Stock Market Sentiment Analysis

Automatic Sentiment Analysis of Postings in Social Media

Author: Magdalena Lauber

Advisor:
Christian Gütl
Graz University of Technology
Institute for Information Systems and Computer Media

Co-Advisor:
Wei Liu
University of Western Australia
School of Computer Science

September 24, 2012

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Stock Market Sentiment Analysis

Automatisierte Sentiment-Analyse von Postings in sozialen Netzwerken

Autor: Magdalena Lauber

Betreuer:
Christian Gütl
Technische Universität Graz
Institut für Informationssyteme und Neue Medien

Co-Betreuer:
Wei Liu
University of Western Australia
School of Computer Science

September 24, 2012

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Abstract

With the widespread use of the Internet, the exchange of information and opinions is no longer limited to a small circle of friends but includes a vast pool of people that are neither personal acquaintances nor well-known professionals. This has opened a new channel for market research and interaction between company and consumer. Consumers seeking opinions and commenting on experiences of other people as well as providing their own recommendations online have produced enormous amounts of data that can be a valuable asset to market research.
However, manual analysis of market sentiment alone can no longer cope with the flood of information, making it a necessity to come up with techniques for automated processing. Especially in the area of the highly volatile stock market where opinions can jump from good to bad in a matter of hours, automated sentiment analysis of news and social media can extract and clarify relevant information and events in order to support smart trading decision.

In this study, the performance of a number of single classifiers has been evaluated on a corpus of postings from the III stock market forum using three sentiment categories (buy, sell, hold). The techniques behind these classifiers range from simple knowledge-based methods to supervised techniques such as naive Bayes, language models and support vector machines. For the lexical classifiers, several different lexicons as well as a context sensitive method based on SentiWordNet have been implemented, resulting in accuracies around 40-42%. These results made it clear that modeling the syntax and semantics of a language cannot compare to the supervised models. Evaluation of the language model classifiers, one trained on tokens (words) and one trained on characters, respectively, with different n-gram settings, yielded a much better accuracy of 53-58%. Similar performance is achieved by the naive Bayes method. The best results have been reached with the support vector machine. After the initially reaching 45-50%, it was possible to boost the accuracy to 76% by using feature selection and reduction techniques such as occurrence counting and categorical proportional difference.

A hybrid system combining the best performing classifiers from each area has been used to evaluate whether the weaknesses of one classifier can be balanced by the strengths of others. Experiments with various methods for the combination of probabilistic evidence, ranging from a simple weighted average to the Dempster-Shafer theory have shown that although the theories seemed to fit the problem, small details kept the hybrid classifier from performing as well as expected. Reaching accuracies around 67%, the combined system did not manage to outperform the best single classifier.

Zusammenfassung

Durch die zunehmenden Verbreitung des Internets ist der Austausch von Information und Meinungen nicht mehr auf einen kleinen Bekanntschaftskreis eingeschränkt, sondern weitet sich auf eine große Menge Menschen aus, die weder persönliche Bekannte noch anerkannte Authoritäten auf einem speziellen Gebiet sind. Genau diese Eigenschaft ist es, die völlig neue Ebenen der Kommunikation zwischen Kunde und Anbieter entstehen lässt. Konsumenten sind bei der Meinungsfindung nicht mehr auf einige wenige Quellen beschränkt, sondern suchen aktiv nach Erfahrungsberichten und Meinungsäußerungen anderer, und bieten im Gegenzug eigene Empfehlungen an. Diese Mechanismen haben eine enorme Menge an Daten erzeugt, die eine wertvolle Quelle für die Marktforschung darstellt. Die manuelle Analyse der Informationen alleine, wie sie in der traditionellen Marktforschung angewandt wird, ist längst kein adäquates Mittel mehr, um die Datenflut zu bewältigen. Methoden zur automatischen Verarbeitung sind ein Muss, um die manuelle Analyse zu unterstützen oder fallweise sogar zu ersetzen. Speziell im Bereich des höchst unbeständigen Aktienmarktes, in dem die Meinungen binnen Stunden von "'gut"' nach "'schlecht"' umschwenken, kann eine automatisierte Stimmungsanalayse von Nachrichten und extrahierten Inhalten von sozialen Netzwerken Benutzer und Manger von Fonds beim Filtern von relevanten Informationen und Ereignissen unterstützen und damit zu sinnvollen Geschäftsentscheidungen beitragen.

Diese Arbeit beschäftigt sich damit, die Performanz einer Anzahl von einzelnen Classifiern bei der Unterscheidung von drei Kategorien kaufen, verkaufen, halten zu evaluieren. Grundlage der Auswertungen ist ein Corpus von Postings des III Aktienforums. Die Techniken, die für die Classifier verwendet wurden, reichen vom einfachen wissensbasierten Methoden bis hin zu Ansätzen des maschinellen Lernens wie Naive Bayes Netzwerke, Sprachmodelle und Supportvektor-Maschinen. Für die lexikonbasierten Modelle wurden mehrere verschiedene Lexika manuell erstellt, sowie eine kontext-sensitive Methode basierend auf SentiWordNet implementiert. Diese Classifier erreichen für die drei Kategorien durchschnittlich eine Vorhersagegenauigkeit von 40-42%, was eindeutig darauf hinweist, dass Syntax- und Semantik-Modelle für eine Sprache kaum mit den überwachten Lernmethoden konkurrieren können. Eine Evaluierung der Sprachmodelle, von denen eines auf Wort-, das andere auf Buchstaben-Basis und jeweils mit verschiedenen N-Gram Einstellungen trainiert wurde, zeigt ein sehr viel besseres Ergebnis von 53-58%. Ähnliche Genauigkeit erreichte auch die Methode, die sich auf ein Naive Bayes Netz stützt. Die weitaus besten Ergebnisse wurden mit einer Supportvektor-Maschine erreicht. Nach anfänglichen 45-50% Genauigkeit konnte durch Featureselektions- bzw. Reduktions-Techniken wie etwa Occurrence Counting oder Categorical Proportional Difference die Performanz auf 76% gesteigert werden.

