of the software being used (Lingpipe [Alias, 2008]) which accepted only a "bag of words" representation. The algorithm induced a model for each news story "view". The models voted on each news story classification. If each model returned the same classification with a sufficiently high classification confidence then the news story was accepted, and added to the labelled data where new models for each view were induced. The process continued until no additional stories were added to the labelled data. The algorithm is explained in full in Algorithm 6.

There were three possible methods of varying the news stories selected. The variants were: 1. news stories whose headlines had been labelled by the event sentence rules, 2. news stories whose headlines had been labelled by the sentiment sentence rules, and 3. news stories whose headlines had been labelled by the sentiment and event sentence rules.

4.2.2 Document level strategies evaluation

The evaluation used in this chapter to compare the various labelling strategies was an estimated average F-measure gathered from a holdout process against a small gold standard. The task was a multi-class classification, and therefore an average F-measure was calculated. There are two methods of calculating average F-measure: micro and macro average [Özgür, Özgür, and Güngör, 2005]. The averaging method used in these experiments was a micro average. Micro average was used as this was the default in the software used for the experiments.

The micro average method is described by Özgür, Özgür, and Güngör [2005] as the following equations,

$$F(micro - averaged) = \frac{2\pi p}{\pi + p} \tag{4.17}$$

where p is obtained by the following equation,

$$p = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FN_i)} \tag{4.18}$$

where TP is a true positive, FN is a false negative and M is the number of categories.

π is obtained by the following,

[3]This observation was confirmed by a manual inspection.

Algorithm 6: Self-Training Algorithm.

Input: UL: A list of unlabelled stories
Input: LD: A list of labelled stories
Input: Const: A predefined constant for classifier confidence
** Instantiate Rule Classifier **;
$rc \leftarrow newRuleClassifier()$;
** Train models from labelled data **;
** HC -> Induce model for headlines from labelled data **;
$hc \leftarrow TrainWithHeadlines(LD)$;
** DC -> Induce model for description from labelled data **;
$dc \leftarrow TrainWithDescriptions(LD)$;
** DC -> Induce model for news story text from labelled data **;
$sc \leftarrow TrainWithStoryText(LD)$;
$ul \leftarrow ()$;
$counter \leftarrow 0$;
forall the $story \in UL$ **do**
    ** Classify news story with each model **;
    $ruleC \leftarrow rc.classify(story.headline)$;
    $headC \leftarrow hc.classify(story.headline)$;
    $descC \leftarrow dc.classify(story.description)$;
    $textC \leftarrow sc.classify(story.text)$;
    ** Check classification confidence **;
    if $headC.conf < Const$ *or* $descC.conf < Const$ *or* $textC.conf < Const$ **then**
        $next$;
    ** Check classification agreement **;
    if $ruleC = headC \wedge headC = descC \wedge descC = textC$ **then**
        $LD \leftarrow LD.add(story, ruleC)$;
        $counter \leftarrow counter + 1$;
    else
        $ul \leftarrow ul.add(story)$;
** Termination, no further candidates **;
if $counter = 0$ **then**
    return LD;
** Recursive call to Self-Train Function **;
return $(SelfTrain(LD, ul, const))$;

$$\pi = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FP_i)} \tag{4.19}$$

where FP is a false positive.

The evaluation process had a hand labelled set of 935 documents (evaluation set) which were labelled into three categories: neutral, positive and negative. The competing strategies did not have access to this data in the labelling or training phase. In addition, there were 421984 unlabelled news stories from which the labelling strategies would label a selection as training data for a Naive Bayes learner which would be used to classify the small hand labelled set of documents.

The rule, combined and self training strategies had a number of common variations which was not shared by the market alignment strategy. The rule strategy had three variations: 1. label news stories with event rules only, 2. label news stories with sentiment rules only and 3. label news stories with both event and sentiment rules. The combined labelling strategy used the rule variations and market fluctuations to label news stories, consequently there were three sets of labelled data produced by this strategy one for each of the following combinations: 1. event rules and market fluctuations, 2. sentiment and market fluctuations and 3. event and sentiment rules with market fluctuations. The self-training variations used the combined strategy variations to label the initial training data. The self-training algorithm used: 1. event, 2. sentiment, and 3. event and sentiment rules as the rule classifier.

The labelling strategies which relied upon market alignment used the FTSE to assist the labelling process. The FTSE is a British financial index and was chosen because at the time the experiments were conducted the news corpus mainly contained stories from the UK media. The labelling strategies which used market alignment used a number of thresholds which ranged: 1. 0.5% to 2.0% of the FTSE market value for positive news stories and 2. -0.5% to -2.0% of market value for negative news stories. These limits were dictated by the trading evaluation described in Chapter 6.

The results are documented in Table 4.8. There are two models induced for each variation strategy: one model induced from news story headline text and another from news story text. There are a range of results for the labelling strategies which used market fluctuations, and consequently the results are presented for these strategies as a range. The F-measure was calculated with Lingpipe's [Alias, 2008] implementation which calculates a micro average F-measure[4].

The worst performance in terms of estimated F-measure was the market alignment strategy whose results were inferior to all of the competing strategies. The remaining strategies have similar estimated F-measures, but it is arguable that the combined rule and market alignment labelling strategies are superior because they have marginally higher estimated F-measure than the competing strategies.

This evaluation was not intended to be a comprehensive evaluation, but a baseline evaluation

[4]More information can be found at `http://goo.gl/vB23t`

Labelling Strategy	Model	Estimated F-measure
Sentence Rules (Event only)	Headline	0.67
Sentence Rules (Event only)	Text	0.47
Sentence Rules (Sentiment only)	Headline	0.51
Sentence Rules (Sentiment only)	Text	0.47
Sentence Rules (Sentiment and Event)	Headline	0.64
Sentence Rules (Sentiment and Event)	Text	0.52
Market Alignment	Headline	0.29 - 0.45
Market Alignment	Text	0.31 - 0.42
Combined (Event + Market Alignment)	Headline	0.59 - 0.73
Combined (Event + Market Alignment)	Text	0.41 - 0.57
Combined (Sentiment + Market Alignment)	Headline	0.43 - 0.58
Combined (Sentiment + Market Alignment)	Text	0.37 - 0.52
Combined (Event and Sentiment + Market Alignment)	Headline	0.56 - 0.65
Combined (Event and Sentiment + Market Alignment)	Text	0.34 - 0.50
Self-train (Event + Market Alignment)	Headline	0.57 - 0.70
Self-train (Event + Market Alignment)	Text	0.37 - 0.54
Self-train (Sentiment + Market Alignment)	Headline	0.43 - 0.58
Self-train (Sentiment + Market Alignment)	Text	0.37 - 0.52
Self-train (Event and Sentiment + Market Alignment)	Headline	0.55 - 0.66
Self-train (Event and Sentiment + Market Alignment)	Text	0.37 - 0.52

Table 4.8: Estimated F-measure for competing labelling strategies.

intended to show the viability of each strategy. The applicability of each strategy to trading is shown in Chapter 6.

4.3 Comparing Sentence and Document Level Strategies

The goal of the work presented in this chapter is to obtain and aggregate the value of news retrieved for each day. There have been presented two alternate approaches to analyse these news stories which were at the: 1. sentence level and 2. document level. This section will compare these alternatives in terms of their ability to produce a daily score for news stories which is similar to a human annotator.

The evaluation used a company specific ontology news retrieval system (see page 63) to recommend news for "Microsoft" for the date 20-06-2009, which was randomly selected for the evaluation. The news retrieval system recommended 64 news stories. The news stories were hand scored assigning a value of 1 to a positive news story, 0 to a neutral news story

and -1 to a negative news story. An average news story score was calculated for the day.

It was not possible to compare all the strategies and their variations, consequently one representative strategy was selected from the sentence level and one from the document strategy. The selected sentence level strategy was the combination of event and sentiment rules at the news story text level. The selected document strategy was the combined strategy which used a event rule classifier.

The document level strategy classified news stories into one of three categories: neutral, positive and negative. The classification label was substituted for a value: 0 for neutral, 1 for positive and 1 for negative. An average score for the day was calculated by dividing the total news score for the day.

The sentence level strategy was adapted to produce a document score as it was not possible in the time available to manually label and score all sentences in the selected news stories. The document score was calculated by dividing the score computed for all sentences in a news story by the total number of sentences in a news story. For example, if a total sentence score for a document was 100 and there was 100 sentences in the news story then the document score would be 1. A maximum score for a document was limited to 1 and a minimum score for a document was -1. If a score greater than 1 was returned then it would be replaced by 1, if a score less than -1 was returned it would be replaced by -1 [5]. Values between -1 and +1 were not altered. An average sentence level score was calculated for the day by summing all of the document scores computed by the sentence level strategy and dividing it by the number of stories. The combination of event and sentiment was an average of the averages for the day's sentiment and event scores.

The results are in Table 4.9. The manually labelled data set suggests that overall the day was slightly positive. The sentence level strategy produced a positive value whereas the combined strategy produced a slightly negative value. The majority of values produced by both strategies were 0 which was similar to the manual evaluation. Although the combined strategy produced a marginally negative value, the evaluation provides some evidence that the strategies produce a single day news score similar to a manual annotator.

Average Document Score		
Manually Labelled	Sentence Level Strategy	Document Level Strategy
	Event + Sentiment	Self-Training
0.03±0.66	0.06±0.68	-0.02 ±0.71

Table 4.9: Evaluation of sentence and document level strategies' calculation of a daily news value.

It may be possible that the results described in Table 4.9 could have been achieved by chance,

[5] In practise this was not necessary as all stories were scored with a value between -1 and 1

consequently an alternate evaluation was conducted where the accuracy of the classifications was measured. The results are presented as a percentage of stories estimated correctly. The results are presented in Table 4.10. The sentence level strategy estimated the direction of a news story more accurately than the combined strategy.

Accuracy	
Sentence Level Strategy	Document Level Strategy
Manually Labelled	Combined (Event Rules + Market Alignment)
57.81%	50.00%

Table 4.10: Evaluation of sentence and document level strategies'.

4.4 Discussion of results

This chapter has presented two strategies for classifying information in news stories. The strategies attempted to classify news stories at two levels: sentence and document. The sentence level used rules to score event and / or sentiment phrases in a sentence. A value for day's news is calculated by aggregating the score for all the sentences of news stories published in a single day. The sentence strategies were evaluated on a manually labelled gold standard.

The document level strategies used: rules, market fluctuations, and a combination of rules and market fluctuations to label news stories to be used to train a classifier. The competing labelling strategies were evaluated with an estimated F-measure using a small labelled set of news stories.

The strategies were evaluated with two types of tests. The first evaluation tested the ability of a strategy to classify a sentence or a document with the correct label and / or score. The second test evaluated the ability of a strategy to produce a single score for a day's news. The evaluations were limited because it was not possible to manually label large amounts of data. It is debatable whether large amounts of labelled data would provide a more through evaluation because these strategies were designed for trading, and there was no authoritative method for labelling news stories with reference to their effect on a financial market. The work presented in this chapter provides some evidence that the presented strategies are able to classify news stories or sentences. The ability of these classification and labelling strategies to enhance a trading strategy will be evaluated in Chapter 6.

Chapter 5

Direct Speech Analysis

A news story may contain direct speech from a person who has a strong connection with the subject matter. A person who has direct connection to a news story may have knowledge which is not explicitly expressed in the news story, therefore the person's words in the quoted speech may have more effect on a stock or financial market than the remaining news story text. There are a number of examples where direct speech alone has dramatically moved a market or share value. This chapter describes two novel approaches which were designed to classify direct speech with the aim of using this information for financial trading.

5.1 Introduction

There are many examples of direct speech which have had a direct impact on financial markets. The following are illustrative examples of this effect.

Gerald Ratner was the CEO of Ratners which was a successful UK chain of low cost jewellers in 1980s and 1990s. In 1992 Gerald Ratner gave a *humorous* speech to the Institute of Directors (IOD). Two quotes were reported in the UK mass-media [Ratner, 2007].

The first quote was:

> We also do cut-glass sherry decanters complete with six glasses on a silver-plated tray that your butler can serve you drinks on, all for £4.95. People say, "How can you sell this for such a low price?", I say, because it's total crap.

The second quote was:

> We sold a pair of earrings for under £1,which was cheaper than a prawn sandwich from M&S, but probably wouldn't last as last as long.

Ratners the company lost 500 million pounds in market value and had to change its name to Signet to distance itself from his speech [Ratner, 2007]. The phrase *doing a Ratner* entered the British lexicon as a metaphor for making a serious error.

The second example is the case of Mervyn King who was the governor of the Bank of England in 2008. He stated:

> So, taken together, the combination of a squeeze on real take-home pay and a decline in the availability of credit poses the risk of a sharp and prolonged slowdown in domestic demand. It now seems likely that Britain is entering a recession.

The effect of the quote was to push the level of the pound to a five year low against the dollar, and to remove 400 points from the Dow[1]. The information in the speech by King was not new because his hypothesis had been stated earlier in the press by journalists and finance professionals, but did not have the same effect on the markets.

The last example is the case of David Sheppard who in 2001 was the CEO of *Topman* which is a UK based clothing retailer. He gave an interview to *Menswear Magazine* which was widely reported in the British mass-media. In the interview he replied to a question which asked him who were the target customers of his company. In reply he stated his customers were:

> Hooligans who were more likely to wear a suit for a court appearance than to a job.[2].

The Arcadia Group, which was the parent group of Topman lost 4 pence from their share price.

The three previous examples demonstrate the ability of a *candid* quotation to effect a share price or financial market. The research literature, however, was sparse when searched for work which tried to identify a causal effect of direct speech and share price movement. Collingwood [2009] proposed that a CEO was the physical embodiment of a company, consequently what the CEO says or does can influence the company's future prospects and therefore its share price. Higgins and Bannister [1992] suggest that the communications of a CEO may contain information which could indicate the future direction of the company's share price.

CEOs speech may contain useful information, it may not always be truthful. Spindler [2006] states that lying may be a successful business strategy because it can manipulate certain

[1]More details about the speech can be found at: `http://news.bbc.co.uk/2/hi/business/7682723.stm`.

[2]More details about Sheppard's comments can be found at: `http://www.telegraph.co.uk/news/uknews/1334682/Topman-tailors-for-beer-swilling-hooligans.html`

audiences, for example customers, staff or traders. Larcker and Zakolyukina [2010] state that lying by CEOs can be detected because their lies have certain linguistic features.

News stories can also quote finance professionals such as analysts. There is a substantial body of work which documents the effects of "recommendations" made by analysts, but the research literature is sparse when evaluating the effect of an analyst's quotes in news stories.

Kuperman, Athavale, and Eisner [2003] suggest that the motivation for finance professionals who allow themselves to be quoted in the mass-media is threefold: 1. increasing their influence upon their fellow finance professionals, 2. providing wider exposure to their stock recommendations and 3. increasing business of their employer. The explicit hypothesis of Kuperman is that analysts wish to influence stock prices because it is beneficial to their clients and to their employer. Daly [2009] suggests that analysts pronouncements can effect the behaviour of investors. Busse and Clifton Green [2002] found that analysts who appear on television can influence the market within *seconds*. Tamura and Hiromichi [2002] stated that analysts form a consensus about a certain market or stock. If the Tamura and Hiromichi hypothesis is correct than if the analysts who are quoted in the media form a consensus about a financial market or stock then they may represent analysts in general. In summary, analysts quoted in news stories can influence the market because they act as *opinion formers* which influence other less high profile analysts which in turn influence their clients. This is in turn shapes investors attitudes towards markets and stocks.

In this context, it is claimed that text mining methodologies are appropriate to discover the relationship between the content of direct speech and its influence on share prices. This chapter's main goal is to describe two novel strategies which use direct speech analysis with the goal of identifying information which is useful for trading.

The proposed approach of this chapter is to classify direct speech in news stories into three categories: negative, neutral and positive. The label is replaced with a value, for example, a "positive label" could be replaced with a value of 1 and a "negative value" could be replaced with a value of -1. The substituted values for the quotes published on a given day are summed to produce a single value which is then used as a feature in a stock trading system. This approach faced two main challenges: 1. there was no freely available labelled data to train a classifier and 2. no resources to manually label large amounts of data, therefore the proposed methods were required either: 1. to create their own labelled data or 2. use external resources. Given these difficulties associated with direct speech data, two paths were explored: 1. use of linguistic strategies to label the data, and 2. manually label a small number of documents and apply semi-supervised strategies to add labels to some of the unlabelled documents.

Direct speech used in this chapter was contained in news stories. The extraction of this direct speech was achieved with the Open Calais web service [Reuters, 2010b] which extracted for each quote the following: 1. quote, and 2. the speaker. Open Calais, often, but not always,

extracted the following for a quote: 1. the speaker's affiliation and 2. the speaker's job title.

5.2 Related Work

The research literature search covered a number of areas which were pertinent for the classification of direct speech into sentiment categories. The researched areas included: bootstrapping, pre-compiled sentiment resources and semi-supervised learning. The bootstrapping and pre-existing linguistic sentiment resources can be grouped under the heading of "linguistic strategies" and therefore they are predicated upon the linguistic characteristics of the data and do not require labelled data. The bootstrapping strategies allow the extraction and labelling of linguistic units (unigrams, bigrams, multiword expressions, etc) from unlabelled data to be used in a future classification strategy. Pre-existing linguistic resources are pre-constructed resources which have labelled linguistic units with a polarity score which indicates a classification of the linguistic unit, i.e. a linguistic unit which has been assigned a score of greater than 0 could be classified as "positive". The following subsections will describe related work in the bootstrapping, pre-existing linguistic resources (sentiment resources) and semi-supervised learning.

5.2.1 Bootstrapping strategies

A first step in a bootstrapping strategy could be to create dictionaries with a linguistic unit (unigram, bigram, etc) and a sentiment label. These dictionaries could be used to assign a text to categories by scoring dictionary terms which are present in a text. A potential strategy for dictionary construction is to use known sequences of words to create multi-word expressions from unlabelled data. The extracted expressions can be assigned a sentiment orientation based on their co-occurrence with known sentiment words.