Ein hybrides System, das die Classifier mit den besten Ergebnissen von jedem Bereich kombiniert, wurde verwendet um herauszufinden, ob die Schwächen der einen Methode durch die Stärken

einer anderen ausgeglichen werden kann. Experimente mit verschiedensten Methoden zur Kombination von Wahrscheinlichkeiten wie etwa ein einfaches gewichtetes Mittel bis hin zur komplexen Dempster-Shafer Theorie wurden durchgeführt. Obwohl die Theorie auf das Kombinationsproblem perfekt zugeschnitten zu sein scheint, erreichte der hybride Classifier nur Genauigkeiten um 67% und bleibt damit hinter dem besten Einzel-Classifier zurück.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

……. ……..
date (signature)

Acknowledgments

Several people have helped me with this thesis in various ways. I would like to thank them all for the time they took to make this project happen and support my efforts:

- My supervisor Christian Gütl, for patiently waiting for progress and understanding the delay that my full-time job caused as well as for his constructive comments to improve the thesis.
- My supervisor Wei Liu, for taking the time to discuss my ideas and supporting me not only during my stay in Perth but also after leaving for Austria.
- Henry Leung, for providing me with a data corpus and thus sparing me the tedious work of collecting the data myself.
- Wilson Wong, for making his tool for word sense disambiguation available for me and explaining the algorithms behind it.
- George Tsatsaronis, for providing me with his tool for word sense disambiguation and helping me to integrate it into my implementation as well as answering a vast quantity of questions.
- Cynthia Whissell, Sanjiv Das, Mike Chen, Margaret Bradley and Peter Lang for permitting the use of their manually compiled sentiment lexicons in my experiments.
- Jana Sperschneider, for helping me during various Latex formatting crisis in the beginning and for the great company in the office and tea breaks which made the project even more fun.
- Friends and family, for stubbornly reminding me every month or so that I still have some work to do finish.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation and Background

"What do you think?"

This question invariably comes up when one is confronted with multiple choices. An important part of the decision-making process is and always has been to find out what other people, friends and trusted authorities think about the topic or product in question. This basic principle of reliance on word of mouth existed long before the World Wide Web became widely used. However, with the widespread use of the Internet, the information exchange is no longer limited to ones circle of friends but includes a vast pool of people that are neither personal acquaintances nor well-known professionals. Consumers seeking opinions and commenting on experiences of other people as well as providing their own recommendations online have produced enormous amounts of data.

This change of mechanisms is not only experienced by private persons, but also by companies who are witnessing a profound transformation in consumer interaction. A number of concurrent technology and social trends are changing the way companies connect with consumers, and more importantly how consumers interact with each other and influence buying decisions. Consumers are becoming more knowledgeable about product functionality through online reviews and pricing comparisons. The growth of online social networking represents an increasingly popular channel of recommending products and services not only to immediate friends, but to a much broader audience. The number of visits to social networking tools like Facebook and Twitter has grown rapidly in the past years, and social network users are sharing personal recommendations more frequently (see Figure 1.1).

The trend of consumers adopting advanced mobile devices is accelerating the use of social networking and makes sharing opinions on products even more easy. According to a survey of Forrester Research (2009), the use of mobile phones to access social networks doubled from five percent in the first quarter of 2009, to ten percent in the third quarter of 2009. As a reaction, retailers and companies are more actively trying to engage with consumers through these channels as well as to monitor and analyze the opinions they express.

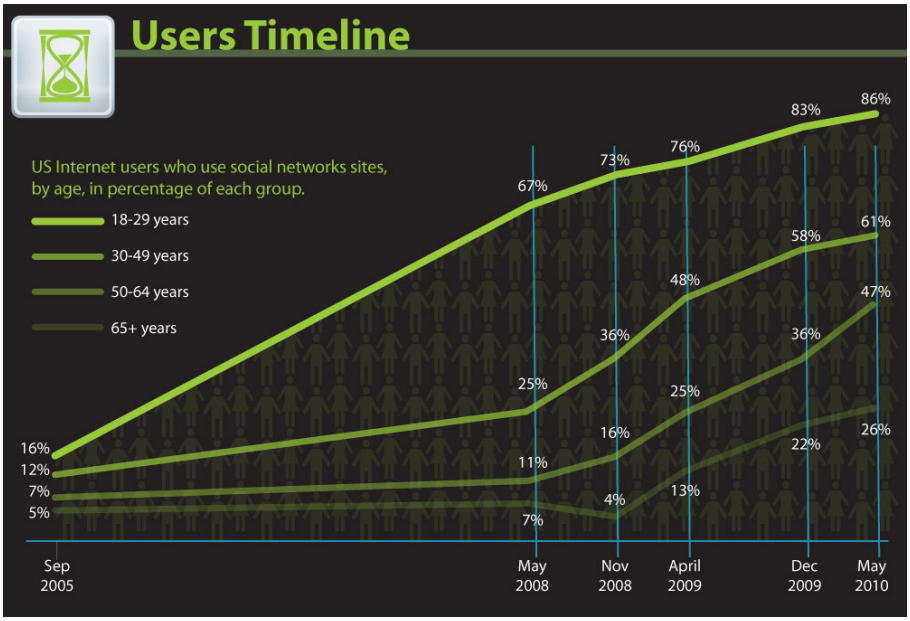

Figure 1.1: Growth of Social Media (The Search Engine Journal, 2011)

Service	Percent of Review Users Identifying Review as Having a Significant Influence on their Purchase*
Restaurant	79%
Hotels	87%
Travel	84%
Automotive	78%
Home	73%
Medical	76%
Legal	79%

** Based on responses indicating at least 4 on a 5 point scale*

Figure 1.2: Influence of online reviews on purchase decision (comScore & The Kelsey Group, 2007)

Service (Suggested Average Price)	Excellent (5 Stars)	Good (4 Stars)	Lift
Restaurant Meal ($20)	$37.95	$25.44	49%
Restaurant Meal ($50)	$59.93	$41.40	45%
Hotel ($100)	$137.36	$99.73	38%
Home ($250)	$252.15	$209.50	20%
Travel ($350)	$366.72	$299.81	22%
Legal ($60)	$104.36	$52.51	99%
Medical ($15)	$29.67	$23.54	26%

Figure 1.3: Amount consumers are willing to spend for a 5-Star rated service (comScore & The Kelsey Group, 2007)

Surveys like comScore & Kelsey (2007) clearly show the influence of the available information:

- 81% of US American Internet users have done online research on a product at least once;
- 20% do so on a typical day;
- Among readers of online reviews of restaurants, hotels, and various services (e.g., travel agencies or doctors), between 73% and 87% report that reviews had a significant influence on their purchase (see Figure 1.2);
- Consumers report being willing to pay from 20% to 99% more for a 5-star-rated item than a 4-star-rated item (see Figure 1.3);
- 32% have provided a rating on a product, service, or person via an online ratings system, and 30% have themselves posted an online comment or review regarding a product or service.

A more recent survey of The Nielsen Company (a) - with 25,000 participants much larger than the comScore & The Kelsey Group (2007) survey - shows that nine in every ten Internet consumers worldwide (90 percent) trust recommendations from people they know, while seven in every ten (70 percent) trust consumer opinions posted online. Figure 1.4 shows how the level of trust in various forms of advertising rank has changed between 2007 and 2009. An important point to notice is that in the online sector, opinions posted by other consumers outranks more traditional advertisement such as banner ads, search engine ads or emails distributed by companies by far.

Throughout 2011, Nielsen repeated the same survey, this time with 28,000 respondents in 56 countries even larger than the previous ones. The results imply largely the same trends as the changes between 2007 and 2009. While recommendations known people remain the number one source for information, online consumer reviews are still the second most trusted form of advertising with 70 percent of global consumers surveyed online indicating they trust this platform, an increase of 15 percent in four years. The 2011 survey results are shown in Figure 1.5 and Figure 1.6. For a more detailed analysis of the survey such as differences between continents or countries refer to The Nielsen Company (b).

In order to adapt the strategies of traditional opinion and market research to the new developments these surveys highlight, it is necessary to have a look at the methods applied in these areas. Opinion research, a discipline of empirical social research is a process based on statistical, psychological and empirical methods for analyzing and observing social phenomenons (Hillmann, 1994). The goal of opinion research is to identify opinions, orientations, moods, expectations and needs of consumers regarding a certain topic in order to predict future actions and identify possible influencing factors.

Market research on the other hand denotes the systematic investigation and observation of states and processes on a market. The main interest is the analysis of sales markets in order to estimate customer behavior and improve the positioning of products and services whilst minimizing risks. The easier part of the analysis relies on objective facts like income, age, sex or occupation of the customers. What is much more difficult to analyze is the subjective impressions and opinions customers have, which leads back to opinion research.

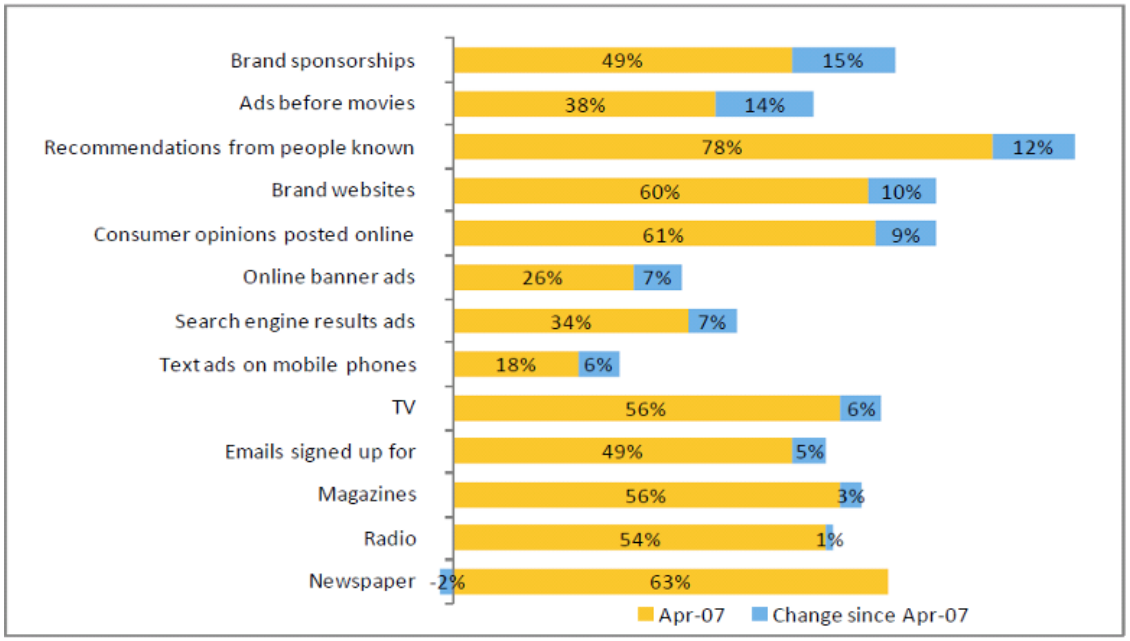

Figure 1.4: Global Online Consumer Survey: Forms of advertising ranked by changes in levels of trust from April 2007 to April 2009 (The Nielsen Company, a)

Usually, information about opinions is gathered through primary or secondary data acquisition as defined by Eckstein (2010). In the area of primary acquisition methods like oral or written interviews, group discussions, active observation or experiments aim to collect as much first hand data about subjective perceptions and behavioral patterns of consumers as possible. Secondary acquisition is concerned with the analysis of already existing data. Input for this sort of acquisition are usually previously conducted polls, demographic data, national and international surveys and so on. Up until the late nineties it was mainly manual work that was, and to a great part still is, used to compile the desired information. Opinion polls and customer satisfaction surveys are being conducted, information about the customers behavior is being collected manually in order to provide a comprehensive picture of public opinion on a certain topic or product.

With the widespread use of the Internet, a new channel for market research and interaction between company and consumer has opened. The digital age has produced a flood of freely provided information on the Internet that can be a valuable asset to market research and potentially be used as source for opinion mining. Be it posts in forums, blogs, social media networks or reviews on conventional web sites - consumers provide an endless amount of subjective ratings of statements, products and companies. But while the sources for gathering information have shifted towards the Web and social media, the process of evaluating it has not yet adapted. Manual work is still used frequently to sort through the data, although it is clear that the amount of information demands for an automated approach.

The supply of opinions, obviously, is not the problem (see also the study of Lyman and Varian (2003)). Sorting through that surplus of information, however, is. This leads to the situation that rather than gathering information, companies as well as private users need to be more concerned