An example of this approach was described by Liu [2007a] who published a set of patterns of POS tags[3] (Table 5.1) which extracted a set of candidate bigrams. The pointwise mutual information was used to estimate the likelihood of each bigram,

$$PMI(t1, t2) = log \frac{Pr(t1, t2)}{Pr(t1)Pr(t2)} \tag{5.1}$$

where $t1$ is the first term (word) of a bigram, $t2$ is the second term (word) of a bigram and Pr is the probability of an expression.

Bigrams which had a PMI score of 0 or less were removed.

[3] A reference for POS tags can be found at: `http://goo.gl/ABLUn`

First Word POS Tag	Second Word POS Tag	Third Word POS Tag (not extracted)
JJ	NN or NNS	anything
RB, RBR, or RBS	JJ	not NN or NNS
JJ	JJ	not NN or NNS
NN or NNS	JJ	not NN or NNS
RB, RBR, or RBS	VB, VBD, VBN or VBG	anything

Table 5.1: Bigram POS sequences for the creation of sentiment bigrams.

The sentiment orientation for each bigram was calculated by its co-occurence with an anchor term of known sentiment orientation. Liu suggested the use of the words "poor" and "excellent" as the negative and positive anchor terms. The sentiment orientation calculation is given by,

$$SO(b) = log\frac{hits(b \text{ NEAR } p)hits(n)}{hits(b \text{ NEAR } n)hits(p)} \tag{5.2}$$

where b is a bigram, p is a "positive" anchor term and n is a "negative" anchor term.

Liu worked with Internet search engines, and therefore the terminology he used in Eq.5.2 refers to terms associated with Internet search. The term *hits* is synonymous with frequency because "hits" is typically the number of the documents returned by a search engine which contain the target word [Liu, 2007a]. *Near* is an operator for the Alta Vista search engine which determined maximum distance between two search terms (words) before they would be included in a search result [Liu, 2007a]. The "Near" operator is no longer supported by Alta Vista, but anecdotal evidence suggests the maximum distance was 10 words, i.e. two search terms had to have no more than 10 words separating them in a document before a document would be included in a search result.

A document or sentence is scored by taking an average of the SO score of the bigrams it contains. Liu claims an accuracy of 66% for movie reviews and 88% for automobile reviews [Liu, 2007a].

An alternate method for labelling linguistic units from unlabelled data is to use *polarity clues*. Polarity clues [Wilson, Wiebe, and Hoffmann, 2005] was a resource released by the MPQA project[4]. The resource provided a set of sequences which combine: words, word stems and POS tags which are known as clues. In each sequence there would be a word or stem (root of a word) which would have a pre-determined polarity (sentiment direction). The resource can be illustrated with a "clue" from the resource. The clue is:

[4]MPQA project could be found at: `http://www.cs.pitt.edu/mpqa/`

type=weaksubj len=1 word1=abandoned
pos1=adj stemmed1=n priorpolarity=negative

The clue contains the following information:

- Strength of the polarity of the clue.
- Length of the target (in words) of the clue.
- The target of the clue.
- The target POS tag. Depending on the context words may have different POS tags, for example, the word "receding" can be a: noun, verb or adjective depending upon the context.
- If the target is stemmed or not. If it is stemmed then the polarity clue can be applied to all words with the same root, rather than the single word.
- The polarity of the target word or stem.

The first entry in the example clue is type=weaksubj which indicates that the target of the clue has weak subjectivity. A weakly subjective clue describes a word which is subjective in some circumstances, whereas the alternate strongly subjective clue is subjective in all circumstances [Wilson, Wiebe, and Hoffmann, 2005]. The next entry in the clue is the len=1 which indicates that the target is one word. The next item in the sequence is word1=abandoned, which is the target of the "clue" and will be assigned a sentiment or polarity category. The next part of the sequence is pos1=adj which indicates that use of "abandoned" must be as an adjective rather than its verbal state. The next in the sequence is stemmed1=n which indicates that the polarity will be assigned to the word "abandoned" and not its stem. The last in the sequence is priorpolarity=negative which indicates that the polarity of the target word "abandoned" is "negative".

Another approach is to label a small amount of data into known categories and expand linguistic units from each category with external linguistic resources. Liu [2007a] suggests the following strategy: 1. *Construct separate seed sets for nouns, adjectives, verbs and adverbs from labelled data*, 2. *Label the nouns, adjectives, verbs and adverbs (positive, negative)*, 3. *The separate seed sets are grown by looking for synonyms and antonyms from WordNet until no new words can be added to the set* and 4. *Manually inspect the final set and remove any words with incorrect labels.* The expanded dictionaries can used in a rule based approach to label documents.

An alternative to using linguistic resources to increase the number of entries in a dictionary was proposed by Riloff and Weibe [2003]. She used extraction patterns with terms of known

polarity to label "high confidence" subjective sentences in unlabelled data. The information extracted from the labelled "high confidence" subjective sentences was used to create new extraction patterns which were used to label new "high confidence" subjective sentences. The process continued until no new sentences were labelled and no new information could be extracted from the unlabelled data. A further alternative was to use linguistic "hints" to propagate labels from words of known polarity to words with an unknown polarity. Hatzivassiloglou and McKeown [1997] used linguistic connectors to propagate a label from one word to another. Connectors are words which link one word with another [Petofi, 1988], for example: and, or, but, whereas, etc. The Hatzivassiloglou and Columbia strategy used a set of words with known polarity label. They extracted words which were connected to the known words by a connective. The extracted words were determined to be of the same polarity orientation as the known words by a log linear model regression model. The extracted words are assigned to a category by a clustering process.

5.2.2 Sentiment resources

It is often possible to use pre-built sentiment dictionaries to determine the sentiment orientation of a text. There are several dictionaries available, but this section will discuss two: SentiwordNet[Esuli and Sebastiani, 2006] and WordNetAffect[Strapparava and Valitutti, 2004]. The two sentiment dictionaries are based upon WordNet. SentiwordNet labels synsets[5] which are groups of related words that are synonyms of each other. SentiwordNet labels each synset of WordNet as either: positive, negative or objective. The labelling of a sysnet is achieved by a set of classifiers each trained on different data. WordNetAffect labels concepts which are subsets of synsets into negative, positive or neutral categories. The categories are based upon emotions, for example the word *anger* would indicate a negative category. WordNetAffect was built by manual labelling of a core set of sysnets whose labels were propagated into the remaining unlabelled synsets of WordNet through connections to the labelled synsets.

5.2.3 Semi supervised learning

There are many methods of semi-supervised learning (see review by Zhu [2008] for an extensive survey). The work described here will focus on a particular approach to this problem - *self-training*. Semi-supervised learning uses a mix of unlabelled and labelled data with the aim of propagating labels from the labelled data to the unlabelled data from which a robust classifier can be induced. Self-training propagates labels by first inducing a classifier from labelled data. The induced classifier classifies all the unlabelled data. The classifications which are *high confidence* are added to the labelled data. A new classifier is induced, and

[5]Synsets can also be referred to as synonym rings.

the process repeated until a stopping condition is met [Abney, 2007].

Classical self-training does not limit the unlabelled data the classifier can label and add as training data. Variations of self-training can limit the unlabelled data the classifier has access to, which can improve the quality of the induced model [Chang, Ratinov, and Roth, 2007]. Applying a constraint to the classifier can limit the amount of data the classifier has access to. There are two documented types of constraints: hard constraints such as voting or co-training [Abney, 2007] and soft constraints such as prior knowledge. A hard constraint explicitly limits the unlabelled data the learner can add as labelled data. Soft constraints do not explicitly limit the unlabelled data, but penalize the learner if it violates a pre-determined constraint. The literature search was restricted to using prior knowledge as a method of constraining the learner because it was thought that the bootstrapping strategies described in at the start of this subsection on page 100 could assist in identifying *prior knowledge* from labelled data.

Druck, Mann, and Mccallum [2008] used *common knowledge* as a soft constraint. They provided the example of classifying news stories into sports categories. They contended that the word "puck" would act as a soft constraint because it was a strong indicator of ice hockey stories, consequently a classifier which classified a news story with the term "puck" as anything other than "ice hockey" would be penalised. Chang, Ratinov, and Roth [2007] used a combination of constraints: unary, dictionary based and n-ary constraints to provide *better* training examples for a self training strategy. Chang, Ratinov, and Roth [2008] provided a framework for using constraints in a semi-supervised strategy which he called *Constrained Conditional Models*. Carlson, Betteridge, Wang, Jr., and Mitchell [2010] proposed the training of many classifiers each with different constraint strategies. The classifiers then act as constraints on each other. There are many published papers which describe the use of constraints for a particular task. The constraints used in these works are often derived with knowledge from outside of the data available for the classification task. There is a problem for this strategy in a domain where there is no external or common knowledge to create constraints.

5.3 Proposed Strategies

A quote or direct speech reported in text could be considered to be a sentence, or a group of sentences which may be susceptible to the sentence level strategy described in the previous chapter (see page 71). The sentence level strategy was not successful when applied to quotes which was due to the linguistic characteristics of the quotes for example, invented words and euphemisms. The failure of the sentence level strategy necessitated the construction of separate strategies to classify direct speech.

The following subsections describe two contributions of this thesis to semi-supervised clas-

sification of direct speech. The two strategies are Guided Self-Training and a contextual classification strategy. The Guided Self-Training strategy uses high precision rules to correct mistakes made by a classifier induced from initially labelled data whereas the contextual strategy relies upon separating the data into groups and applying a separate labelling and classification strategy to each group.

5.3.1 Guided Self-Training

The first proposed solution is *Guided Self-Training* (GST) which uses high precision rules to correct high confidence classifications by a base learner in a semi-supervised learning process. GST was initially designed to classify quotations, but it can be used as a general strategy for situations where *weak* models are induced from the initial training data. GST uses a high precision rule classifier (HPRC) to guide selections made by an induced model. HPRC guides the selection of candidates by labelling a subset of unlabelled documents which will be known as the *high confidence pool*. A HPRC labels a subset because high precision classifications are normally made at the expense of recall. This pool of high confidence classifications is used to limit the selection of examples in each iteration of a self-training (ST) process by testing the learner's labelling of high confidence candidates. The learner's label is altered if it is different to the label applied by the HPRC. GST accepts the learner's label: 1. if a classified document is not part of the high confidence pool or 2. a learner classifies a high confidence pool document with the same label as the HPRC. The newly labelled documents are then added to the labelled data and new iteration is started. This process continues until a specific stopping condition is met.

GST does not prescribe a method for constructing a high precision rule classifier. There are a number of possible rule classifiers, for example Weibe's high precision subjectivity classifier [Riloff and Weibe, 2003]. A selected rule classifier must have a high precision which typically comes at the expense of recall. A classifier which does not have high precision may change correct high confidence classifications by a base learner which may impair the training process.

The GST method is described in Algorithm 7. GST takes two main inputs: the labelled (LD) and unlabelled (UD) data sets. The outer loop (lines 3-26) represent the typical self-training iterations. The uniqueness of the proposal are the following:

- Documents classified by the base learner with a high confidence which are contrary to the high precision classification (the pool of high confidence candidates) are assigned to the high precision classification. These documents are assigned to the labelled data for training in the next iteration.
- The high precision classifier can abdicate (i.e no decision) and therefore high confidence candidates can be selected by the base learner with out the explicit agreement of the

high precision classifier.

Algorithm 7: Description of GST Candidate Selection Cycle.

Input: LD: Labelled Data
Input: UD: Unlabelled Data
Input: sThr: Minimum classification confidence for document to be added to LD
Input: Rule: A high confidence Subjectivity classifier which return a classification for a document
Input: Learner: the classification algorithm that is to be self-trained
Output: Learner: trained classifier

```
**Learn a classifier**;
Model ← Learner(LD);
repeat
    lClass ← Model.classify(UD);
    rClass ← Rules.classify(UD);
    CD ← {};
    AD ← {};
    forall the d ∈ UD do
        if lClass.confidence[d] ≥ sThr then
            UD ← UD \ d;
            if rClass[d] ≠ NULL and rClass[d] ≠ lClass[d] then
                CD ← CD ∪ {< d, rClass[d] >};
            else
                AD ← AD ∪ {< d, lClass[d] >};
    count ← Count(CD);
    if count == 0 then
        count ← Count(AD);
    TD ← ReturnRandomDocs(AD, count);
    UD ← UD ∪ (AD \ TD);
    LD ← LD ∪ CD ∪ TD;
    **Get a new model**;
    Model ← Learner(LD);
until No more new candidates;
return Model;
```

In summary, the GST algorithm extracts features (words) from manually labelled data to create a list of words. The list of words are expanded by using an external linguistic resource (WordNet) to form a new enhanced list of words. A high precision rule classifier uses the

enhanced list of words to "correct" classifications made by a base learner trained on the original labelled data in iterative self-training process.

5.3.2 Contextual Classification Strategy

The contextual strategy relied upon an assumption about the data. The assumption was that there were two groups of people whose comments are reported in the mass-media and that their motivation for speaking differ. One hypothetical group will be known as the "biased group" and the other group will be known as the "unbiased group". Members of the biased group will most of the time have quotes that are expected and thus have no effect on the market. This means that if quotes of people with job title A most of the time do not effect the market then job title A must belong to the biased group. On the contrary if the majority of quotes of people belonging to job title B, have some effect (positive or negative) on the market, then job title B must belong to the unbiased group.

The discovery of the job titles associated with each group was achieved by manually aligning a set of quotes with a market. The job titles of the respective speakers were determined by Open Calais. For each job title we calculated the proportion of quotes aligned with market movements. If this proportion was found higher than 50% the job title was assigned to the biased group. Conversely, if the proportion of the quotes aligned with no market movement was the higher than 50% the respective job title was assigned to the biased group. An example of the discovered job titles are in Table 5.2. The two resulting sets of job titles were used as a decision rule to determine if a quote made by a speaker belongs to either the biased or unbiased group. If the job title of this speaker does not belong to either of the groups the quote is by default assigned to the biased group.

Biased	Non-Biased
Chairman, CTO, Co-head, Company President	Chief Economist, Credit Analyst, CBI chief economic adviser

Table 5.2: An example of job titles assigned to a group.

An overview of the contextual classification strategy is presented in Figure 5.1. The contextual strategy has three separate classifiers. Two classifiers for quotes assigned to the biased group, and one for quotes assigned to the unbiased group. The first classifier for the biased group classifies the quotes into: expected or surprising categories whilst the second classifier classifies the "surprising" quotes into positive or negative categories. The classifier for the unbiased speakers classifies quotes into positive and negative categories.

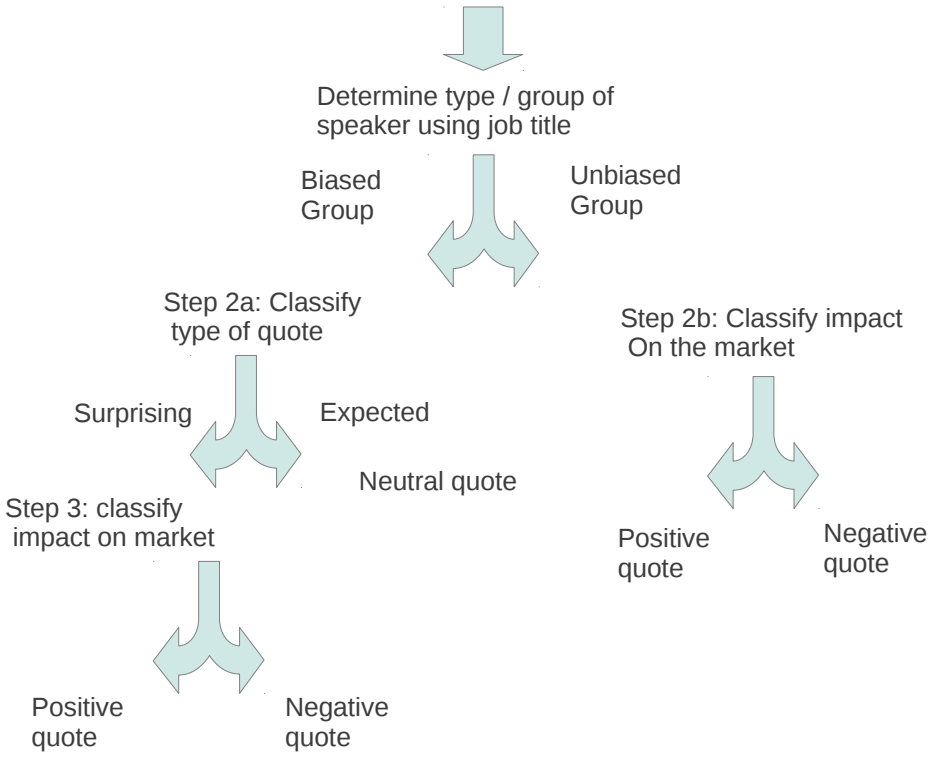

Figure 5.1: Overview of the contextual classification strategy

The first step is to train a classifier to classify a quote as either "surprising" or "expected". This was achieved by manual alignment of quotes with market movements. The annotator selected a quote at random, identified a stock ticker associated with the company which was associated with the speaker. The annotator verified the stock price on the day on the quote as made. If the stock price changed sufficiently to satisfy the annotator that the quote had induced the movement then the quote was labelled as "surprising" otherwise it was labelled as "expected".

The small amount of labelled quotes was expanded with a simple form of label propagation which clustered the labelled quotes with the larger amounts of unlabelled quotes. A quote was represented as a "bag of words". The labelled quotes were clustered with separate batches of unlabelled documents. Each batch contained 1000 labelled quotes. The batches were limited to this number to ensure that the clustering process was conducted in a timely manner. The clustering method used in this process was the RapidMiner "Top Down Clustering operator" [Mierswa, Wurst, Klinkenberg, Scholz, and Euler, 2006]. The 'Top Down Clustering operator" was used initially to see if there was any relationship between quotes in clusters

at differing levels. The depth of the clustering was the default of 5, no pre-set number of clusters was set. The labels from the labelled quotes in a cluster were propagated to the unlabelled data if: 1. a single class of labelled quotes accounts for 75% of quotes in a single cluster and 2. there are no quotes from the other class in the cluster. It is accepted that the size of the clusters will directly effect the performance of this technique, but the described experimental set up was satisfactory for the described experiment.