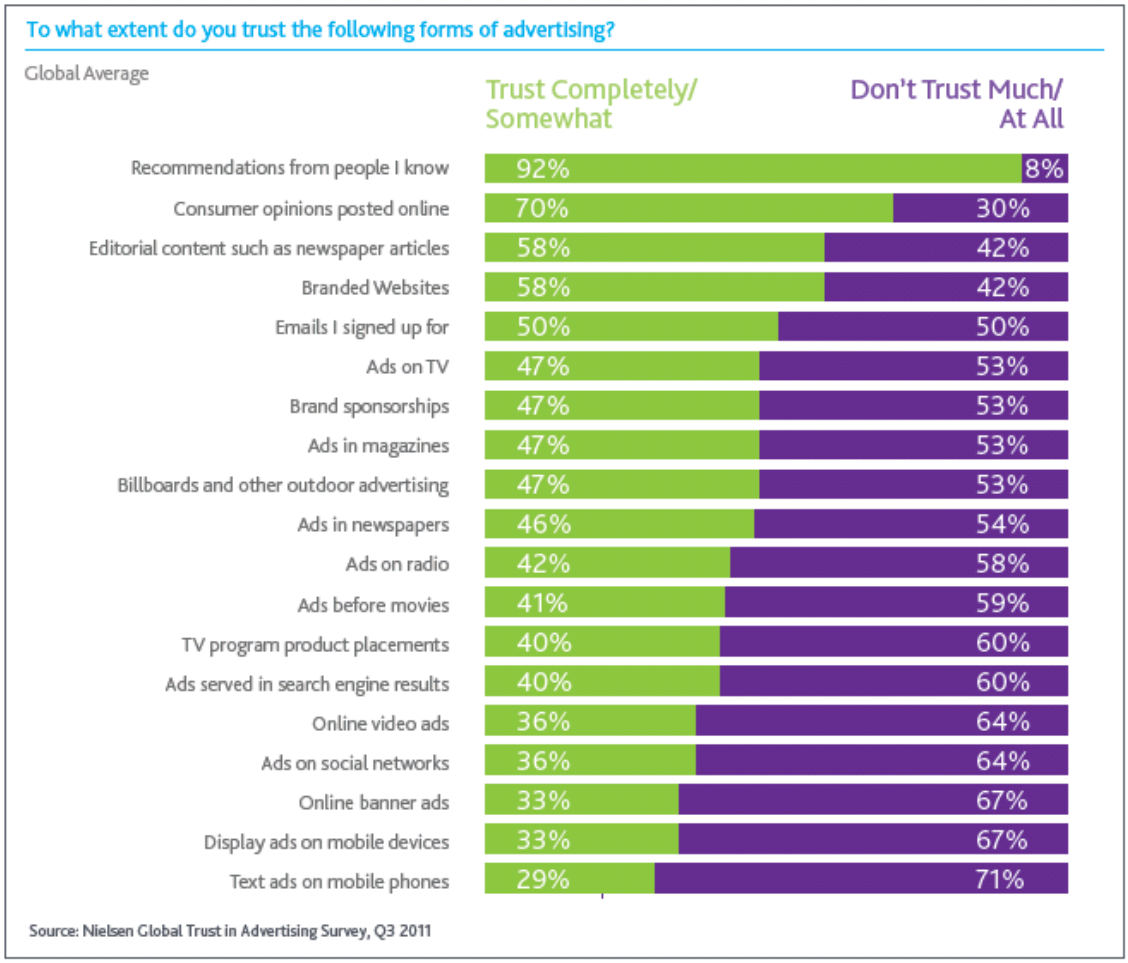

Figure 1.5: Global Online Consumer Survey: Levels of trust in various forms of advertising, Q3 2011 (The Nielsen Company, b)

with the selection, filtering and processing of the overabundance of available data. Unfortunately, the quality of the data, its relevance and correctness is often insufficient or hard to assess and demand for new methods of information retrieval and processing.

Automated *sentiment analysis* aims to provide a solution to the challenges mentioned above. It strives to offer means to monitor social media, process the information and determine the attitude of a person with respect to some product or topic. The attitude may be their judgment or evaluation, their affective state or the intended emotional communication. Generally, sentiment analysis belongs to the area of natural language processing, computational linguistics and text mining. The basic task is to categorize some input entity according to the opinion expressed in it, i.e the product is good or bad.

In general, opinion mining in the context of the stock market follows the same principle as any other product a consumer can buy. Shareholders write reviews, share recommendations and exchange views on a company's stock. An interested party can inform themselves about opinions of fellow investors, join the discussion and take advice from others. An important influence, as with all

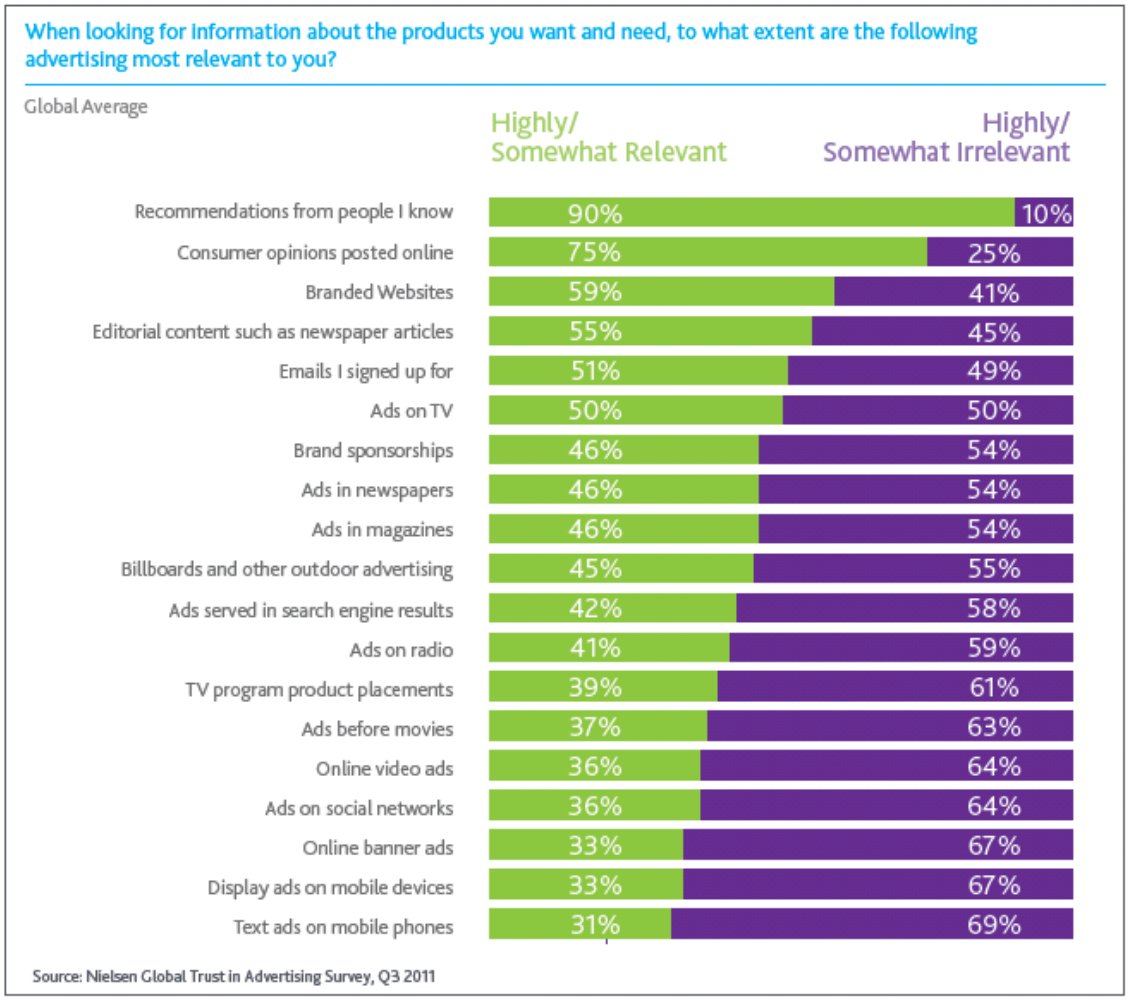

Figure 1.6: Global Online Consumer Survey: Trust in sources of information, Q3 2011 (The Nielsen Company, b)

other products, buying or selling decisions can be influenced by information collected online. Beyond that, opinions about the product *"stock"* are much more tightly tied to the company itself and the general economic situation than other products like books, movies or mp3-players. Announcements about corporate control, regulatory policy and macroeconomic conditions are just a few of the fundamental factors that share prices react to. However, as early as 1989, Cutler and Poterba (1989) shows that all the previously mentioned values relating to company news contribute only about one-third to the factors influencing stock returns. Even including world news regarding for example wars, changes in financial policies or weather conditions can still not account for even half of the stock market movement. The market situation is highly volatile and ready to change in a matter of hours. More on that topic can be found in 2.3.