The next step is to train a new classifier to separate quotes which have been classified as surprising into positive and negative categories. This was achieved by aligning the documents labelled in the previous step as surprising with price rises or falls in a given stock price. The aligning process was conducted manually. The annotator had the final decision whether the market movement inferred that the quotes was positive or negative. A new classifier was trained with the labelled quotes. At this stage, there were two trained classifiers: 1. a classifier which had been trained with expected and surprising data and 2. a classifier which had been with trained with negative and positive quotes.

The last step was to create a decision rule for the members of the unbiased group. The process was similar to the high precision subjectivity classifier described by Riloff and Weibe [2003]. The quotes which are to be used as training data for a classifier are discovered by the following process: 1. manually label a number of quotes made by members of unbiased group, 2. manually select sentimental words from the labelled data, 3. expand extracted words with WordNet, 4. separate words into positive and negative categories, 5. use a decision rule to label unlabelled quotes made by members of the unbiased group with a label (positive or negative), and 6. train a new classifier with the quotes labelled from this process.

The strategy can be used as a whole, however it is possible to use the trained classifiers separately. The motivation of using the classifiers separately would be to measure the predictive capability of either the unbiased or the biased group of speakers. These variations of the contextual strategy are evaluated in Chapter 6.

5.3.2.1 Experiments with Direct Speech

This section describes some initial experiments with direct speech. The experiments are shown to support the hypotheses made on page 107.

The first experiment was designed to provide evidence that there existed two groups of speakers in the quote collection, and that these groups had separate lexicons which could be identified through lexicon analysis. The two groups will be known as either: 1. biased or 2. unbiased. The biased group will contain members who are compelled to speak in a misleading manner and on occasion are compelled to speak truthfully. The unbiased group will contain members who are compelled to speak truthfully.

The initial experiment analysed the lexicon of hypothesized members of the biased and

unbiased groups. The members were selected by their job role: 1. CEO (biased group) and 2. analyst (unbiased group). The experiments used the quotes published between October 2008 and September 2010. In this group of quotations there were 54,845 quotes attributed to "analysts" and 38,089 quotes attributed to CEOs. The experiment extracted adjectives from quotes because adjectives are known to be the conveyors of the opinionated language [Wiebe, Wilson, Bruce, Bell, and Martin, 2004]. The affinity of an adjective with either the CEO group of quotations or analyst group of quotations was calculated with Pointwise Mutual Information (PMI) as follows,

$$PMI(adj, cl) = \log_2 \frac{Pr(adj, cl)}{Pr(adj)Pr(cl)}. \tag{5.3}$$

where *adj* represents an adjective, Pr represents a probability, and *cl* is the job role of the speaker.

Adjectives which were assigned a score greater than zero when calculating their affinity with class "CEO" were assumed to be part of the CEO lexicon. The experiment produced a list of 1,401 adjectives. The adjectives were ranked by PMI score in descending order, consequently adjectives which had a strong affinity with the CEO's lexicon had a higher rank than adjectives which had a weaker affinity with the lexicon. An inspection of the CEOs adjective lexicon revealed that the majority of the adjectives were *positive*. The positive adjectives typically had higher ranks than the negative adjectives. The first negative adjective was ranked 87th. The positive adjectives were overly positive, for example, 'superb" and "immense', whereas negative adjectives were not exaggerated. There were a number of jargon / domain specific adjectives in the CEO's lexicon, for example, 'win-win" and "mission-critical". A frequency analysis of unigrams in CEOs' quotes was made. The frequency analysis did not consider stop words. A manual inspection was made of the 100 most frequent and infrequent unigrams. The most frequent unigrams consisted of mainly positive unigrams whereas the infrequent unigrams were: negative, spelling errors or invented words. In summary, the lexicon of the CEO is overwhelmingly positive, which is, nevertheless a contradiction because the quotes were harvested between 2008 and 2010, which was a time of a severe economic crisis.

The same experiment was repeated for quotes which were made by analysts. A PMI score was assigned to adjectives which were present in quotes made by people with the job title "analyst". The PMI score was calculated as per Eq.5.3 for each adjective. Adjectives with a PMI score of 0 or less were removed. The experiment realized 415 adjectives, consequently the analyst's adjective lexicon was 70% smaller than the CEO's lexicon, although there were more quotes attributed to analysts than to CEOs. The adjectives were ranked in descending order of PMI score. The first negative adjective, "*speculative*", was ranked 2nd, conversely the highest ranked negative adjective in the CEO's lexicon was ranked 87th. In addition the negative and positive adjectives were not exaggerated. A frequency analysis

failed to find a division between positive and negative adjectives because both positive and negative adjectives appeared in frequent and infrequent unigram groups. In summary, there is clear evidence that there are significant differences in the lexicons of CEOs and Analysts. The lexicon analysis provides evidence for the initial hypothesis that people with connections to companies (biased group) will use language similar to rhetoric[Cyphert, 2010] and lying [Larcker and Zakolyukina, 2010] whereas *independent* people (unbiased group) will use more measured language.

The second experiment evaluated the advantage of using: 1. a speaker's group membership and 2. unigrams in the quotes, as features. This was compared against the following features: 1. speaker's job title and quote unigrams, and 2. quote unigrams. The data was hand labelled into three classes: positive, negative and neutral. There were 155 labelled quotes. A 10 times 10 fold cross validation as implemented in RapidMiner [Mierswa, Wurst, Klinkenberg, Scholz, and Euler, 2006] was used to calculate an average accuracy measure. A Naive Bayes classifier was used in the evaluation. The results are in Table 5.3. The results for this experiment demonstrate that the addition of a "job role" feature is detrimental to performance whereas the addition of a speaker's group membership provides a small advantage over the experiment which used only unigrams.

Classifier	Features	Avg. Accuracy
Naive Bayes	Unigrams	0.64 ±0.01
Naive Bayes	Unigrams + JR	0.39 ±0.01
Naive Bayes	Unigrams + OMR	0.72 ±0.010

Table 5.3: Estimated Accuracy from 10 times 10 fold cross-validation. JR= Job Role, OMR = Opinion Maker Role (Biased or Unbiased) and Avg. = Average.

5.4 Comparison of the two direct speech classification strategies.

This section will describe experiments which provide an evaluation of the proposed strategies. The first experiment evaluated the complete contextual strategy, and Guided Self Training (GST), against several competing strategies on a separate labelled set of hand labelled documents. A new set of documents was used for this experiment to ensure that there was no bias in the set of documents (used in the previous experiment) towards the contextual strategy.

The experiment used 1322 hand labelled quotes which were classed as either: negative, neutral or positive. 32% of the quotes were made by members of the unbiased group and 68% of quotes were made by either: members of the biased group or unidentified speakers.

The contextual strategy classified both quotes from members of the biased and unbiased groups.

GST relies upon a high precision rule classifier (see page 105). There is no recommended rule classifier, but in this experiment Weibe's [Riloff and Weibe, 2003] high precision subjectivity classifier was chosen because previous experiments have found that it has a higher precision than competing rule classifiers [Drury, Torgo, and Almeida, 2011].

GST and the contextual strategy were evaluated against baseline strategies that had been used in initial attempts to classify direct speech. The strategies were:

- Inductive: An inductive strategy induces a classification model using only the labelled data [Abney, 2007].
- Self-Training: An iterative process where at each step a model is induced from the current labelled data and it is used to classify the unlabelled data set. The model assigns a confidence measure to each classification. If the classification confidence measure is greater than a predefined threshold then the respective unlabelled cases are added to the new iteration training data with the classifier assigned label. At the end of the cycle the learner is trained on the new labelled data set. This cycle continues until a stopping condition is met [Abney, 2007]. To ensure an equitable comparison the stopping condition for the self-training variants and GST were 50 iterations.
- Voting strategy: is a variant of GST and uses two classifiers, Weibe's high precision rule classifier and a learner. A document is added to the labelled data if the rule classifier and learner agree.
- Veto strategy: is a variant of GST and uses two classifiers, Weibe's high precision rule classifier and a learner. A document is added to the labelled data if the rule classifier and learner disagree. The label is applied by the rule classifier.

The individual strategies labelled a set of documents which was used to train a Naive Bayes classifier. Each strategy had 10% of the labelled data to either: 1. construct dictionaries and 2. train classifiers. Each strategy had access to 384296 unlabelled quotes. The evaluation was a "holdout" evaluation which used Lingpipe's [Alias, 2008] F-measure calculation. Lingpipe calculates a micro average F-measure [6] across all the classes[7]. The calculation of the F-measure was on the 1190 labelled quotes that the strategies did not have access to in the training phase. The results are in Table 5.4.

The contextual and GST strategies outperformed the competing baseline strategies. The difference between the contextual strategy and the nearest competitor was small, but the strategy has a significant advantage over the remaining strategies.

[6] see page 91

[7] More information can be found at `http://goo.gl/DvMi3`

Strategy	Estimated F-measure
Inductive	0.35
Self-training	0.31
Voting	0.51
Veto	0.13
Contextual	0.53
GST	0.58

Table 5.4: Estimated F-measure for competing strategies.

The previous experiment used a small amount of manually labelled documents, consequently the selection of the quotes may not have been an accurate representation of the quote collection. An additional set of experiments were conducted for GST on linguistically similar domains where all the data was labelled. The experiments were conducted for GST only, because unlike the contextual strategy it did not rely upon insights into the data. The additional experimentation was conducted on user generated reviews for 1. airline meals [airlinemeals.net, 2010], 2. university lecturers [ratemyprofessors.com, 2010] and 3. music concerts and records [reviewcentre.com, 2010]. The domains were chosen because reviews are often written in an informal manner similar to speech, consequently the domains shared the following characteristics with the direct speech collection: 1. invented words, 2. slang, 3. profanity, 4. non standard spelling and grammar, 5. multi-word expressions and 6. non standard punctuation.

The rating was taken as an indication of the polarity of the review. The criteria for class assignment is described in Table 5.5. Documents not satisfying the criteria for class assignment were removed from the experiments. The resulting labelled data sets were used to compare: 1. Two separate base learners (Naive Bayes and Language Models) and 2. Alternative strategies. The alternative strategies were: 1. Inductive (LD) which was a classifier trained with only labelled data (LD), 2. Self-Training which was initially a learner trained with labelled data which classified the unlabelled data and added "high confidence" classifications to the labelled data. The labelled data was used to induce a new classifier. This process continued until a stopping condition was met., 3. Voting, which used the high precision rule classifier used by GST and a learner in a self-training process. Unlabelled data was added if the learner and the high precision rule classifier agreed on their classifications., 4. Inductive (LD and RC) this strategy trained on the labelled data and the data selected by the rule classifier (RC) used in GST., and 5. Veto strategy which used GST's rule classifier and a learner in a self-training process. An unlabelled document was only added if the rule classifier and learner disagreed. The label applied in this strategy would be that of the rule classifier.

Domain	Positive Category	Negative Category
Airline Meals	4 -5 Stars	1-2 Stars
Teacher Reviews	"Good Quality"	"Poor Quality"
Music Reviews	4-5 Stars	1-2 Stars

Table 5.5: Polarity Criteria.

The experiments used a micro average F-measure. The estimated F-measure experiments used increasing larger random selection of documents as training data. The smallest selection of data was 1% of the total and the largest 5%. The increments were in steps of 1%, for example the second iteration of the experiment was 2%, the third 3% etc. At each iteration the experiment was repeated 20 times, for example the 1st iteration there would be 20 random samples of 1% and 20 estimations of F-Measure estimated in a "holdout" evaluation. The test set for the evaluation was the total domain less the data selected for training. For example, if 1% of the data was used for training than the remaining 99% would be used for testing.

An overview of the process is the following: 1. randomly select training data (the LD set in Algorithm 7) and 2. hide the labels of the remaining documents to create the UD.

The *Airline Food Domain* results are presented in Figures 5.2 and 5.3. The results demonstrate a clear advantage for the proposed strategy for both classifiers. The results demonstrate a significant gain in F-Measure at the 2% of domain for training for both classifiers. The gain in F-Measure halts at the 3% of domain for training. The two inductive strategies gain F-Measure as training data increases.

The *Teachers Domain* results are presented in Figures 5.4 and 5.5. The results demonstrate a clear advantage for the proposed strategy. In common with the airline food domain the Guided Self-Training(GST) shows a large gain in F-Measure at 2% of domain for training. The gain in F-Measure is more pronounced for language models. GST demonstrates a reduction in F-Measure with further increases in training data. The reduction in F-Measure is within the mean standard deviation. The inductive strategies in common with the airline food domain gains F-Measure with increases in training data. The self-training strategy gains in F-Measure increase with training data, but at a faster rate than the inductive strategies. The voting schemes also demonstrate a gain in F-Measure, but at a lower rate than the inductive and self-training strategies.

The *Music Review Domain* results are presented in: Figures 5.6 and 5.7. The results demonstrate that the proposed strategy shows a distinct disadvantage over the competing strategies. The lack of improvement achieved by GST could have been due to the robustness of the initial models. Robust models would not have made "obvious errors" which would have been corrected by the rule classifier. A lack of corrections may have accounted for GST's lack of improvement over the competing methods.

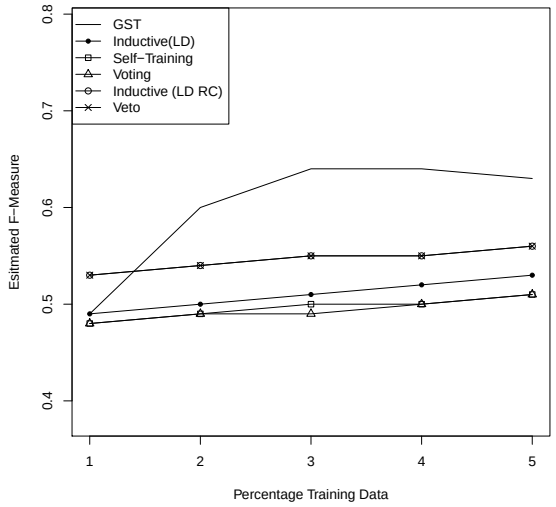

Figure 5.2: Estimated average F-measure for each strategy using Language Models as a classifier for airline meals reviews. LD = labelled data and RC = Rule Classifier Selected Data.

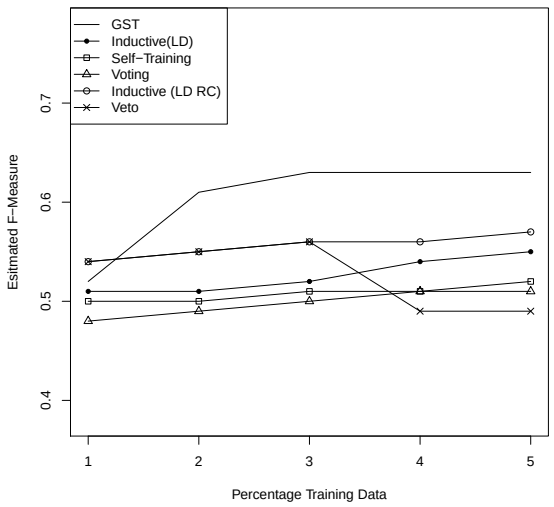

Figure 5.3: Estimated average F-measure for each strategy using Naive Bayes as a classifier for airline meals reviews. LD = labelled data and RC = Rule Classifier Selected Data.

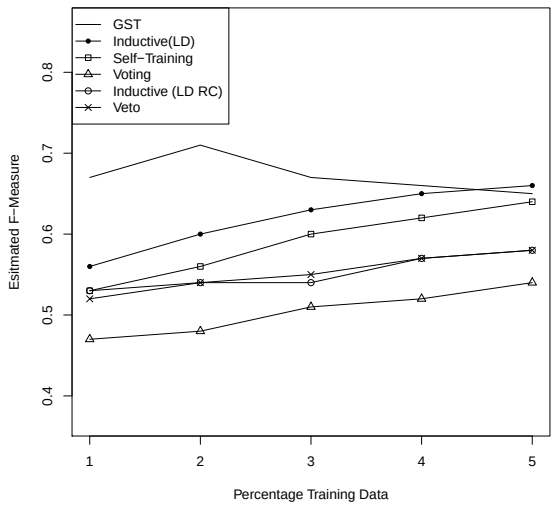

Figure 5.4: Estimated average F-measure for each strategy using Language Models as a classifier for teacher reviews. LD = labelled data and RC = Rule Classifier Selected Data.

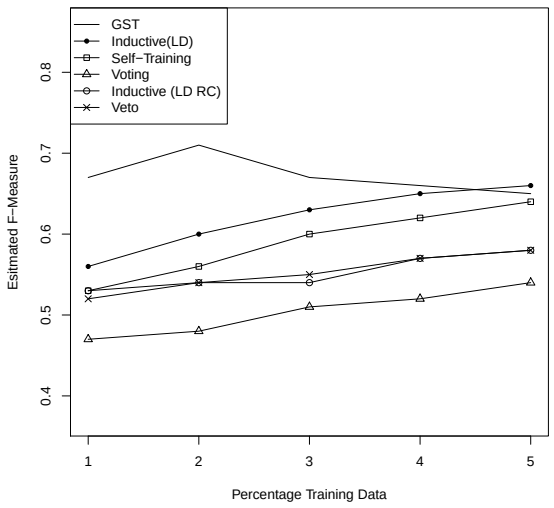

Figure 5.5: Estimated average F-measure for each strategy using Naive Bayes as a classifier for teacher reviews. LD = labelled data and RC = Rule Classifier Selected Data.

The experiments in the Teacher and Airline meal domains show a peaking of F-measure,

and a decline in F-measure. The decline in F-measure with increasingly larger training data may be due to the more robust models making less mistakes to correct. It may have been possible that the HPRC "corrected" valid classifications which would have induced a worse performance than if the HRPC had made no corrections.