On the Internet, there is an overabundance of data to be found regarding the stock market, ranging from news articles, press releases of companies to blogs and heated discussions on social media platforms such as Twitter. While news articles and such are usually of a good quality (correct use

of language, a lot of context, etc.), data extracted from social media unfortunately has none of this quality. Most research in sentiment analysis is done on data with great quality and do not take into account any social media information, thus restricting itself to one of very few domains for which a corpus is available. Additionally, many studies do not attempt to combine classifiers to a complex system but concentrate one or two standalone methods. The stock market domain has not received much attention in the beginning, but is starting to attract more and more interest (see Section 3.6). Apart from the fact that it would be most useful to be able to predict market movements, the domain poses the additional challenge of needing three categories (buy, sell, hold) as opposing to traditional sentiment analysis (positive, negative). Applying the methods of the traditional positive/negative classification to the three-way categorization of the stock domain, accuracy is bound to decrease.

Apart from this problem, both of the two main approaches as described further in Section 3.4 have their disadvantages. The language based methods concentrate on modeling the syntactical features of the language as closely as possible, thus demanding excellent linguistic knowledge and restricting the use of these classifiers to th one language they have been designed for. Further improvement of the accuracy is only possible through more detailed modeling of syntactical features and the classifiers quickly reach their limits. Additionally, bad quality data cannot be handled by such a classifier. The machine learning approach can overcome these shortcomings, however, they need an annotated data corpus for training. Again, the quality of the data plays an important role. Whether the single training samples provide enough context, are correctly labeled can make a difference in the resulting accuracies.

This thesis aims to provide a system designed to extract, analyze and classify posts from stock market forums in order to determine the affective content. The goal of the project is to meet the challenges that this special domain poses as opposed to domains typically used for experiments in sentiment analysis. Manual work alone cannot hope to cope with the flood of available information any more. More and more people use the internet to communicate their opinions and rely on the recommendations of other users. The methods proposed by researchers active in this field are to be analyzed and combined to a hybrid system in order to determine whether such a complex approach is able to outperform single classifiers. Different types of techniques, both knowledge-based as well as machine learning methods will be integrated into the hybrid to find out if it is possible to use simple classifiers with a low accuracy to improve the accuracies of well-performing classifiers. In addition, the assumption that the choice of the method used for combination of evidence can have a considerable influence on the results will be examined.

1.2 Outline of the Thesis

The remainder of this work is organized as follows: Chapter 2 gives a detailed analysis of the fundamental concepts of sentiment analysis. The general approaches and problems are discussed shortly, and the closely related field of subjectivity detection is introduced. A summary on the special area of sentiment analysis for the stock market domain concludes the chapter.

Chapter 3 gives an insight into related fields of work as well as into the current research on sentiment analysis. After the introduction to the fundamental concepts, various methods presented by the research community are analyzed and possibilities of improvement are discussed. The concept for an implementation of a hybrid approach is briefly outlined, whose individual parts are further described in the subsequent chapters.

Chapter 4 provides a detailed description of the design and architecture of the hybrid system that has been implemented for this thesis. All modules contributing to this system are presented, from the data management, preprocessing and the single classifiers to the combination of them. All libraries, tools and services used are introduced and their application in the context of the system described.

In Chapter 5 the concept of a corpus and its use in the context of this thesis is introduced. The data corpus used for training and testing in this thesis is presented and compared to other available corpora, putting a focus on how the differences might influence the result of the applied classifiers. A short discussion about the limitations of the chosen corpus and the methods applied to overcome them concludes the chapter.

Chapter 6 examines unsupervised learning in the context of sentiment analysis. A brief general introduction into the area is followed by discussion and comparison of a number of so-called sentiment lexicons. A detailed description of techniques making use of these lexicons is given, and the problems and difficulties highlighted. Ideas for possible improvements and refinements of the algorithms are pointed out. The chapter ends with a presentation and interpretation of experimental results gathered throughout a series of tests.

The next Chapter 7 delves into the area of supervised learning such as support vector machines or statistical language models and how these approaches can be applied in the context of sentiment analysis. A detailed description of techniques suitable for implementation in this project is followed by a discussion of ideas for combination with knowledge-based methods and further improvements. Finally, the evaluation results gathered throughout a number of experiments are presented and interpreted.

In Chapter 8, the previously introduced methods are combined into a hybrid system. Several different ways to combine results from the single classifiers are discussed and checked for their applicability in the special context of this project. The selected methods are then tested with the classifiers chosen in the previous chapters and the results evaluated.

The thesis concludes with Chapter 10 where problems and ideas for future work are identified. An outline of how to extend this project to include further data sources such as news reports or company statements as well as to group the sentiment information extracted from single postings into a useful tool for stock market analysis is discussed briefly.

2 The Concept of Sentiment Analysis

The basic task of sentiment analysis is to provide means to classify a given natural language text according to the opinion expressed in it. Sentiment in this context is defined as a thought, view, or attitude. The aim is to automatically identify feelings and emotions and thus offer a better understanding regarding the emotions of a large group of individual people about a given topic or product. Using various techniques, it strives to counter the vast amount of information on the Internet that has made manual review and processing infeasible. For commercial as well as political purposes, such an analysis may provide insights that are highly interesting for monetary and social reasons. Obviously, the advantage of automatic analysis is the ability to evaluate large quantities of text without or with as little as possible manual intervention.

This can be done at various levels such as document or sentence. Usually, the classification uses two to three categories: positive or negative, and frequently, neutral. Obviously, the more fine-grained the categories, the harder it gets to achieve good results. The distinction between facts and opinions might seem easy enough on the surface, but in practice separating these two parts include a number of very difficult linguistic problems.

Traditional text classification algorithms scan a piece of text in order to extract and analyze keywords, which works well for the identification of simple statements of fact. However, when it comes to opinions expressed in natural language, phrasings a lot more subtle are involved. While direct expressions of opinions like "`I hate this movie`" are fairly easy to spot, most of the time much more finesse and sophisticated language is used. The following section briefly discusses the problems and difficulties connected to Sentiment Analysis and illustrates them with examples.

2.1 Problems and Difficulties

Sentiment analysis is a complex task, even for humans. Considering the statement "`It's fifteen degrees outside.`", it is not unambiguously clear whether it is neutral, positive or negative. The answer to that question depends on the person reading it. If you are skiing in the Alps, you can probably expect melting snow and bad conditions, which makes the statement negative. If you are looking forward to a nice weekend out, it is positive, and for many people it would just be neutral information. Sentiment analysis is difficult even for human analysts in ambiguous or more complex situations. For automated methods, it is even more complicated and not always as simple or as clear-cut as expected.

Human language is far from being clear and direct. Natural language texts are often vague at best, delve deep into all sorts of subtleties or do not even conform to grammar rules. Especially when it comes to blogs and forums, the text quality often tends to be quite bad. Starting with grammatically incorrect texts and sloppy use of language to heavy usage of slang, forum posts even more than blogs are hard to interpret. With much of the available information sources tainted in these ways, it is difficult for automated mechanisms to process them.

Many of the approaches base their evaluation on the knowledge of linguistic structures. When confronted with grammatically incorrect data, the rate of success drops significantly. However, even with a correct use of language and no usage of slang or abbreviations, there are obstacles that are hard to overcome: "`The greatest thing about this movie is the plot. Even without a brain it's easy to understand.`" From a data-processing machines point of view, it identifies "`great`" and "`easy to understand`", which are potentially positive phrases. A human reader on the other hand has access to a lifetime of experience with various uses of language, can immediately see the irony in the statement and classify it correctly as negative. The use of sarcasm and irony rises the difficulty of automated analysis greatly.

Furthermore, posts in forums are often very short. The less context is available, the harder it gets to extract the information necessary for a classification. Consider a posting like this: "`Don't know how you got .095 - I tried all day and finally gave in for .10 `" (Source: HotStockMarket Forums[1]). Without any context, it is hard to decide whether this is good or bad news. In order to do a meaningful evaluation of posts in forums, it might be necessary to first filter available data for relevancy and discard the rest. Taking qualitatively bad texts into account might reduce the accuracy of classification significantly.

Another big problem are differences in language depending on the domain. The word "`complex`" can have a very positive connotation when used in the context of a movie review, whereas it tends to indicate a negative opinion when used in connection with the handling of a new mobile phone. That means that depending on the domain, the same classifier can work well in one but might not achieve a reasonable accuracy when applied to other fields. Additionally, words and phrases that are good indicators in one domain might not even occur in another. For example, a classifier based on a list of positive and negative words generated for movie reviews will not do well classifying stock market forums. While a list of words containing phrases like "`complex plot`" or "`great character`" will be applicable in the movie domain, posts in stock market forums are highly unlikely to even contain these words. Last of all, consumers tend to extensively use abbreviations as well as symbols associated with meaning, so called emoticons, to communicate sentiment towards a certain entity. These symbols are widely understood by the community but again difficult to interpret. Emoticons are mostly used in connection with some statement. Therefore, it is not enough to just interpret the symbol on its own, but it needs to be associated with the correct part of the text first. Doing this might offer a chance to enhance the intensity of an opinion detected in a piece of text.

In summary, analysis of favorable and unfavorable opinions is a task that requires high intelligence and deep understanding of the textual context, drawing on common sense and domain knowledge

[1] HotStockMarket Forums: http://www.hotstockmarket.com/forums/

as well as linguistic knowledge. The interpretation of opinions can be debatable even for humans. Even grammatically correct texts with no use irony, slang or abbreviations are hard to process. Consider this statement: "I admit it's a really awful movie ... Hell, the plot is such a mess that it's terrible, but I loved it anyway". A human would easily detect the positive sentiment. Bag-of-features classifiers as well as lexical ones would presumably find these instances difficult, since there are many words indicating a sentiment opposite to that of the entire review.

Because it does not account for the subtleties of natural language, Automated Sentiment Analysis is highly unlikely to ever be as accurate as human analysis. However, according to experiences with Amazons Mechanical Amazon Mechanical Turk, even humans only agree 79% of the time. That means although the accuracy of automated analysis is statistically well below perfect at the moment, it can at least be used to supplement manual analysis. Relying on automated analysis only, however, might cause real problems for companies, especially if they are basing any internal work flow or processes on the basis of automated social media monitoring. For example, imagine that all negative conversations are being sent to the customer care team to respond to relevant comments. If two-thirds of the "negative" conversations sent are actually positive then this process starts to break down. Perhaps more importantly, a lot of the negative conversations will never make it to the customer care team in the first place (having been incorrectly classified as positive) and unhappy customers do not get their problems dealt with.

2.2 Subjectivity Detection

Subjectivity detection or *opinion identification* is the second large research direction in and an extension to the traditional sentiment analysis. It deals with the question whether or not some input even is opinionated in the first place. Textual information can be broadly categorized into two main types: facts and opinions. Facts are objective expressions about entities, events and their properties, whereas opinions are subjective expressions that describe people's sentiments. Work in the area of sentiment analysis often assumes that the input entities represent an opinion of some kind. In an appropriate context, a sentence like "The stock price rose by 3 percent." might suggest good news, while in other contexts it does not represent an opinion of the author at all, but merely states a fact.

Taking this into account, it might be necessary for some applications to decide if a given input entity contains subjective or objective information and to clean out the objective parts. The Blog Track of the 2006 Text REtrieval Conference[2] (TREC) has the focus on exactly this topic. A number of projects, such as Esuli and Sebastiani (2006a), Pang and Lee (2004), Hatzivassiloglou and Wiebe (2000) and Wiebe and Riloff (2005) address the issue of if and to what degree objective information influences the results of Polarity Classification and discuss methods for sentence-level or sub-sentence-level Subjectivity Detection. Yu and Hatzivassiloglou (2003) provide methods for sentence-level analysis and for determining whether a document is subjective or not, but do not combine this with document polarity classification. Pang and Lee (2004) on the other hand, present a method based on minimum cuts that combines sentence-level Subjectivity Detection with

[2] 2006 Text REtrieval Conference: http://trec.nist.gov/

document-level Polarity Classification. The evaluation results suggest a significant improvement of 4% for the subsequent polarity classification by removing objective and therefore irrelevant or misleading text.

At the moment, the research community does not quite agree as to whether subjective entities bias the result of the classification, and if so, whether these entities influence the outcome in a negative manner. Mihalcea, Banea, and Wiebe (2007) summarize the outcome of several other works using subjectivity analysis as follows: *"The problem of distinguishing subjective versus objective instances has often proved to be more difficult than the subsequent polarity classification, so improvements in subjectivity classification promise to positively impact sentiment classification."* It is interesting to see that while Pang and Lee (2004) as well as others argue in their earlier work that objective information is irrelevant or even misleading to the polarity classification and can therefore be discarded, they do point out in their later work (see Pang and Lee (2008)) that objective information represents context that might be useful:

1. In determining the polarity of opinionated texts where the authors express their sentiment through statements like **"`this laptop is great`"**, (arguably) objective information such as **"`long battery life`"** is often useful to help determine the overall sentiment.