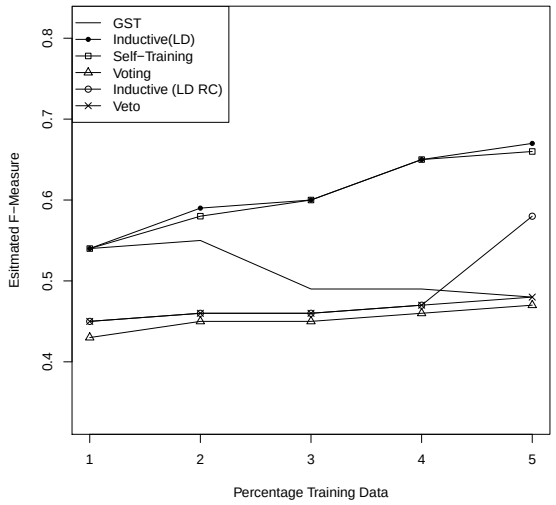

Figure 5.6: Estimated average F-measure for each strategy using Language Models as a classifier for music reviews. LD = labelled data and RC = Rule Classifier Selected Data.

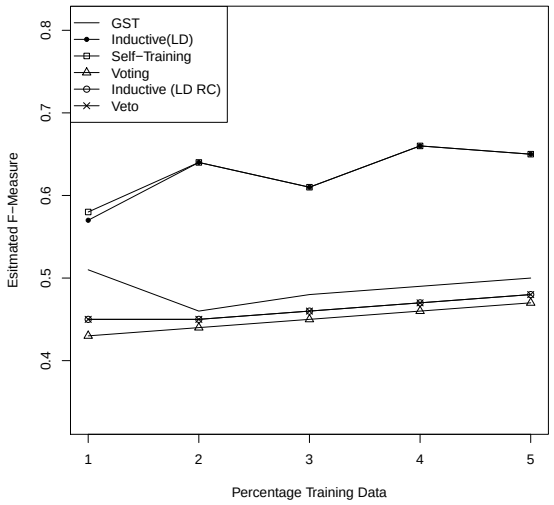

Figure 5.7: Estimated F-measure for each strategy using Naive Bayes as a classifier for music reviews. LD = labelled data and RC = Rule Classifier Selected Data.

The full results for this experiment are available in the appendix on page 160. The results for strategies which had access to rule selected data frequently have a higher precision measure, but this improvement is frequently at the cost of lower recall. For example the mean recall and precision for the voting strategy in the Airline Food domain was 0.5 and 0.7, where as the inductive strategy yielded recall and precision of: 0.51 and 0.62. A possible explanation for this phenomenon is the fact that the high precision classifier may only classify a very specific sample of documents. The addition of these documents labelled by the high precision classifier to the initial data set of the models could be biasing the classifier towards learning very specific rules, which may negatively impact on recall, but may boost precision. The GST method does not suffer from a decrease in recall. A possible explanation could be the high precision classifier is being used with a different purpose within GST when compared to the "(LD+RC)" learners. In the GST strategy a high precision classifier is used to supervise the classifications of a standard base learner with the goal of avoiding very obvious mistakes. In the "(LD+RC)" learners the rules are used to add more labelled data to the training set available to the learners. These are two different uses of the high precision classifier and our experiments clearly provide evidence towards the advantage of our proposal. In effect, GST

Manually Labelled	Quote Classification Strategy
Manually Labelled	0.03±0.41
Contextual Strategy	0.03 ±0.34
GST	-0.07±0.61

Table 5.6: Results for single day's news value.

improvement in precision is not offset by a drop in recall.

5.4.1 Single news value strategy evaluation

As stated on page 99 the aim of the quote classification strategies was to produce a single value. The aim of this evaluation was to ensure that a day's worth of classified quotes produce a similar value to a manually annotated day's worth of quotes.

The evaluation used a company specific ontology news retrieval system (see page 63) to recommend news for "Microsoft" for the date 20-06-2009, which was randomly selected for the evaluation. The news retrieval system recommended 64 news stories. The quotes were extracted from each news story, and were hand scored. There were in total 289 quotes. A value of 1 was assigned to a positive quote, 0 to a neutral quote and -1 to a negative quote. A mean quote news value was calculated for the day. Both strategies were used. The results are documented in Table 5.6. The manually labelled quotes indicate that the day was marginally "positive". The contextual quote classification strategy also returned the same value as the mean manually labelled value, but the standard deviation was different. GST returned a marginally negative average document score.

The selected day was broadly neutral, therefore a large number of the quotes were labelled with a value of 0. It was possible that the strategies may have achieved these scores by chance, for example, a strategy would have achieved a similar score if it had marked half the documents as negative and the other half as positive. In this example, the average score would have been near to zero which would have been close to the hand labelled score, but with a small number of quotes' sentiment direction being estimated correctly. A complimentary evaluation was conducted where the accuracy of the sentiment direction estimation was measured. In this circumstance accuracy is a number expressed as a percentage of the manually labelled quotes estimated correctly by the contextual strategy and GST. The results are documented in Table 5.7. The results reflect the difference in single day's news value with the contextual strategy achieving a significantly higher accuracy than GST.

Strategy	Accuracy (%)
Contextual	81.31%
GST	56.05%

Table 5.7: Accuracy for a single day's quotes.

5.5 Summary

This chapter presented work which was intended to classify direct speech reported in the mass media. Direct speech is a difficult domain to work with because of the linguistic complexity of direct speech and the lack of labelled data. GST and the contextual strategy were specifically designed to classify direct speech, however GST can be used as a general strategy to classify texts in domains with small amounts of labelled data.

The two approaches are novel classifications strategies and are a contribution of the thesis to the field of sentiment analysis. The contextual strategy is a novel contribution because it uses speaker motivation and its effect upon the speaker's lexicon to classify text. The research literature contains a small number of works which have attempted to classify direct speech, and these works did not consider speaker motivation in their classification approach. In addition, the approach separated the speakers by motivation and applied separate labelling and classification strategies. This part of the strategy seems to be unique in the field of sentiment classification.

GST is a novel contribution to the field of semi-supervised learning with soft constraints. The novelty of this strategy is that it generates its soft constraints from the labelled data which are then expanded with linguistic resources such as WordNet. The articles discovered in the research literature use soft constraints from outside of the data and are typically based upon "common knowledge". The advantage of GST is that allows a human to be ignorant about the domain he is working with, but still be able to use soft constraints.

The strategies presented on this chapter used either: 1. small manually labelled data sets or 2. fully labelled substitute domains to evaluate each strategy. In these evaluations the proposed strategies had an advantage over the competing ones, but these advantages may not necessarily translate into a trading advantage. The experiments will be evaluated for any trading advantage later on in thesis. The work presented in this chapter does indicate that in the direct speech domain that these proposed strategies have a measurable advantage over competing strategies.

Chapter 6

System Experiments

The thesis thus far has described the individual parts of the system. The individual parts have included: web crawling, text extraction, information retrieval and text classification. At each stage the component part of the system has been evaluated. The evaluation presented so far has not considered the impact of each part of the system upon trading. This chapter presents two trading evaluations : 1. the sentence, document and quote classification strategies presented in Chapters 4 and 5, and 2. the complete system.

The evaluation of the classification strategies used a market index (FTSE250) to compare the returns (points difference) of each classification strategy. In addition to comparing each classification strategy the evaluation allowed the selection of one classification strategy at the: 1. sentence, 2. document and 3. quote levels to use in the complete system evaluation. The complete system was the following: 1. an information retrieval strategy (IRS) which selected news stories for a specific company, and 2. a classification strategy which classified the news stories selected by the IRS. The values garnered in step two were then added as feature(s) to an existing stock trading system, which then used the information to trade with the goal of checking the impact of this extra information on the results of the system.

The layout of the chapter will initially concentrate upon the trading evaluation of each individual classification strategy and their associated variants. The trading evaluation will explain the experimental set-up, and the associated configurations. The experiments which use a Naive Bayes classifier will present the results by classifier confidence. This was to demonstrate the effect, if any, of classifier confidence on the trading results.

The chapter will then concentrate on the company specific trading experiments. The hypothesis of this thesis was that adding news features to a standard quotes only trading system would improve its performance. To test this hypothesis the evaluation compared the stock trading system's performance with technical indicators and technical indicators with news feature(s). In addition, the two competing information retrieval systems documented

in Chapter 3 were evaluated. Finally the evaluation considered varying combinations of classifications strategies. The chapter will then summarize the experiments and draw a brief conclusion.

6.1 Classification strategies trading evaluation

The evaluation of the classification strategies was designed to show their ability to trade successfully. A decision was made to use a market index because a market index value incorporates all known information (including news information) in its value. A further decision was made to classify news stories published when the market was closed and trade when the market opens the next day, consequently the market would only react to these news stories when it opened. The difference may be reflected in the closing value which is on the same day as market opening. To improve our chances of detecting this reaction a market index needed to be chosen where there was minimal pre-market trading. Pre-market trading[1] is trading when the market is closed. These trades are executed as the market opens, and the impact of the news is often reflected in the market opening price. In the trading data available the FTSE 250 had minimal pre-market trading and therefore was chosen for the evaluation. There was one evaluation measure, points difference. Market values are measured in points. Points difference is the difference between market opening and market closing values. In the evaluation this value is the absolute value of the points difference. It is added or subtracted if a trade was executed correctly, therefore a strategy which returned a positive points difference could be deemed as being profitable whilst a negative value indicates a strategy had returned a loss.

The competing strategies were evaluated using a Monte Carlo Simulation. A Monte Carlo simulation involved splitting of the time which was covered by the news story corpus into 20 randomly selected divisions of 400 contiguous days. News stories published in the first 250 days of randomly selected block of time were used as "training data" and news stories which were published in the last 150 days were used as "testing data".

The three competing classification strategies were compared in terms of their ability to correctly drive the trading decisions. Trading decisions for each of the days in the testing periods are made before the market opens and are based on the outcome of the application of one of the three alternative classification strategies to the news of the previous day. Namely, each of the strategies is used to derive a single aggregate value for these news that will guide the decision on which trading action to carry out. If the aggregated score is higher than zero a buy order to open a long position is issued to be carried out when the market opens. If the aggregated score is below zero a sell order is issued to open a short position. Both positions will be closed at the end of the day, with the difference between the closing and opening

[1] See `http://en.wikipedia.org/wiki/Extended_hours_trading`.

prices in that day determining the points difference associated with the position.

The three classification strategies belong to two types of approaches to text classification. The sentence level method is a purely linguistic strategy, while the document and quote methods are based on supervised learning algorithms that require a set of labelled examples (in one case documents and in the other quotes). These two different approaches have an impact on how the respective classifiers use the available training data. An overview of the process followed on each of the 20 random train+test divisions is presented in Figure 6.1. The sentence level strategy uses the training data to construct linguistic resources such as dictionaries. The document and quote level strategies use the training data to obtain a labelled training set that is then used to train a Naive Bayes classifier.

On each day in the test period a trading decision is made based on the outcome of the classification strategies. This outcome is an aggregated score of the news of the previous day. The way this aggregated score is obtained is different for the three strategies. The sentence level classifier proceeds as follows:

1. Split the news stories of the previous day into sentences.
2. Use the linguistic rules derived from the training data to assign a score to each of these sentences.
3. Sum up the scores of the sentences to obtain the aggregated score value.

The document level and quote classifiers use the following strategy:

1. Transform the news stories into testing instances. In the case of document level each story originates a single instance using the bag of words representation. For the quotes classifier the stories are searched for quotes and each of these is transformed into a testing instance again using a bag of words representation.
2. Use the classification models learnt from the training data to obtain the set of probabilities of each instance belonging to the classes: negative, neutral or positive.
3. Decide the classification of each instance using the set of probabilities and a user-defined parameter that sets the minimum confidence required for this decision. For each instance if the class with highest probability is above this minimum confidence value the instance is given that class, otherwise the instance is not classified (i.e. the classifier abdicates).
4. Map the assigned class into a numeric score according to some user-defined function (e.g. negative into -1, neutral into 0 and positive into 1).
5. Sum up the scores of all classified instances to obtain the aggregated score value.

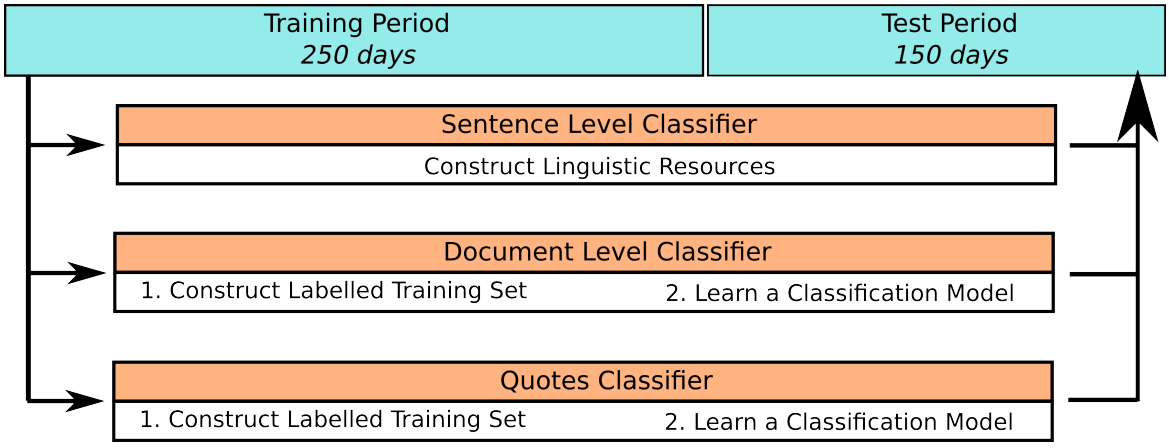

Figure 6.1: An overview of classification strategies.

6.1.1 Sensitivity analysis of the inductive classifiers

The use of the supervised learning algorithms (document level and quotes) requires several decisions to be made. In this section we study the impact on the results of using different values for these decisions. Namely, we explore: 1. different alternatives for constructing the labelled training set, 2. different values for the minimum required confidence of classifications and 3. classifying different parts of a news story, which in this circumstance was story text and story headings (headlines).

The different strategies considered for obtaining the labelled training sets were fully described in Chapters 4 and 5. In this sensitivity study the following alternatives were considered:

Ev - labels are obtained by using event rules.

Ma - labels are obtained by using market alignment.

MaEv - labels are obtained by using market alignment and event rules.

MaSe - labels are obtained by using market alignment and sentiment rules.

MaSeEv - labels are obtained by using market alignment with sentiment and event rules.

MaEvSt - labels are obtained by using market alignment with event rules and self-training.

MaSeEvSt - labels are obtained by using market alignment with sentiment, event rules and self-training.

Se - labels are obtained by using sentiment rules.

SeEv - labels are obtained by using sentiment and event rules.

The sensitivity of the classifier confidence parameter was evaluated. The results for the document level strategies are documented in Figure 6.2. The relationship between classifier confidence and points difference seems unclear as there is no consistent increase or decrease in points value as classifier confidence increases.

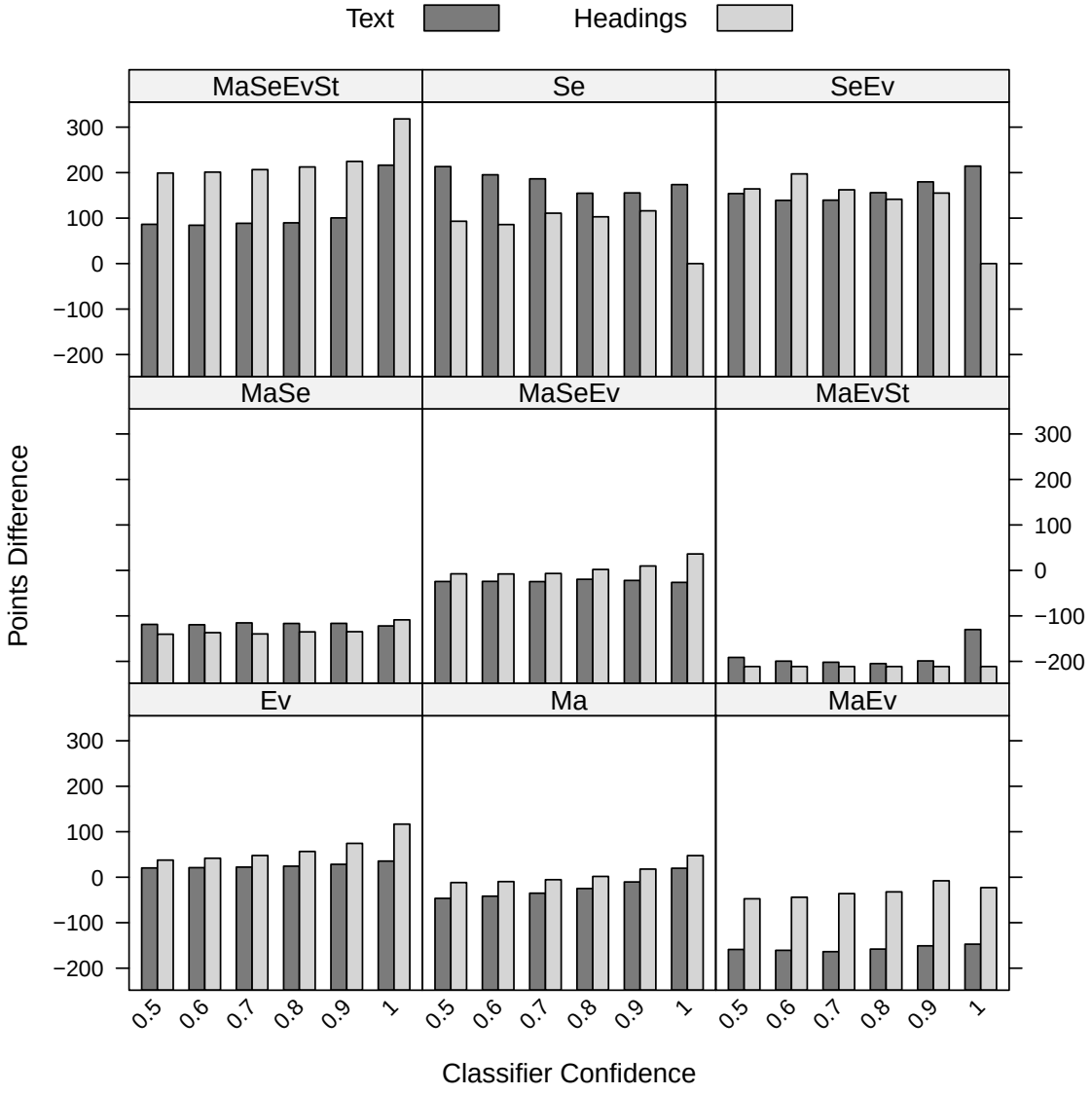

Figure 6.2: Points difference for different classifier confidence levels. Ev = Event rules, Ma= Market alignment, MaEv = Market alignment with event rules, MaSe = Market alignment with sentiment rules, MaSeEv = Market alignment with sentiment and event rules, MaEvSt = Market alignment with event rules and self-training, MaSeEvSt = Market alignment with sentiment, event rules and self-training, Se = Sentiment rules and SeEv = Sentiment and event rules.

The effect of classifier confidence on trading returns (points difference) on quote classification

strategies is documented in Figure 6.3. In common with the document level strategies the relationship between classifier confidence and points difference is unclear.

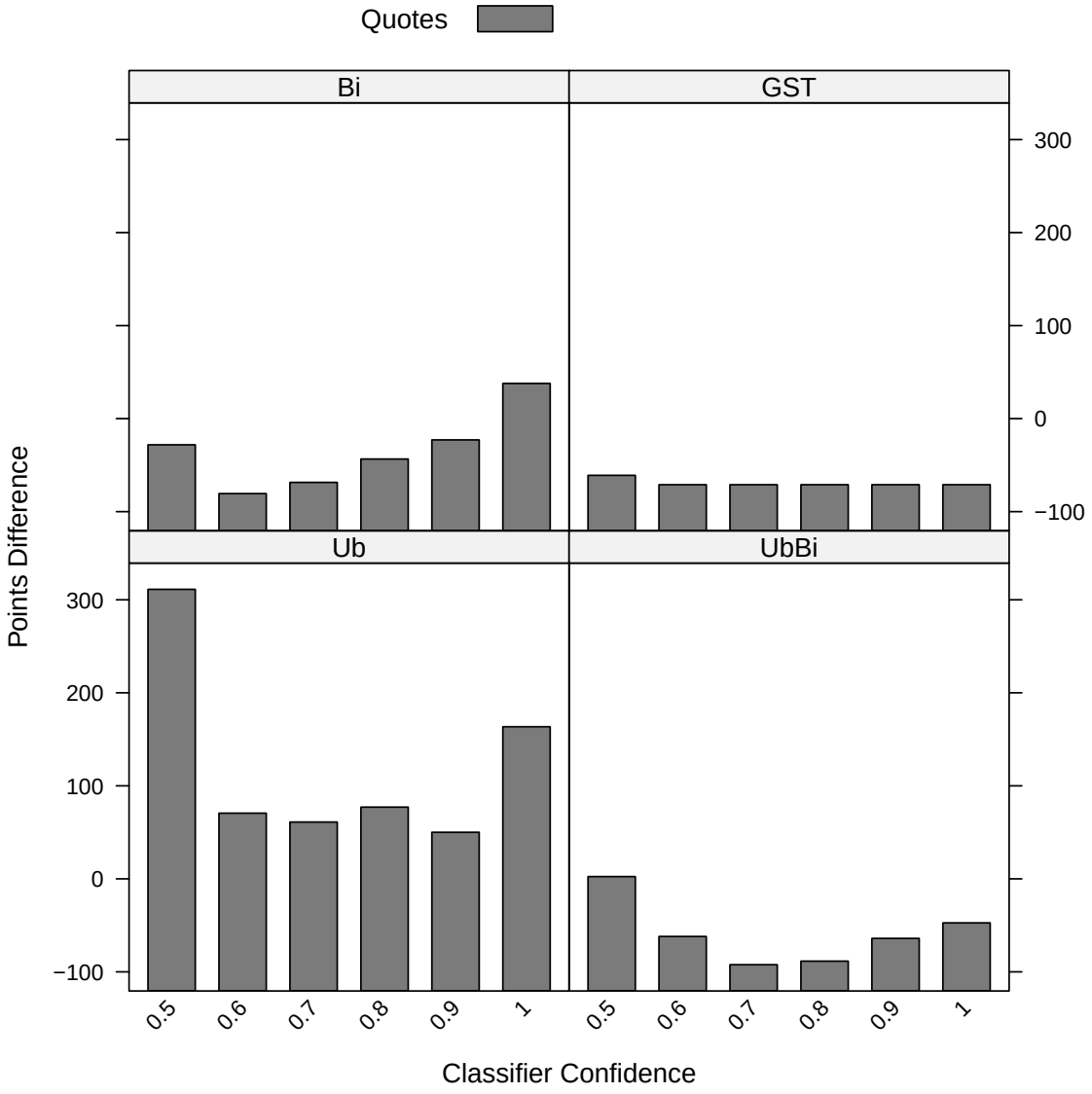

Figure 6.3: Points difference for different classifier confidence levels for quote level classifications. Ub= Contextual strategy, members of unbiased group, UbBi = Contextual strategy, members of unbiased and groups, Bi=Contextual strategy, members of biased group, GST = Guided self-training.

In summary, although the graphs show some variations in points difference for each parameter tested, it was not possible to identify consistent relationship between a parameter and points difference.

6.1.2 Sentence level evaluation

The sentence level strategies were evaluated with the methodology described earlier. Sentence level strategies are linguistic and therefore do not have a classifier confidence. The evaluation used the same 20 divisions for the Monte Carlo simulation as per the previous experiments. The training part of the division was used to construct linguistic resources. The difference between the linguistic and inductive strategies in this evaluation is described in Figure 6.1. The evaluation, again, was by points difference. The results are documented in Table 6.1. The results demonstrate that the use of both sentiment and event rules outperformed the individual use of sentiment or event rules.

Strategy	Text	Points Difference
Event	ST.	73.78 ±315.02
Event	Head.	299.04±262.09
Sent.	ST.	-421.13 ±607.05
Sent.	Head.	-143.71±346.40
Sent. & Event	ST.	719.74 ±419.29
Sent. & Event	Head.	341.63 ±489.41

Table 6.1: Sent.= sentiment, ST. = story text and Head.= headline.

6.2 Full system trading evaluation

A full system is a combination of the constituent parts which have been described in the preceding chapters. A full system therefore performs the following steps: 1. scrape and extract news from the web, 2. select and score news stories for a specific company or market index, 3. classify a series of news stories at either the: document, quote or sentence level and 4. calculate a value for a day's news stories. This section will describe a series of experiments which evaluated a number of variations of the full system.

The full system was compared to a baseline stock trading system inspired in some of the ideas of the system described in Torgo [2010]. Namely, the baseline system is based on predictive models trained to forecast the future daily returns at some time horizon (the experiments considered 1, 3, 5 and 10 days ahead). The predictive models considered as predictors several technical indicators that can be calculated with the past prices. Concretely, the following variables were used as predictors of the forecasting models:

- ATR - is an average true range which is a Welles Wilder' style moving average of the true range [FMLabs, 2012b].

- SMI - is a stochastic oscillator which is a momentum indicator that relates the location of each day's close relative to the high/low range over the past 'n' periods [FMLabs, 2012j].
- ADX - is a directional movement index which is a moving average of the directional movement index [FMLabs, 2012a].
- Aroon - is an indicator which attempts to identify the start of trends [FMLabs, 2012c].
- Bollinger bands - is an indicator which compares a security's volatility over time [FMLabs, 2012d].
- Chaikin volatility - is the rate of change of the trading range FMLabs [2012e].
- Close Location value - is an indicator which relates the day's close to its trading range [StockCharts.com, 2012].
- Ease of movement value - is an indicator which emphasizes the days where the security moves easily and minimizes the days where the security does not move easily [FMLabs, 2012f].
- MACD - is an indicator which is a price oscillator [FMLabs, 2012g].
- MFI - is a ratio of positive and negative money flow over time [FMLabs, 2012h].
- SAR - is a Parabolic Stop-and-Reverse which calculates a trailing stop [FMLabs, 2012i]
- Volatility - this was a variety of volatility indicators: Close-to-Close,OHLC Volatility, and High-Low Volatility. [Sitmo, 2012]

Several learning algorithms were considered to obtain forecasting models using the set up described above. Moreover, for each of these learning algorithms a large set of parameter variants was also considered in the experiments. The learning algorithms were:

- Support vector machines (SVM) as implemented in the R package **e1071** [Dimitriadou, Hornik, Leisch, Meyer, and Weingessel, 2011].
- Multivariate adaptive regression splines (MARS) as implemented in the R package **earth** [Hastie and Tibshirani., 2011].
- Artificial neural networks (NN) using the implementation available in the R package **nnet** [Venables and Ripley, 2002].
- Random forest (RFs) as implemented in the R package **randomForest** [Breiman, 2001].
- Regression trees (RTs) as implemented in the R package **rpart** [Therneau and Atkinson., 2012].

The overall goal of the these experiments was to test its initial hypothesis, which was: adding information from past news stories will improve the predictive performance of an existing stock trading system. With that purpose several experiments were designed where extra features resulting from the text mining strategies described in the previous chapters, were added to the technical features.

The experiments used 4 companies: 1. Microsoft, 2. Apple, 3. Google and 4. IBM to evaluate the system. The four companies were chosen because 1. they were relatively well represented in the news corpus and consequently it was possible to use the information retrieval techniques to select related news stories and 2. to ensure that any results were not biased to a single company. The evaluation results were an average of the combined results returned by all of the companies. The systems were compared using the following measures: 1. F-measure, 2. sell returns and 3. buy returns. These evaluation measures were chosen because company share prices are quoted in currency (pounds, dollars, etc) whereas the market indexes used in the previous experiments are measured in points.

6.2.1 Selection of strategies for full system testing

It was not possible to test every variation of each classification and information retrieval strategy on the full system evaluation. A decision was made to select the best classification strategy for each "level" (sentence, document and quote) of the news story. The best sentence level strategy was determined to be the strategy which returned the highest points difference in the previous experiments. The quote and document level strategies could be configured by classifier confidence, consequently a decision was taken to select a strategy for each level which had the highest mean points difference.

The three chosen strategies for the full system evaluation were:

- Sentence level, event and sentiment rules which classified news story text.
- Document level, self-train which used event & sentiment rules to classify news headings (headlines).
- Quote level, contextual strategy which classified direct speech from members of the unbiased group.

6.2.2 Experimental Setup

The stock trading system relies upon the construction of models from features, and therefore requires training and evaluation data. There was limited news data available because the news corpus contained stories published between October 2008 and April 2011. News stories which were published on weekends and holidays could not be used because the financial

markets were closed. The split selected was 300 days for training and 100 days for evaluation. The experiments were repeated 20 times with differing selections of dates from the training and evaluation data in a process known as Monte Carlo Simulations which is a process intended to eliminate bias from the results. The selected classification strategies had access to news data published between October 2008 and July 2010 to construct models or linguistic resources.

The existing stock system calculates separate evaluation results with a number of individual models. It was not possible to run all the experiments with all of the possible learners. A decision was made to select the three "best" learners to use in the experiments. A baseline experiment was made where technical indicators were used as the only features. The baseline experiment used data between October 2008 and April 2011 with share price information from "Microsoft", "Apple", "IBM" and "Google". The stock system ranked the learners from each experiment. A mean rank was taken for each learner, and the learners with the lowest mean rank were selected for the experiments. The selected learners were a selection of SVMs and a MARS learner.

The experiments were carried out on 4 separate trading horizons: 1,3,5 and 10 days ahead. The trading horizon indicated how many days in the future a position would be closed , for example 10 days ahead would indicate that a position would be closed 10 days after the original trade. The variation of trading horizons was to identify if news effected stock prices for more than one day.

The experiments used the models generated from the baseline technical indicators (see page 125) as well as features extracted from news information. News information is: 1. sentences from news text, 2. quotes extracted from news text and 3. complete news stories. A news information feature is a separate daily score computed for: sentences, quotes or news stories. A daily score was computed for 1. sentences by aggregating the scores of individual sentences, 2. quotes by aggregating the scores of individual quotes and 3. news stories by aggregating scores of the individual news stories published in a single day which were selected for a specific company ("Microsoft", "Apple", "IBM" or "Google"). The news features were combined with the models in the following manner: 1. a single news feature (sentence, quote or news story), 2. two news features (sentence and quote, quote and news story or news story and sentence) or 3. three news features (sentence, quote and news story).

6.2.3 Results

This subsection will present results in three parts: comparison of information retrieval strategies, comparison of combinations of news information features and comparison of information retrieval and news information features with baseline system which uses technical indicators only. The evaluation metrics used for the experiments were: 1. F-measure, 2. buy

returns and 3. sell returns.

The comparison of information retrieval system experiments were designed to identify the influence of a specific information retrieval system (company or industry ontology) on the evaluation metrics, consequently the experiments used an average of the evaluation metrics for all possible experiments grouped by information retrieval strategy. An average was chosen because it was the information retrieval strategy that was evaluated rather than individual combinations of news features and information retrieval strategy.

The combination of news features experiments were designed to identify the influence of using 1, 2 or 3 news features in combination with the technical indicator features irrespective of the information retrieval strategy. The experiments used an average of the evaluation metrics for experiments which used either 1,2 or 3 news information, consequently for each trading horizon there were three results.

The final experiments compared a baseline of technical indicators against all possible combinations of news information features and information retrieval strategies. The results were an average of the evaluation metrics for the experiments conducted for each company ("Microsoft", "Apple", "IBM" or "Google"). An average was chosen to ensure that the results were not biased towards an individual company.

Comparison of Information Retrieval Strategies

The information retrieval evaluation tested the influence of the company and industry ontologies information retrieval strategies upon the trading results. The results are in Table 6.2. The results demonstrate that the news story classification strategies which used the Industry ontology to recommend news stories had superior results at trading horizons 3 and 10 days for buy returns, and trading horizons 3,5 and 10 days for F-measure. The superiority, however, was within the standard deviation. In summary, the influence of the information retrieval strategies on the evaluation measures was negligible.

News Feature Combination

The news feature combination evaluation tested the effect of having 1, 2 or 3 news features created with separate news classification strategies in combination with technical indicator features. The results are displayed in Table 6.3. There are negligible differences between the number of news features and the evaluation measures.

IRS	Trading Horizon	F-measure	Buy Return	Sell Return
Company	1	0.02±0.03	0.00±0.00	0.00±0.00
Industry	1	0.02±0.03	0.00±0.00	0.00±0.00
Company	3	0.20±0.09	0.06±0.05	0.00±0.01
Industry	3	0.25±0.09	0.09±0.07	0.00±0.01
Company	5	0.31±0.09	0.07±0.06	-0.01±0.02
Industry	5	0.33±0.08	0.07±0.06	-0.01±0.02
Company	10	0.30±0.07	0.07±0.06	-0.01±0.02
Industry	10	0.32±0.10	0.08±0.09	0.00±0.03

Table 6.2: Comparison of Information Retrieval Strategies. IRS = Information Retrieval Strategies.

Evaluation against baseline

This evaluation compares the results of adding news feature and combination of news features to the baseline of using technical indicator features. The evaluation separated the news features by information retrieval strategy and trading horizon (1,3,5 and 10 days).

The results for trading horizon 1 day ahead are displayed in Table 6.4.There were a number of experiments which used news features which returned a F-measure score, but the score was marginally above 0. There was no experiment which used news features which scored significantly higher than the baseline.

NNS	Trading Horizon	F-measure	Buy Return	Sell Return
1	1	0.02 ±0.03	0.00 ±0.00	0.00 ±0.00
2	1	0.02 ±0.03	0.00 ±0.01	0.00 ±0.00
3	1	0.02 ±0.04	0.00 ±0.01	0.00 ±0.01
1	3	0.23 ±0.09	0.08 ±0.06	0.00 ±0.01
2	3	0.22 ±0.10	0.07 ±0.06	0.00 ±0.01
3	3	0.22 ±0.08	0.07 ±0.05	0.00 ±0.01
1	5	0.32 ±0.09	0.07 ±0.06	0.00 ±0.01
2	5	0.33 ±0.09	0.08 ±0.07	0.00 ±0.01
3	5	0.32 ±0.09	0.07 ±0.07	-0.01 ±0.02
1	10	0.30 ±0.08	0.08 ±0.09	0.00 ±0.03
2	10	0.31 ±0.09	0.08 ±0.09	0.00 ±0.03
3	10	0.31 ±0.09	0.08 ±0.10	0.00 ±0.04

Table 6.3: Comparison of Combinations of news features. NNS = number of news features.

Features	I.R.	F-measure	Buy Return	Sell Return
TI	NA	0.00±0.00	0.00±0.00	0.00±0.00
TI + Qu.	Comp.	0.00±0.00	0.00±0.00	0.00±0.00
TI + Qu.	Ind.	0.02±0.04	0.00±0.01	0.00±0.00
TI + Ru.	Comp.	0.02±0.04	0.00±0.01	0.00±0.00
TI + Ru.	Ind.	0.02±0.03	0.00±0.01	0.00±0.00
TI + St.	Comp.	0.02±0.04	0.00±0.01	0.00±0.00
TI + St.	Ind.	0.02±0.03	0.00±0.01	0.00±0.00
TI + Qu.+ Ru.	Comp.	0.02±0.04	0.00±0.01	0.00±0.01
TI + Qu.+ Ru.	Ind.	0.02±0.03	0.00±0.01	0.00±0.00
TI + Qu.+ St.	Comp.	0.02±0.04	0.00±0.01	0.00±0.01
TI + Qu.+ St.	Ind.	0.02±0.04	0.00±0.01	0.00±0.00
TI+ Ru.+ St.	Comp.	0.02 ±0.04	0.00 ±0.01	0.00 ±0.01
TI+ Ru.+ St.	Ind.	0.02 ±0.04	0.00 ±0.01	0.00 ±0.01
TI+ Ru+ St. Qu.	Comp.	0.02 ±0.04	0.00 ±0.01	0.00 ±0.01
TI+ Ru+ St. Qu.	Ind.	0.02 ±0.04	0.00 ±0.01	0.00 ±0.01

Table 6.4: Comparison of results for full system trading horizon 1 day. TI= technical indicators, Ru.=Rules, Qu. = quotes, ST. = self-training, Comp. = Company Ontology and Ind. = Industry Ontology.