2. The task of determining whether a piece of objective information is good or bad is still not quite the same as classifying it into one of several topic-based classes, and hence inherits the challenges involved in sentiment analysis.

3. The distinction between subjective and objective information can be subtle. Is **"`long battery life`"** objective? Also consider the difference between **"`the battery lasts 2 hours`"** versus **"`the battery only lasts 2 hours`"**.

Important to notice is, however, that the classification of an entity as neutral (expressing a neutral opinion) does not automatically equal a classification of an entity as objective (lack of opinion). It is possible to have a strong opinion about something being mediocre. Due to the difficulties of this topic and the fact that integrating Subjectivity Detection does not guarantee to improve the accuracy of classification, the system designed and implemented for this thesis focuses on Sentiment Analysis only and does not include Subjectivity Detection.

2.3 Sentiment Analysis with Respect to the Stock Market

Sentiment analysis itself is not a new phenomenon as is further described in the historical overview in Section 3.1. Intense research as well as some few tools for commercial use have been around since the early 2000s. However, the main focus of both research and tools surrounded specific products on popular product review sites. Later, a more generalized approach started to include the evaluation of brand value for companies. The stock market domain has only recently been discovered as a field where the measurement of sentiment is highly valuable, and first tools have been commercially applied.

The definition of *stock market sentiment*, as such, can range from the extraction of the movement on stock exchange to the extraction of news and media information based on their polarity to analysis of the community sentiment about the market movements in social media. However, in this thesis the focus is on the latter part. The stock market is a good example of a closed system that is almost entirely sentiment driven - gossip, rumors, opinions and the often cited gut feeling are what shapes the market. Opinions in the financial and stock market sector as compared to standard product sectors are highly volatile and ready to jump from good to bad in a matter of hours. Facts like P/E, EPS or market cap are by far not the most influential sources for market movement.

A theory developed by Kahneman (1979) suggests that both investors and traders tend to behave quite irrationally based on what they hear from others about what may or may not prevail in the markets. Press releases and news have significant impact on prices, which in turn affect purchase decisions and opinions. Interestingly, the reaction of prices to news are asymmetrical: According to Engle and Victor (1993) and Soroka (2006), good news is related to large changes in prices but only for a short period of time, while bad news have a much longer lasting effect.

Knowledge and timing are crucial elements in any trading or investment decision. Taking into account the swift changes of sentiment in this area, it is even more important than in other fields for competing companies as well as private investors to keep track of these shifts of opinion. Manual extraction of the necessary information is no match for the amount of available data, and failure to react to changes and take countermeasures might result in a distortion of market trading. Devitt and Ahmad (2007) states that with the increase of computational power and lexical as well as corpus resources, the automation of detecting

Automated sentiment analysis aims to extract and clarify relevant information and events in order to support smart trading decisions. Additionally, sentiment analysis can be used to measure the impact of press articles, news releases and financial statements distributed by the company. Another possibly interesting application could be a tool for stock market advice.

Commercial tools such as for example *The Stock Sonar*[3] examine, weigh and score data from forums, blogs, Tweets as well as news media, and try to predict future stock market movement. Their goal is to offer private investors a method to get a compact overview of the great amount of opinions from analysts, media, social media and companies. *Opfine*[4] on the other hand concentrates only on the analysis of financial news in order to determine market sentiment. In this thesis, the focus is on the social media of web forums, where users interact with each other and directly influence the trading decisions of other people.

2.4 Summary

The vast amount of information makes providing a way for automated analysis not only an interesting field of research, but an absolute necessity. Especially in the stock market domain which

[3]The Stock Sonar: http://www.thestocksonar.com
[4]Opfine: http://www.opfine.com/

is mainly sentiment driven and highly volatile, the need for such tools is great. However, there are some problems that are difficult to overcome, most of which are connected to the linguistic subtleties of language.

Whereas linguistic based methods come to their limits when encountering incorrect grammar, slang or abbreviations, even machine learning techniques have problems detecting the correct sentiment when the training corpus contains data with too little context. Additionally, a correctly labeled corpus fitting the domain needs to be available. There are numerous additional and related fields like subjectivity detection which could very likely increase the accuracy of such an analysis. Nevertheless, the main focus of this thesis is on the sentiment classification itself.

In the stock market domain, little research has been published so far, even though there are already a few commercial tools on the Internet. The area is interesting insofar as the market movements are hard to predict and small investors are mostly dependent on commercial analysts for information. Instead of costly expert analysis, private investors increasingly use social media for an exchange of opinions. However, all these activities produce a flood of data which is hard to overlook if not willing to spend hours every day gathering and filtering information. The next chapters discuss the research done by others in this area, identify gaps and suggest a system to overcome some of the problems.

3 Related Work

Sentiment Analysis is a broad area of research that is strongly tied to a number of other disciplines. This chapter gives an introduction to its history and related fields of study. A short discussion on how to formalize the concept of emotions and their triggers in written text leads up to a summary of work previously conducted in this area and an overview over ideas for an implementation.

3.1 Historic overview

Automated Sentiment Analysis is a relatively new area of research. According to Pang and Lee (2008), the year 2001 marks the beginning of a rapidly increasing interest in the field of Sentiment Analysis and Opinion Mining. Although there has been a steady but low research activity in that area before, it was only recently that a great number of researchers have taken up the challenge of creating algorithms and systems for the automatic analysis of natural language texts regarding the opinions and sentiments expressed in them. Reasons for this outburst of interest include the rise of machine learning techniques in natural language processing and the availability of datasets for training, as well as the increasing awareness of commercial and intelligence applications that the area offers. Early works on beliefs as Carbonell (1979) could be called forerunners of sentiment analysis, later on the focus lay more on the interpretation of point of view, metaphor and affect. Examples include Hearst (1992), Wiebe and Bruce (1995) or Sack (1994).

3.2 Related fields of work

It is not easy to separate Sentiment Analysis from other fields of research. A number of concepts and theories from other areas play an important role in the development of an opinion mining project. This section gives a short overview over related fields of work that are strongly tied to Sentiment Analysis.