The trading results for trading horizon 3 days are presented in Table 6.5. There were a number of experiments which returned higher F-measures and Buy Returns than the baseline, but the difference was within the standard deviation for the experiment. In summary, there was a negligible difference between the baseline and experiments which used news features.

The trading results for trading horizon 5 days are presented in Table 6.6. There were a number of experiments which returned higher F-measures and Buy Returns than the baseline, but the difference was within the standard deviation for the experiment. In summary, there was a negligible difference between the baseline and experiments which used news features.

The trading results for trading horizon 10 days are presented in Table 6.7. There were a number of experiments which returned higher F-measures and Buy Returns than the baseline, but the difference was within the standard deviation for the experiment. In summary, there was a negligible difference between the baseline and experiments which used news features.

In summary, there was no demonstrable advantage for the experiments which used news feature(s) and technical indicators over the baseline which used only technical indicators. There were some experiments which outperformed the baseline, but the gain were within the standard deviation of the experiment.

Features	I.R.	F-measure	Buy Return	Sell Return
TI	NA	0.23 ±0.10	0.08 ±0.07	0.00±0.01
TI+ Qu.	Comp.	0.20 ±0.05	0.05 ±0.04	0.00 ±0.01
TI+ Qu.	Ind.	0.25 ±0.09	0.08 ±0.07	0.00 ±0.02
TI+ Ru.	Comp.	0.20 ±0.13	0.07 ±0.06	0.00 ±0.01
TI+ Ru.	Ind.	0.27 ±0.12	0.10±0.11	0.00 ±0.00
TI+ St.	Comp.	0.23 ±0.08	0.07 ±0.06	0.00 ±0.01
TI+ St.	Ind.	0.25 ±0.07	0.07 ±0.05	-0.01 ±0.01
TI+ Qu.+ Ru.	Comp.	0.20 ±0.15	0.07 ±0.06	0.00 ±0.01
TI+ Qu.+ Ru.	Ind.	0.27 ±0.13	0.11 ±0.10	-0.01 ±0.01
TI+ Qu.+ St.	Comp.	0.19 ±0.07	0.05 ±0.04	0.00 ±0.01
TI+ Qu.+ St.	Ind.	0.25 ±0.09	0.08 ±0.06	-0.01 ±0.01
TI+ Ru.+ St.	Comp.	0.22 ±0.08	0.06 ±0.05	0.00 ±0.01
TI+ Ru.+ St.	Ind.	0.23 ±0.08	0.08 ±0.06	0.00 ±0.01
TI+ Ru+ St. Qu.	Comp.	0.17 ±0.11	0.05 ±0.05	0.00 ±0.01
TI+ Ru+ St. Qu.	Ind.	0.26 ±0.10	0.08 ±0.07	-0.01 ±0.01

Table 6.5: Comparison of the results for full system trading horizon 3 days. TI= technical indicators, Ru.= Rules ,Qu. = quotes, ST. = self-training, Comp. = Company Ontology and Ind. = Industry Ontology.

6.3 Discussion of results

This chapter has used trading to evaluate individual news classification strategies as well as the full system which used the news classification and the information retrieval strategies. The news classification strategies were evaluated by points difference. A sensitivity experiment was conducted to evaluate the influence of classifier confidence upon points difference. The results were inconclusive. Three news classification strategies which scored the highest points difference were selected for each level (quote, sentence and story) to use in the full system experiment.

The full system evaluation used the three selected classification measures in conjunction with the information retrieval strategies to classify news for specific companies (Apple, IBM, GOOGLE and Microsoft) and add this information as news based features. The news classification strategies output was added either as a single feature or in conjunction with the remaining classification strategies to produce two or three extra features. These features were added to an existing stock trading system which used technical indicators as features. The evaluation measures were 1. percentage sell returns, 2. percentage buy returns and 3. F-measure.

Features	I.R.	F-measure	Buy Return	Sell Return
TI	NA	0.31 ±0.10	0.06±0.06	0.00 ±0.02
TI+ Qu.	Comp.	0.32 ±0.10	0.07 ±0.07	-0.01 ±0.02
TI+ Qu.	Ind.	0.33 ±0.08	0.08 ±0.07	-0.01 ±0.01
TI+ Ru.	Comp.	0.31 ±0.09	0.06 ±0.06	0.00 ±0.02
TI+ Ru.	Ind.	0.33 ±0.09	0.07 ±0.07	-0.01 ±0.02
TI+ St.	Comp.	0.31 ±0.10	0.07 ±0.07	0.00 ±0.01
TI+ St.	Ind.	0.32 ±0.08	0.07 ±0.07	-0.02 ±0.02
TI+ Qu.+ Ru.	Comp.	0.28 ±0.10	0.07 ±0.07	0.00 ±0.01
TI+ Qu.+ Ru.	Ind.	0.35 ±0.09	0.09±0.08	-0.01 ±0.02
TI+ Qu.+ St.	Comp.	0.32 ±0.09	0.07 ±0.07	-0.01 ±0.02
TI+ Qu.+ St.	Ind.	0.34 ±0.08	0.08 ±0.08	-0.01 ±0.02
TI+ Ru.+ St.	Comp.	0.32 ±0.09	0.07 ±0.07	-0.01 ±0.02
TI+ Ru.+ St.	Ind.	0.32 ±0.09	0.07 ±0.07	-0.01 ±0.02
TI+ Ru+ St. Qu.	Comp.	0.34 ±0.11	0.09 ±0.07	-0.01 ±0.02
TI+ Ru+ St. Qu.	Ind.	0.34 ±0.08	0.09 ±0.08	-0.01 ±0.01

Table 6.6: Comparison of results for full system trading horizon 5 days. TI= technical indicators, Ru.=Rules, Qu. = quotes, ST. = self-training, Comp. = Company Ontology and Ind. = Industry Ontology.

The full system evaluation tested the influence of: 1. information retrieval strategies and 2. addition of news features (1,2 or 3). The experiments revealed a small difference between the information retrieval strategies, but this difference was less than the standard deviation for the experiment. There was a small difference between experiments which used 1,2 or 3 additional news features, but the difference was within the standard deviation for the experiment.

The final experiment evaluated each possible combination of technical indicator, news features and information retrieval strategy against a baseline which used only technical indicators. There were some combinations which outperformed the baseline, but the difference was within the standard deviation for the experiment.

In summary, there was some weak evidence for features derived from news improving a stock prediction system which only used technical indicators. The evidence was inconsistent and demonstrated gains which were within the standard deviation.

Features	Model	F-measure	Buy Return	Sell Return
TI	NA	0.30 ±0.07	0.08 ±0.09	0.00 ±0.03
TI+ Qu.	Comp.	0.29 ±0.07	0.08 ±0.09	0.00 ±0.03
TI+ Qu.	Ind.	0.31 ±0.13	0.08 ±0.11	-0.01 ±0.03
TI+ Ru.	Comp.	0.29 ±0.07	0.08 ±0.09	0.00 ±0.03
TI+ Ru.	Ind.	0.30 ±0.07	0.08 ±0.09	0.00 ±0.03
TI+ St.	Comp.	0.29 ±0.08	0.08 ±0.09	0.00 ±0.03
TI+ St.	Ind.	0.30 ±0.08	0.08 ±0.09	0.00 ±0.03
TI+ Qu.+ Ru.	Comp.	0.30 ±0.08	0.08 ±0.09	0.00 ±0.03
TI+ Qu.+ Ru.	Ind.	0.34 ±0.12	0.09 ±0.10	-0.01 ±0.03
TI+ Qu.+ St.	Comp.	0.29 ±0.08	0.08 ±0.09	0.00 ±0.04
TI+ Qu.+ St.	Ind.	0.33 ±0.13	0.09 ±0.10	0.00 ±0.04
TI+ Ru.+ St.	Comp.	0.30 ±0.08	0.08 ±0.09	0.00 ±0.04
TI+ Ru.+ St.	Ind.	0.32 ±0.10	0.09 ±0.10	-0.01 ±0.03
TI+ Ru+ St. Qu.	Comp.	0.29 ±0.08	0.08 ±0.09	0.00 ±0.04
TI+ Ru+ St. Qu.	Ind.	0.32 ±0.11	0.08 ±0.10	-0.01 ±0.02

Table 6.7: Comparison of results for full system trading horizon 10 days. TI= technical indicators, Ru.= Rules, Qu. = quotes, ST. = self-training, Comp. = Company Ontology and Ind. = Industry Ontology.

Chapter 7

Conclusion

This chapter will discuss the contributions presented in this thesis and the direction of future work as well as a conclusion for the thesis. The contributions will be discussed in general terms rather than the specific claims presented in Chapter 1.

7.1 Discussion of Thesis

This thesis has documented a project which attempted to use news features to improve an existing stock trading system. The overall aim of this project was not realized because the news features did not aid the existing stock trading system. This aim was not the only motive of the project. There were other motives which sought to make advances in the constituent parts of the complete system. The thesis when compared to the secondary motivations was successful as there were demonstrable advances in: 1. ontology engineering, 2. semi-supervised learning, 3. news classification and 4. information retrieval.

The advances in ontology engineering represent a significant advance in modelling a volatile data source. The related work demonstrated that there were no significant works in this area. Furthermore, a number of works suggested that a volatile ontology was an indication of a poor construction or maintenance strategy. This assumption in a volatile domain is false. A "perfect" ontology which is derived from a domain which contains facts which are transitory will be itself volatile. The advance presented in this thesis allows the automatic construction and maintenance of an ontology generated from news stories.

The ontology generated from news was used in the information retrieval variation. The uniqueness of this variation was that it used the structure of the ontology to rank and score news stories. Ontologies in information retrieval are typically used as a form of query expansion. The indexing of a news story is typically undertaken by a separate process which uses statistical measures. Statistical measures such as term frequency inverse document

frequency (TF-IDF) ignore the semantic relationship between terms, therefore statistical relationships between terms may occur by chance. An information retrieval system which uses an indexing strategy which discovers semantic relationships between terms may be more accurate than a statistical system because a semantic relationship between terms has an underlying cause. The advance in information retrieval also used the concept of distance between terms, i.e. the greater the distance between two terms the weaker the relationship. Distance allows the discovery of related terms which never co-occur. The relationship is inferred through the interconnection in the ontology.

The news classification strategies provided two general advances: 1. generation of soft constraints from labelled data and 2. identification of speaker groups to assist sentiment classification. These advances were discovered for the classification of direct speech. Direct speech has some specific problems for sentiment classification which include: non-standard grammar, invented words and differing motivations for speakers. Classical sentiment classification techniques were not suitable. The use of soft constraints from labelled data and their expansion with lexical resources allows a person to generate soft constraints in a domain of which he is ignorant. This differs from traditional strategies which typically generate soft constraints from common knowledge or from resources outside of the domain.

The second advance used the motivation of speakers to assist a classification strategy. The assumption was that differing groups of speakers had differing motivations for speaking, which maybe susceptible to different sentiment classification techniques. The classification strategy used a combination of semantic and sentiment techniques to classify direct speech.

The work conducted for this thesis revealed problems for evaluating classification techniques. The approach chosen for this thesis was to hand label small gold standards against which the classification techniques were tested by an estimated F-measure. The weakness of small labelled sets is that they do not represent the whole collection of texts, consequently a classification strategy which demonstrates an advantage in one set of labelled texts may not show the same advantage in a different set of labelled texts. An alternate method of trading evaluation was used to assist the gold standard evaluation. There were differences between the evaluations, for example, Guided Self Training (GST) gained a similar F-measure score to the contextual strategy when evaluated with a gold standard, however GST performed poorly compared to the contextual strategy in the trading evaluation. This difference may have been due to: 1. the relatively small gold standard and 2. the evaluations measured different qualities of the classified documents, sentences or quotes. The gold standard rewarded a strategy if it classified a document, sentence or quote correctly, whereas the trading evaluation rewarded a strategy if it identified correctly classified documents, sentences or quotes which induced a movement in a market.

This thesis is a significant contribution to the fields of information retrieval, ontology engineering, semi-supervised learning and news classification. It provides the basis for further

investigation in the aforementioned fields.

7.1.1 Future Work

The future work will be centred around two themes: evaluation of classification techniques in domains with limited labelled data and sentiment analysis with background information. The future work with evaluation seeks to compare the relative performance a candidate classifier technique against a random classifier. A random classifier is hypothetically the worst type of classifier because it assigns a text to a category based upon a random probability distribution. It is likely that the features in the text will be equally distributed amongst the categories. A candidate classification strategy may demonstrate its strength or weakness by its relative distribution of features. The nearer a candidate classification strategy distribution of features is to a random distribution, the weaker the aforementioned strategy. At the present time it is not clear how a score can be calculated by comparing a candidate classification distribution of terms against a random distribution, but will be subject of future investigation.

The second direction of the future work will concentrate upon using background information to assist sentiment classification. The work conducted for this thesis produced a domain ontology which represents the interconnection of entities within a given area. This ontology can be used for two processes: reasoning with direct speech and ranking opinion makers for their influence in a given domain.

We conducted some basic experiments in reasoning with a domain ontology and direct speech which used two simple rules to select quotes: 1. a positive quote made by a leader of a company against a competitor and 2. a negative quote made by a leader of a company against his own company. The results are in Table 7.1. The rules were simple, but revealed some "interesting" quotations, for example the quote made by the CEO of Sony, which was sarcastic. It may be possible that with more sophisticated techniques direct speech which has more influence on a financial market could be identified.

Speakers have differing influences upon the financial market. An example of this was presented in Chapter 1 where Mervyn King stated the UK was in recession and sterling dropped in value. This sentiment of this statement had been made earlier by less influential speakers without any noticeable effect on the markets. A possible method is to use social network measures in a similar manner to the one used in the information retrieval section to rank speakers. The advantage of this method is that a speaker's hypothetical rank or influence score will change depending upon the domain. This scenario is an attempt to reflect a real world situation where a person is more influential in one domain than another. For example, Bill Gates works in the education domain through the Bill and Melinda Gates foundation and the technology sector through Microsoft. It is arguable that Gates has a stronger influence in the technology domain that the education sector. The value assigned to a speaker could

Affiliation	Quotation
Sony	he was very happy that Nokia chose Microsoft rather than Android.
Intel	Intel expects the consumer PC market to continue growing in the fourth quarter, but at a slower pace than in the first half of the year.
Apple	We're thrilled to be working with Verizon Wireless to get iPad into the hands of even more customers this holiday season.
Google	Microsoft is Google's top competitor, and questioned why everybody is so obsessed with Facebook and Google.
Nvidia	Manufacturers of smartphones using Google's Android operating system plan to use the Tegra chips.
Halliburton	Halliburton will comply immediately with the judge's order.
Verizon	selling 11 million iPhone this year was very doable.

Table 7.1: "Interesting" quotations derived with simple rules.

be used to increase a value assigned to a quote and therefore increasing the influence of the quote on the day's trading decision.

The work on direct speech classification presented in this thesis did not attempt to locate the subject of a quote. The classification approach assumed that if the quote was selected then it had a connection to a monitored domain. It may be possible to identify the subject of a quote using "anaphora" analysis. The examples in Table 7.1 used simple "anaphora" analysis by assuming that the subject of was the first named entity in a sentence when read left to right. For example, in the sentence "Halliburton will comply immediately with the judge's order" the subject of the sentence was assumed to be Halliburton. This simple approach may not be sufficiently robust for analysing accurately a large number of sentences, and consequently a more sophisticated approach maybe required. A possible alternative to this simple "anaphora" analysis is to use a dependency tree analysis to extract a subject for a sentence. For example, the phrase "Halliburton will comply immediately with the judge's order" dependency tree are the following:

```
(ROOT
  (S
    (NP (NNP Halliburton))
    (VP (MD will)
      (VP (VB comply)
        (ADVP (RB immediately))
        (PP (IN with)
          (NP
            (NP (DT the) (NN judge) (POS 's))
            (NN order)))))))
```

The dependencies from this tree is:

```
nsubj(comply-3, Halliburton-1)
aux(comply-3, will-2)
root(ROOT-0, comply-3)
advmod(comply-3, immediately-4)
prep(comply-3, with-5)
det(judge-7, the-6)
poss(order-9, judge-7)
possessive(judge-7, 's-8)
pobj(with-5, order-9)
```

The subject of this sentence is the dependency "nsubj(comply-3, Halliburton-1)" is the nominal subject of the sentence. The dependency tree is a more sophisticated approach than the rudimentary approach previously described.

To summarize, future work will based around: 1. finding an evaluation method for domains where there is no fully labelled set of data available 2. a more sophisticated version of direct speech analysis which involves accurately identifying a subject of a quote, estimating the relationship between a quote and weighting a quote based upon the influence of the speaker.

7.1.2 Conclusion

To conclude, this thesis presents a number of advances in several areas, and in addition provides a solid basis for future investigation. The overall aim was not met, however this does not invalidate the work presented in this thesis. There are a number reasons why an advantage could not be detected in the company specific trading evaluation, for example, the selected companies may not be susceptible to the proposed trading techniques.

Despite the overall system failing to produce a measurable gain over the baseline system, the component parts of the system represent an advance to their requisite fields and to computer science in general.

Bibliography

S. Abney. *Semisupervised Learning for Computational Linguistics*, chapter Self Training and Co-Training, pages 13–31. Chapman & Hall/CRC, 2007. ISBN 1584885599, 9781584885597.

airlinemeals.net. Airlinemeals, 2010. `http://www.airlinemeals.net/`, consulted in 2010.

G. Akrivas, M. Wallace, G. Andreou, G. Stamou, and S. Kollias. Context - sensitive semantic query expansion. In *Proceedings of IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002)*, pages 109–114, Divnomorskoe, Russia, September 2002.

Alias. Lingpipe 4.1.0, 2008. `http://alias-i.com/lingpipe`.

Blake Andrew. Media-generated shortcuts: Do newspaper headlines present another roadblock for low-information rationality? *The Harvard International Journal of Press/Politics*, 12(2):24–43, 2007.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data The Semantic Web. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*, chapter 52. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-76297-3.

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.

BBC. Nokia appoints microsoft executive as new head, 2010. `http://www.bbc.co.uk/news/business-11257069`, consulted 9/2010.

Allan Bell. *The Language of News Media*. Language in Society. Blackwell, 1991.

Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and V. S. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, pages 1–4, 2007.

J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing and Management*, 43(4):866–886, 2007.

A Blum and T Mitchell. Combining labeled and unlabelled data with co-training. In *Workshop on Computational Learning Theory*, pages 92–100, 1998.

Johan Bollen and Huina Mao. Twitter mood as a stock market predictor. *IEEE Computer*, 44(10):91–94, 2011.

Antulio Bomfim. Pre-announcement effects, news, and volatility: Monetary policy and the stock market. Technical report, Federal Reserve System, 2000.

Jethro Borsje, Leonard Levering, and Flavius Frasincar. Hermes: a semantic web-based news decision support system. In *The 23rd ACM Symposium on Applied Computing (SAC 2008), Special Track on Web Technologies*, 2008.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Jeffrey A. Busse and T. Clifton Green. Market efficiency in real time. *Journal of Financial Economics*, 65(3):415–437, September 2002.

Ivn Cantador and Pablo Castells. Semantic contextualisation in a news recommender system. In *Proceedings of the Workshop on Context-Aware Recommender Systems*, 2009. `http://ids.csom.umn.edu/faculty/gedas/cars2009/`.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM, 2010.

Carlos Carvalhob, Nicholas Klaggea, and Emanuel Moencha. The persistent effects of a false news shock. *Journal of Empirical Finance*, 18(4):597–615, November 2011.

Ming Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the Annual Meeting of the ACL*, pages 280–287, 2007.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Constraints as prior knowledge. In *ICML Workshop on Prior Knowledge for Text and Language Processing*, pages 32–39, 2008.

S. Chapman. Simmetrics (open source extensible library of similarity or distance metrics), 2008. `http://goo.gl/cn4cd`.

N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990.

Harris Collingwood. Do ceos matter? *The Atlantic*, 2009. `http://www.theatlantic.com/magazine/archive/2009/06/do-ceos-matter/307437/`.

David Culter, James Poterba, and Lawerence Summers. Speculative dynamics. *The Review of Economic Studies*, pages 529–546, 1991.

Bontcheva Cunningham, Maynard and Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

Dale Cyphert. The rhetorical analysis of business speech : Unresolved questions. *Journal of Business Communication*, 47(3):346–368, 2010.

Mohamed Yehia Dahab, Hesham Hassan, and Ahmed Rafea. Textontoex: Automatic ontology construction from natural english text. *Expert Systems with Applications*, 34: 1474–1480, 2008.

John A Daly. *Analyst statements, stockholder reactions, and banking relationships : do analysts' words matter?* PhD thesis, University of Texas, 2009.

Angela Davis and Jeremy Piger. Beyond the numbers: An analysis of optimistic and pessimistic language in earnings press releases. Technical report, Federal Reserve Bank, 2006.

W. F. M. De Bondt and R. H. Thaler. Does the stock-market overreact? *Journal of Finance.*, 40:793–805, 1985.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher Manning. Generating typed dependency parses from phrase structure parses. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias, editors, *Proceedings of Language Resources and Evaluation*, pages 449–454, 2006.

Thomas Dean and Keiji Kanazawa. Probabilistic temporal reasoning. In *Proceedings of AAAI*, pages 524–529, 1988.

Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer, and Andreas Weingessel. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2011. R package version 1.6.

Rim Djedidi and Marie-Aude Aufaure. Change management patterns (cmp) for ontology evolution process. In *Proceedings of International Workshop on Ontology Dynamics*, IWOD2009, 2009.

Rim Djedidi and Marie-Aude Aufaure. Onto-evoal an ontology evolution approach guided by pattern modeling and quality evaluation. In *Proceedings of the 6th international conference on Foundations of Information and Knowledge Systems*, FoIKS'10, pages 286–305. Springer-Verlag, 2010.

Gregory Druck, Gideon S. Mann, and Andrew Mccallum. Learning from labelled features using generalized expectation criteria. In *Proceedings of SIGIR*, pages 595–602, 2008.

Brett Drury, Luis Torgo, and J.J Almeida. Guided self training for sentiment classification. In *ROBUS WORKSHOP RANLP Proceedings*, pages 9–16, 2011.

A Esuli and F Sebastiani. Sentiwordnet a publicly available lexical resource for opinion mining. In Maria Teresa Lino, Maria Francisca Xavier, Rute Costa Fátima Ferreira, and Raquel Silva, editors, *Language Resources and Evaluation (LREC)*, pages 417–422, 2006.

C Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5:345, June 1962. ISSN 0001-0782.

FMLabs. Average directional movement index. Web Site - `http://www.fmlabs.com/reference/default.htm?url=ADX.htm`, 11 2012a.

FMLabs. Average true range (atr). Web Site -`http://www.fmlabs.com/reference/default.htm?url=ATR.htm`, 11 2012b.

FMLabs. Aroon. Web Site - `http://www.fmlabs.com/reference/Aroon.htm`, 11 2012c.

FMLabs. Bollinger bands. Web Site - `http://www.fmlabs.com/reference/Bollinger.htm`, 11 2012d.

FMLabs. Chaikin volatility. Web Site - `http://www.fmlabs.com/reference/ChaikinVolatility.htm`, 11 2012e.

FMLabs. Ease of movment values. Web Site - `http://www.fmlabs.com/reference/ArmsEMV.htm`, 11 2012f.

FMLabs. Macd. Web Site - `http://www.fmlabs.com/reference/MACD.htm`, 11 2012g.

FMLabs. Money flow index. Web Site - `http://www.fmlabs.com/reference/default.htm?url=MoneyFlowIndex.htm`, 11 2012h.

FMLabs. Sar. Web Site - `http://www.fmlabs.com/reference/SAR.htm`, 11 2012i.

FMLabs. Stochastic oscillator. Web Site - `http://www.fmlabs.com/reference/default.htm?url=StochasticOscillator.htm`, 11 2012j.

Gabriel Fung, Jeffrey Xu Yu, and Wai Lam. News sensitive stock trend prediction. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge*, pages 481–493, 2002.

Yassine Gargouri, Bernard Lefebvre, and Jean guy Meunier. Ontology maintenance using textual analysis. In *Proceedings of Multi-Conference on Systemics, Cybernetics and Informatics SCI*, 2003.

Győző Gidófalvi. Using news articles to predict stock price movements. Technical report, University of California, 2001.

Sam Glucksberg. Sentiment analysis: Emotion, metaphor, ontology and terminology. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of Language Resources and Evaluation (LREC)*, pages 94–101, 2008.

Frank Goossen, Wouter IJntema, Flavius Frasincar, Frederik Hogenboom, and Uzay Kaymak. News personalization using the cf-idf semantic recommender. In Rajendra Akerkar, editor, *Proceedings of the International Conference on Web Intelligence Mining and Semantics*, WIMS '11, pages 10:1–10:12, 2011.

Hatcher Gospodnetic, Otis and McCandless. *Lucene in Action*. Manning Publications, 2009. ISBN 1-933988-17-7.

Peter Ager Hafez. Construction of market sentiment indices using news sentiment. Technical report, Ravenpack, 2009.

Peter Ager Hafez. How news events impact market sentiment. Technical report, Ravenpack, 2010.

Matt Hamblen. Analysis: Free nokia gps could hurt tomtom, garmin. web site, 2010. `http://goo.gl/BOPJV`.

Trevor Hastie and Rob Tibshirani. *earth: Multivariate Adaptive Regression Spline Models*, 2011. R package version 3.2-1.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181, 1997.

Elaine Henry. Market reaction to verbal components of earnings press releases. *Journal of Emerging Technologies in Accounting*, 3:1–19, 2006.

Richard Higgins and Brendan D. Bannister. How corporate communication of strategy affects share price. *Long Range Planning*, 25(3):27–35, 1992.

Alexander Hogenboom, Paul van Iterson, Bas Heerschop, Flavius Frasincar, and Uzay Kaymak. Determining negation scope and strength in sentiment analysis. In *Proceedings IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2589–2594, 2011.

Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. Ontology-based news recommendation. In *Proceedings of the EDBT/ICDT Workshops*, EDBT '10, pages 16:1–16:6, 2010.

K. Izumi, T. Goto, and T. Matsui. Trading tests of long-term market forecast by text mining. In *Proceedings of International Conference on Data Mining Workshops*, pages 935–942, 2010.

Nathalie Japkowicz. Learning from imbalanced data sets: A comparison of various strategies. In Raghu Ramakrishnan, Salvatore J., Stolfo Roberto, J. Bayardo, and Ismail Parsa, editors, *ACM SIGKDD*, pages 10–15. AAAI Press, 2000.

Neville Ryant Karin Kipper, Anna Korhonen and Martha Palmer. A large-scale extension of verbnet with novel verb classes. In *Proceedings of Euralex International Congress*, pages 173–185, 2006.

Antonina Kloptchenko, Tomas Eklund, Barbro Back, and Jonas Karlsson. Combining data and text mining techniques for analysing financial reports. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 12(1):29–41, 2004.

M Kompman and M Bielikova. Content based news recommendation. In Giovanni Semeraro, editor, *E-Commerce and Web Technologies:*, pages 61–72. Springer, 2010.

M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In Douglas H. Fisher, editor, *Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

Jerome Kuperman, Manoj Athavale, and Alan Eisner. Financial analysts in the media: Evolving roles and recent trends. *American Business Review*, 21(2):74–80, 2003.

D.F. Larcker and A. Zakolyukina. Detecting deceptive discussions in conference calls, 2010. URL `http://ssrn.com/abstract=920470`.

Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Language models for financial news recommendation. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 389–396. ACM Press, 2000.

Ivo Lašek. Dc proposal: model for news filtering with named entities. In *Proceedings of the 10th international conference on The semantic web - Volume Part II*, ISWC'11, pages 309–316. Springer-Verlag, 2011.

Chang-Shing Lee, Yuan-Fang Kao, Yuan-Fang Kuo, and Mei-Hui Wang. Enhancement of domain ontology construction with a crystallizing approach. *Expert Systems Applications*, 38(6):7544–7557, 2011.

Douglas B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, November 1995. ISSN 0001-0782.

Leonard Levering, Flavius Frasincar, and Jethro Borsje. A semantic web-based approach for building personalized news services. *International Journal of E-Business Research*, pages 35–53, 2009.

Beth Levin. *English verb classes and alternations: a preliminary investigation" by Beth Levin.* The University of Chicago Press, 1993.

Bing Liu. *Web Data Mining*, chapter Opinion Mining, pages 411–438. Springer, 2007a.

Bing Liu. *Handbook of Natural Language Processing.* Springer, 2007b.

Bing Liu. *Web Data Mining*, chapter Information Retrieval and Web Search, pages 183–236. Springer, 2007c.

Bing Liu. *Web Data Mining*, chapter Link Analysis, pages 237–272. Springer, 2007d.

H. Liu and P. Singh. Conceptnet a practical common-sense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

L. Lloyd, D. Kechagias, and S. Skiena. News and blog analysis with lydia. *12 Internation Conference Spire*, pages 161–166, 2005.

Y.C. Lu, Y.C. Wei, and W. S. Chang. The application of text mining on financial corpus to the earning warning model for corporate financial distress. In *Proceedings of the 17th Conference on the Theories and Practices of Securities and Financial Markets*, 2009.

Ronny Luss. Predicting abnormal returns from news using text classification. In *Advances in Machine Learning for Computational Finance.* UCL, 2009.

Erik Mannens, Sam Coppens, Toon De Pessemier, Hendrik Dacquin, Davy Van Deursen, and Rik Van de Walle. Automatic news recommendations via profiling. In *Proceedings of the 3rd international workshop on Automated information extraction in media production*, AIEMPro '10, pages 45–50, 2010. ISBN 978-1-4503-0164-0.

Christopher Manning and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

John McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

John McManus. An economic theory of news selection. In *Annual Meeting for Education in Journalism and Mass Communication*, 1988.

I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 935–940, 2006.

Gautam Mitra and L Mitra, editors. *The Handbook of News Analytics in Finance*, chapter Applications of news analytics in finance, pages 1–42. Wiley Finance, 2011a.

Gautam Mitra and Leela Mitra, editors. *The Handbook of News Analytics in Finance*, chapter News Analytics: Framework, techniques and metrics, pages 43–69. Wiley Finance, 2011b.

Gautam Mitra and Leela Mitra, editors. *The Handbook of News Analytics in Finance*, chapter How news events impact market sentiment, pages 129–145. Wiley Finance, 2011c.

Marc-Andre Mittermayer and Gerhard F. Knolmayer. Newscats: A news categorization and trading system. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 1002–1007. IEEE Computer Society, 2006. ISBN 0-7695-2701-9.

Jaimie Murdock, Cameron Buckner, and Colin Allen. Two methods for evaluating dynamic ontologies. In Joaquim Filipe and Jan Dietz, editors, *Proceedings of Knowledge Engineering and Ontology Development*, pages 110–122, 2010.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Journal of Linguistic Investigations*, 30(1):1–20, 2007. `http://nlp.cs.nyu.edu/sekine/papers/li07.pdf`.

D Newman, C Chemudugunta, P Smyth, and M Steyvers. Analyzing entities and topics in news articles using statistical topic models. In Sharad Mehrotra, editor, *Proceedings of IEEE International Conference on Intelligence and Security Informatics*, pages 93–104, 2006.

Vicotor Niederhoffer. The analysis of world events and stock prices. *Journal Of Business*, 44 (2):193–219, 1971.

Natalya Noy, Abhita Chugh, William Liu, and Mark Musen. A framework for ontology evolution in collaborative environments. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 544–558. Springer Berlin / Heidelberg, 2006.

Oxford Thesaurus of English. Oxford thesaurus of english, 2010. `http://www.askoxford.com/worldofwords/thesauri/?view=uk`, consulted in 2009.

F Panetta. The stability of the relation between the stock market and macroeconomic forces. *Economic Notes*, pages 417–450, 2002.

D Peramunetilleke and RK Wong. Currency exchange rate forecasting from news headlines. In X. Zhou, editor, *Proceedings of 13th Australasian Database Conference.*, pages 131–139, 2002.

J Petofi, editor. *Text and discourse constitution: Empirical aspects, theoretical approaches*, chapter Connective relations—connective expressions—connective structures. De Gruyter, 1988.

F. Provost. Machine learning from imbalanced data sets 101. In Nathalie Japkowicz, editor, *AAAI Workshop on Imbalanced Data Sets*, pages 1–4, 2000.

ratemyprofessors.com. Ratemyprofessors, 2010. `http://www.ratemyprofessors.com/`, consulted in 2010.

Gerald Ratner. *The Rise and Fall... and Rise Again.* Wiley, J, 2007.

D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *World Wide Web Conference Series*, pages 502–511, 2004.

Reuters. Calais web service linked data, 2010a. `http://d.opencalais.com/er/company/ralg-tr1r/9e3f6c34-aa6b-3a3b-b221-a07aa7933633.html`, consulted in 2010.

Reuters. Calais web service, 2010b. `http://opencalais.com/`, consulted in 2009.

reviewcentre.com. Reviewcentre, 2010. `http://www.reviewcentre.com/`, consulted in 2010.

E. Riloff and J. Weibe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112, 2003.

Mohammad Robbani and Sekhar Anantharaman. An econometric analysis of stock market reaction to political events in emerging markets. In *Proceedings of Second Annual ABIT Conference*, 2004.

Eduardo Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Correlating financial time series with micro-blogging data. In *Proceedings of International Conference on Web Search and Data Mining*, pages 513–522. ACM, 2012.

Thomas Scharrenbach, Claudia d'Amato, Nicola Fanizzi, Rolf Groutter, Bettina Waldvogel, and Abraham Bernstein3. Unsupervised conict-free ontology evolution without removing axioms. In *Proceedings of International Workshop on Ontology Dynamics*, IWOD2010, 2010.

Erick Schonfeld. Google+ added $20 billion to google's market cap. web page (`http://techcrunch.com/2011/07/10/google-plus-20-billion-market-cap/`), July 2011. http://techcrunch.com/2011/07/10/google-plus-20-billion-market-cap/.

Robert Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfintext system. *Transactions on Information Systems*, 27 (2):1–19, 2009.

Thomas Schuster. News events and price movements. price effects of economic and non-economic publications in the news media. Technical report, EconWPA, May 2003.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *Proceedings, On the Move to Meaningful Internet Systems Conference*, pages 1223–1237. Springer-Verlag, 2002.

Sitmo. Volatility indicators. Web Site - (`http://www.sitmo.com/eq/172`), 11 2012.

James Spindler. Why shareholders want their ceos to lie more after dura pharmaceuticals. Technical Report 06, USC, July 2006. `http://ssrn.com/abstract=1572705`.

Steffen Staab and Rudi Studer, editors. *Ontology Evaluation*, pages 293–313. International Handbooks on Information Systems. Springer, 2004a. ISBN 3-540-40834-7.

Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*, chapter What is an Ontology?, pages 293–313. International Handbooks on Information Systems. Springer, 2004b. ISBN 3-540-40834-7.

StockCharts.com. Close location value. Web Site - `http://stockcharts.com/education/IndicatorAnalysis/indic_AccumDistLine.html`, 11 2012.

Carlo Strapparava and Alessandro Valitutti. Wordnet-affect: an affective extension of wordnet. In Maria Teresa Lino, Maria Francisca Xavier, Rute Costa Fátima Ferreira, and Raquel Silva, editors, *Proceedings of Language Resources and Evaluation (LREC)*, pages 1083–1086, 2004.

V. S. Subrahmanian and Diego Reforgiato. Ava: Adjective-verb-adverb combinations for sentiment analysis. *IEEE Intelligent Systems*, 23(4):43–50, 2008. ISSN 1541-1672.

Tamura and Hiromichi. Individual analyst characteristics and forecast error. *Financial Analysts Journal*, 58(4):28–35, 2002.

X. Tang, C. Yang, and J. Zhou. Stock price forecasting by combining news mining and time series analysis. In *Web Intelligence and Intelligent Agent Technologies*, pages 279–282, 2009.

Paul Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *Journal of Finance*, 63:1437–1467, 2008.

Terry M Therneau and Beth Atkinson. *rpart: Recursive Partitioning*, 2012. R package version 3.1-51.

James Thomas and Katia Sycara. Integrating genetic algorithms and text learning for financial prediction. In *Genetic and Evolutionary Computing 2000 Conference Workshop on Data Mining with Evolutionary Algorithms*, pages 72–75, 2000.

Luis Torgo. *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010. `http://www.liaad.up.pt/~ltorgo/DataMiningWithR`.

Maria Vargas-Vera and David Celjuska. Event recognition on news stories and semi-automatic population of an ontology. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 615 – 618, 2004.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

Pietro Veronesi. Stock market overreactions to bad news in good times: a rational expectations equilibrium model. *Review of Financial Studies*, 12(5):975–1007, 1999.

Ellen Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

Ellen M. Voorhees and Donna. Harman. Common evaluation measures. In *The Eleventh Text Retrieval Conference (TREC)*, pages 200–251, 2003. NIST Special Publication.

J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin. Learning subjective language. *Computational Linguistics*, 30(3):277–308, 2004.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354. Association for Computational Linguistics, 2005.

Wuthrich, Cho, and Leung. Daily prediction of major stock indices from textual www data. In *4th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining*, pages 364–368, 1998.

Yahoo. Nokia replaces ceo with microsoft exec in smart phone war, 2010. `http://finance.yahoo.com/news/Nokia-Replaces-CEO-With-ibd-4210132396.html?x=0&.v=1`, consulted 9/2010.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Dept. of Computer Sciences, University of Wisconsin-Madison, 2008.

G Zipf. *Human Behavior and the Principle of least effort: Introduction to human Ecology*. Addison Wesley, 1949.

Arzucan Özgür, Levent Özgür, and Tunga Güngör. Text categorization with class-based and corpus-based keyword selection. In Pinar Yolum, Tunga Güngör, Fikret S. Gürgen, and Can C. Özturan, editors, *ISCIS*, pages 606–615, 2005.

Appendix A

Definitions

A.1 Definitions

The thesis will make references to "software tools", linguistic and machine learning terms which the reader may be unfamiliar. This section will define each term used in the thesis so there is a single point of reference rather than distributing the definitions through out the thesis.

Gate	Gate is a collection of software libraries which is used for processing text. There are two modes of operation: interactive and API. The interactive mode uses a graphical interface where a user can produce "pipelines" where output from a library is piped into the next library. The API access allows a programmer to have access to the GATE libraries and therefore it is possible for the programmer to include "Gate functionality " into their programmes. More information can be found at the GATE website `http://gate.ac.uk/`
JAPE	JAPE is an acronym for Java Annotation Patterns Engine. The pattern engine allows the construction of expressions expressed in the JAPE syntax to capture sections of text from a "Gate Document ". The JAPE expressions are executed by a JAPE transducer. Multiple JAPE transducers can be chained together allowing the output of one transducer to be operated on by another transducer. More information can be found at the Gate website `http://gate.ac.uk/sale/tao/splitch8.html#chap:jape`

ANNIE	ANNIE is an acronym for: a Nearly-New Information Extraction System. The extraction system contains a: tokenizer, gazetteer, sentence splitter, part of speech tagger, semantic tagger and orthographic coreference. The work conducted for the thesis used the tokenizer, gazetteer, sentence splitter and part of speech tagger. These tools will be defined later on in this table. More information can be found at the Gate website`http://gate.ac.uk/sale/tao/splitch6.html#chap:annie`.
ANNIE Tokenizer	The tokenizer affixes token information to each linguistical unit in the text. The token information indicates if the linguistical unit is a: word, number or punctuation.
ANNIE Gazetter	The Gazetter is a series of lists which contain a "named entity" and its label. These lists are used to identify named entities in text and affix its label to the relevant text.
ANNIE Sentence Splitter	The sentence splitter is a series of JAPE rules which splits the text into sentences. The rules rely upon "split" punctuation tokens marked by the tokenizer.
ANNIE POS Tagger	The Part of Speech (POS) Tagger affixes part of speech information to text. Part of speech information at its most basic indicates if the word is a: Noun, Adjective, Adverb or Verb. Part of speech taggers can add other part of speech information. For example the Brown Corpus part of speech tag set can be found at: `http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html`

Lingpipe	Lingpipe is a collection of Java libraries which can be used to process text. The lingpipe library contains chunkers, taggers and classifiers. The work for the thesis used only the classifier which will be defined later in this table.
Lingpipe Classifier	Lingpipe contains several "generative classifiers": Language Models, Naive Bayes and Bernoulli. The classifiers classify text into predetermined categories by analysing "features" in text. The classifiers use a "model" which has been induced in training phase from pre classified documents to classify the unlabelled documents.
RapidMiner	RapidMiner is a GUI for machine learning. Experiments are conducted by chaining together operators (machine learning strategies) and the results are outputted to a file. More information can be found at: `http://rapid-i.com/content/view/181/190/`
Lucene	Lucene is an "inverted index" which is used for full text search. A lucene index is a collection of "documents" which store fields. The fields contain textual information. The search process is conducted at the field level where a query is used to look for "tokens" in the relevant fields. A token is a linguistical unit which may be a: unigram, bigram or a custom implementation. The selection of tokens for indexing is controlled by the person who produces the index. More information can be found at `http://lucene.apache.org/core/`.
OWLAPI	OWLAPI is a API which allows the construction, querying and management of Ontologies. More information can be found at: `http://owlapi.sourceforge.net/`

A.1.1 Specific term definition

Feature	A feature is a variable which has a name and a value. The variable describes some part of the data being presented to the learner. In this thesis a feature is always a unigram.
Generative Learner	A generative learner represents observable data (labelled data) as a joint distribution. The joint distribution is used to estimate a category or value for unknown data points (unlabelled data).
Discriminative Learner	In contrast a discriminative learner separates the space which encapsulates data points into regions. The regions are used to assign categories or values to the data points captured in the region
Linguistical Unit	A selection of natural units which text can be broken down into for analysis, for example unigram or bigrams
n-grams	is a contentious sequence of n items in text. The sequence may be letters (letter n-grams) or words (word n-grams). In thesis when the term n-gram or any derivative is used then reader must assume that they are word n-grams.
Inverted Index	is a data structure which maps "tokens" to locations in a database file.
Ontology	is a formal representation of knowledge in a domain. The representation of knowledge consists of: concepts and relations between concepts. The concepts used in this thesis are "business / finance" concepts, for example, names of companies
OWL	Web Ontology Language (OWL) is knowledge representation language for ontologies
RDF	Resource Description Framework which is used in the modelling of information. RDF is often used to model ontologies
Regular Expressions	Regular Expressions is a pattern language which matches strings in a larger text

An example of a Jape Rule is presented below:

```
Phase:TokenUse
//Rule takes three named entities, PEA (Property of Economic Actor),
// BEV (Business Event Verb) and company. The rule works on
// sentences which are delimited by split.

Input: Split PEA BEV Company
Options: control = brill
Rule: Event

(
  ({Company.rule =="rename"}  {BEV.rule =="rename"} {PEA.rule =="rename"}) |
 ( {PEA.rule =="rename"} {Company.rule =="rename"}  {BEV.rule =="rename"} )|
 ( {Company.rule =="rename"} {PEA.rule =="rename"}  {BEV.rule =="rename"} ) |
 ({PEA.rule =="rename"}  {BEV.rule =="rename"} {Company.rule =="rename"})

)

:Event -->
:Event.Company = {rule="Event"}
```

Appendix B

Publications

B.1 Thesis Publications

- Brett Drury and J. J. Almeida. Construction of a local domain ontology from news stories. In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, pages 400–410. Springer-Verlag, 2009.

- Brett Drury, Gaël Dias, and Luís Torgo. A contextual classification strategy for polarity analysis of direct quotations from financial news. In *RANLP*, pages 434–440, 2011.

- Brett Drury and José João Almeida. Identification of fine grained feature based event and sentiment phrases from business news stories. In Rajendra Akerkar, editor, *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, 2011.

- Brett Drury, Luis Torgo, and José João Almeida. Classifying news stories to estimate the direction of a stock market index. In *6th Iberian Conference on Information Systems and Technologies*, 2011.

- Brett Drury, José João Almeida, and M.H.M Morais. Magellan: An adaptive ontology driven "breaking financial news" recommender. In *6th Iberian Conference on Information Systems and Technologies*, 2011.

- Brett Drury, Luis Torgo, and Jose Joao Almeida. Guided self training for sentiment classification. In *Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing*, 2011.

- Brett Drury, José João Almeida, and M. H. M. Morais. Construction and maintenance of a fuzzy temporal ontology from news stories. *IJMSO*, 6(3/4):219–233, 2011.

- Brett Drury, Luís Torgo, and José João Almeida. Classifying news stories with a constrained learning strategy to estimate the direction of a market index. *IJCSA*, 9(1):1–22, 2012.

- Brett Drury and José João Almeida. The minho quotation resource. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), 2012.

- Brett M. Drury and José João Almeida. Predicting Market Direction from Direct Speech by Business Leaders. In Alberto Simões, Ricardo Queirós, and Daniela da Cruz, editors, *1st Symposium on Languages, Applications and Technologies*, volume 21, pages 163–172. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

- Brett Drury, José João Almeida, and M. H. M. Morais. An error correction methodology for time dependent ontologies. In *CAiSE Workshops*, pages 501–512, 2011.

Appendix C

Supplementary Results

C.1 Comparative Information Retrieval

Date	Headline	Company Rank	Industry Rank
2010-04-12	Tencent to invest $300m in DST	24	1
2010-04-12	Telecom Italia trims growth targets	-	2
2010-04-12	IBM helps New York go after tax deadbeats	2	-

C.2 Company Specific Ontology

Concept	Relation	Concept Type	Description
Steve Ballmer Steve Ballmer	EmployedAs EmployedBy	Chief Executive Microsoft	Commonly know information. Steve Ballmer is often quoted in all news media
Steven Sinofsky Steven Sinofsky	EmployedAs EmployedBy	Head of the Windows Group Microsoft	Less well known He is quoted less frequently than Steve Ballmer
Helmut Panke Helmut Panke	EmployedAs EmployedBy	Director Microsoft	Less well known It is difficult to find where he is quoted directly
John Lilly John Lilly	EmployedAs EmployedBy	Chief Executive Mozzilla	Reasonably well known Head of a competitor company
and 55 more entities with 2 or more assertions			

Table C.1: Subset of assertions for entity type person

Concept	Relation	Concept
Microsoft	CompetitorOf	Google
Microsoft	CompetitorOf	SalesForce.Com
Microsoft	Employs	Ray Ozzie
Microsoft	Employs	Sean Poulley
Microsoft	Produces	Windows
Microsoft	Produces	Silverlight
Microsoft	PlannedPurchaseOf	Yahoo
Microsoft	LocatedIn	Seattle
Microsoft	HasRelationship	Dell
Microsoft	HasRelationship	Verizon
Microsoft	ParentOf	Aquantive
Microsoft	Produces	Works
... and more than 200 further assertions		

Table C.2: Selection of assertions which concern Microsoft

C.3 Experimental Results - Data

	% of Data for Training					
		1	2	3	4	5
Algorithm	Classifier	F-Measure	F-Measure	F-Measure	F-Measure	F-Measure
Fully Supervised	NB	0.91				
Fully Supervised	LM	0.98				
GST	NB	0.52 ±0.05	0.61 ±0.01	0.63 ±0.01	0.63 ±0.01	0.63 ±0.01
GST	LM	0.49 ±0.04	0.60 ±0.02	0.64 ±0.01	0.64 ±0.01	0.63 ±0.02
Voting	NB	0.48 ±0.00	0.49 ±0.00	0.50 ±0.01	0.51 ±0.01	0.51 ±0.01
Voting	LM	0.48 ±0.00	0.49 ±0.00	0.49 ±0.00	0.50 ±0.00	0.51 ±0.00
Inductive (LD)	NB	0.51 ±0.01	0.51 ±0.01	0.52 ±0.01	0.54 ±0.01	0.55 ±0.01
Inductive (LD)	LM	0.49 ±0.02	0.50 ±0.01	0.51 ±0.01	0.52 ±0.01	0.53 ±0.01
Inductive (LD+RC)	NB	0.54 ±0.00	0.55 ±0.00	0.56 ±0.00	0.56 ±0.00	0.57 ±0.00
Inductive (LD+RC)	LM	0.53 ±0.00	0.54 ±0.00	0.55 ±0.00	0.55 ±0.00	0.56 ±0.00
Self-Training (LD)	NB	0.50 ±0.01	0.50 ±0.01	0.51 ±0.01	0.51 ±0.01	0.52 ±0.01
Self-Training (LD)	LM	0.48 ±0.01	0.49 ±0.00	0.50 ±0.00	0.50 ±0.01	0.51 ±0.00
Veto	NB	0.54 ±0.00	0.55 ±0.00	0.56 ±0.00	0.49 ±0.00	0.49 ±0.00
Veto	LM	0.53 ±0.00	0.54 ±0.00	0.55 ±0.00	0.55 ±0.00	0.56 ±0.00

Table C.3: Airline Meals Experimental Results

	% of Data for Training					
		1	2	3	4	5
Algorithm	Classifier	F-Measure	F-Measure	F-Measure	F-Measure	F-Measure
Fully Supervised	NB	0.96				
Fully Supervised	LM	0.99				
GST	NB	0.67 ±0.04	0.71 ±0.02	0.67 ±0.03	0.66 ±0.02	0.65 ±0.02
GST	LM	0.58 ±0.01	0.75 ±0.01	0.76 ±0.01	0.74 ±0.02	0.73 ±0.02
Voting	NB	0.47 ±0.01	0.48 ±0.01	0.51 ±0.01	0.52 ±0.01	0.54 ±0.01
Voting	LM	0.45 ±0.01	0.48 ±0.01	0.49 ±0.01	0.51 ±0.01	0.53 ±0.01
Inductive (LD)	NB	0.56 ±0.03	0.60 ±0.02	0.63 ±0.03	0.65 ±0.02	0.66 ±0.02
Inductive (LD)	LM	0.52 ±0.02	0.59 ±0.03	0.61 ±0.02	0.64 ±0.02	0.66 ±0.02
Inductive (LD+RC)	NB	0.53 ±0.00	0.54 ±0.00	0.54 ±0.05	0.57 ±0.00	0.58 ±0.00
Inductive (LD+RC)	LM	0.52 ±0.00	0.53 ±0.00	0.55 ±0.00	0.56 ±0.00	0.57 ±0.00
Self-Training (LD)	NB	0.53 ±0.03	0.56 ±0.02	0.60 ±0.02	0.62 ±0.03	0.64 ±0.02
Self-Training (LD)	LM	0.49 ±0.02	0.55 ±0.03	0.57 ±0.02	0.60 ±0.02	0.62 ±0.02
Veto	NB	0.52 ±0.00	0.54 ±0.00	0.55 ±0.00	0.57 ±0.00	0.58 ±0.00
Veto	LM	0.52 ±0.00	0.53 ±0.00	0.55 ±0.00	0.56 ±0.00	0.57 ±0.00

Table C.4: Teacher Review Experimental Results

	% of Data for Training					
		1	2	3	4	5
Algorithm	Classifier	F-Measure	F-Measure	F-Measure	F-Measure	F-Measure
Fully Supervised	NB	0.96				
Fully Supervised	LM	0.99				
GST	NB	0.51 ±0.01	0.46 ±0.02	0.48 ±0.02	0.49 ±0.02	0.50 ±0.03
GST	LM	0.54 ±0.01	0.55 ±0.01	0.49 ±0.01	0.49 ±0.02	0.48 ±0.02
Voting	NB	0.43 ±0.00	0.44 ±0.00	0.45 ±0.01	0.46 ±0.01	0.47 ±0.01
Voting	LM	0.43 ±0.00	0.45 ±0.01	0.45 ±0.01	0.46 ±0.01	0.47 ±0.01
Inductive (LD)	NB	0.57 ±0.09	0.64 ±0.07	0.61 ±0.07	0.66 ±0.07	0.65 ±0.06
Inductive (LD)	LM	0.54 ±0.09	0.59 ±0.11	0.60 ±0.07	0.65 ±0.08	0.67 ±0.06
Inductive (LD+RC)	NB	0.45 ±0.00	0.45 ±0.00	0.46 ±0.01	0.47 ±0.01	0.48 ±0.01
Inductive (LD+RC)	LM	0.45 ±0.01	0.46 ±0.00	0.46 ±0.00	0.47 ±0.00	0.48 ±0.01
Self-Training (LD)	NB	0.58 ±0.01	0.64 ±0.07	0.61 ±0.07	0.66 ±0.08	0.65 ±0.06
Self-Training (LD)	LM	0.54 ±0.09	0.58 ±0.12	0.60 ±0.08	0.65 ±0.09	0.66 ±0.07
Veto	NB	0.45 ±0.00	0.45 ±0.00	0.46 ±0.01	0.47 ±0.01	0.48 ±0.01
Veto	LM	0.45 ±0.00	0.46 ±0.01	0.46 ±0.00	0.47 ±0.00	0.48 ±0.01

Table C.5: Music Reviews Experimental Results

	% of Data for Training					
		1	2	3	4	5
Algorithm	Classifier	F-Measure	F-Measure	F-Measure	F-Measure	F-Measure
GST	LM	0.13 ±0.00	0.15 ±0.00	0.17 ±0.00	0.21 ±0.00	0.25 ±0.00
Inductive (RD+LC)	LM	0.58 ±0.00	0.58 ±0.00	0.59 ±0.00	0.59 ±0.00	0.59 ±0.00

Table C.6: GST Strategy with lower precision classifier