3.2.1 Linguistics and Computational Linguistics

Linguistics is the scientific study of human language, from the sounds and gestures of speech up to the organization of words, sentences, and meaning. Linguistics is also concerned with the relationship between language and cognition, society, and history. Interesting for Sentiment Analysis is mainly the part concerned with written text, meaning and cognition, especially the

subfield of linguistics called Computational Linguistics (CL). CL is a discipline which is concerned with the computational aspects and rule-based modeling of the human language.
It belongs to the cognitive sciences and overlaps with the field of Artificial Intelligence (AI), a branch of computer science aiming at computational models of human cognition. CL deals not only with formal theories about the linguistic knowledge that a human needs for generating and understanding language, but is also concerned with the development and implementation of formal models simulating aspects of the human language, and constitutes the basis for the evaluation and further development of the theories. The main goal is to create systems capable of understanding human language in order to improve human-machine interaction.

3.2.2 Information retrieval

Information retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of such information items should provide the user with easy access to the information in which he is interested. Determining which documents of a collection contain the keywords in the user query is usually not enough to satisfy the needs. In fact, the user of an IR system is concerned more with retrieving information about a subject than with retrieving data which satisfies a given query.

Baeza-Yates and Ribeiro-Neto (1999) separate between data and information retrieval: While a pure data retrieval language aims at retrieving all objects which satisfy clearly defined conditions such as those in a regular expression, an IR system usually deals with natural language texts that are not always well structured and can be semantically ambiguous. This demands for some sort of "interpretation" of the contents of the information items (documents) and a ranking according to a degree of relevance. This involves extracting syntactic and semantic information and using this information to match the user's needs. The difficulty is not only knowing how to extract this information but also knowing how to decide its relevance. Thus, the notion of relevance is at the center of information retrieval. These mechanisms of processing and interpreting written natural language is of great interest to Sentiment Analysis. However, while in IR all aspects of a text is of interest, the focus in Sentiment Analysis lies on the emotional content.

3.2.3 Cognitive Psychology

What is the sentimental value of words? How are emotions expressed in written text? What processes are involved in the interpretation of language? These questions are hard to answer and lead to another discipline important to Sentiment Analysis, namely psychology. In particular, findings from a branch called cognitive psychology plays a major role in simulating linguistic processes in addition to linguistic theories. Cognitive psychology studies mental processes including how people think, perceive, remember and learn. The core focus is on how people acquire, process and store information. Within this discipline, it is mainly the area of psycholinguistics that examines the cognitive processes constituting human language use. The relevance of computational modeling for psycholinguistic research is reflected in the emergence of a new sub-discipline: computational psycholinguistics. For an introduction to this field refer to Crocker (1996).

3.3 Concepts of emotions in written text

An important basis for the automated analysis of opinions in written text is the understanding how a human reader interprets emotions in a text. Identifying the mechanisms and linguistic clues that are used to infer the emotions of the writer are vital to the development of an automated system. Formalizations, rules and regulations proposed by the linguistic research community can be utilized to help with the extraction. One of the basic models has been developed as early as 1957: Osgood, Suci, and Tannenbaum (1971) examine how the meaning of words can be mapped in a semantic space. In a statistical process they call *factor analysis*, they determine three major aspects a term is defined by:

1. **Overall positive/negative evaluation (E): Is it good or bad for me?** Boiy, Hens, Deschacht, and Moens (2007) state that the evaluation dimension contains all choices of words, parts of speech, word organization patterns, conversational techniques, and discourse strategies that express the orientation of the writer to the current topic. Osgood et al. estimate that about 50% of the meaning can be reduced to a simple plus or minus evaluation, where evaluation is often expressed by using adjectives, as for example in "`This movie was awesome.`".

2. **Assessment of potency (P): How strongly do I feel about that?** This dimension contains all elements that express the strength of a sentiment and the commitment of the writer to the emotions his statements convey. Osgood et al. assign another 25% of identifiable meaning of a term to the judgment of potency. Intensifiers (more/less) are usually used to strengthen or weaken both positive and negative emotions. These words by themselves do not carry sentiment, but modify the intensity of emotional words co-occuring with them. An example would be "`good`" versus "`very good`".

3. **Commentary on the degree of activity (A): Is it fast or slow, active or passive, hot or cold?** Osgood et al. often find a movement dimension in responses that people make to an object or idea, but it is not as prominent as judgments of potency. An example would be "`The computer starts up fast.`" versus "`It takes a long time to boot.`".

While these categories have initially been proposed as dimensions of a semantic space, this formalization of emotional content has been utilized for the automated identification of emotion in written texts. Figure 3.1 shows a three-dimensional model as presented by Osgood et al.

3.4 Approaches to Sentiment Analysis

Motivated by a range of different problems over a great number of corpora, researchers have introduced a wide variety of approaches over the last ten years, starting from very basic ideas like keyword counting over more complex statistical and machine learning methods to hybrid systems that combine several basic and complex methods in some manner. This section examines the key concepts involved in these methods and discusses techniques relevant to the different solutions proposed by the scientific community. Most prior work in Sentiment Analysis use knowledge based

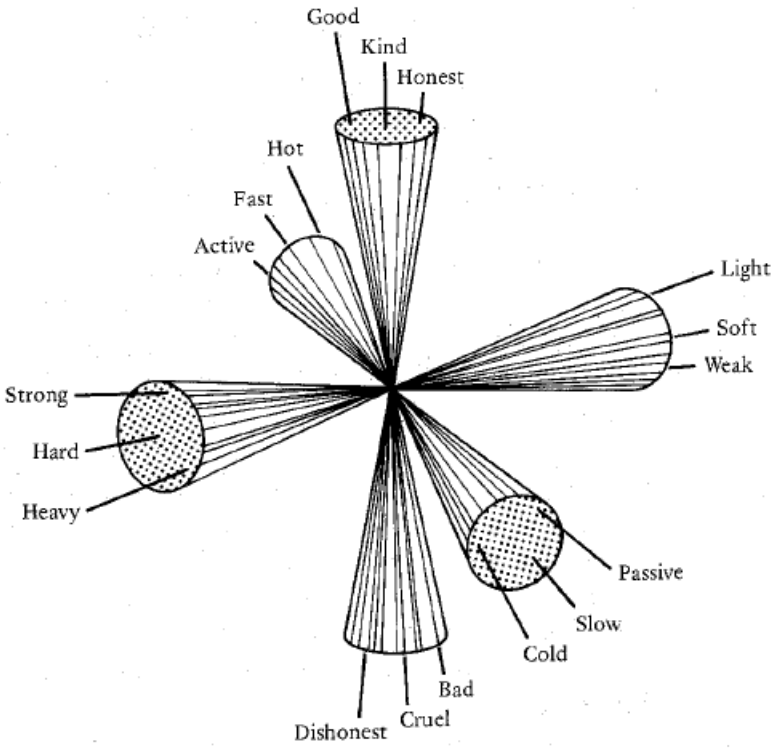

Figure 3.1: The Evaluation (E), Potency (P), and Activity (A) structure of affective meaning (Osgood, 1974)

approaches (the so-called lexical approach) that classify the sentiment of texts based on simple linguistic patterns and lexicons defining the sentiment polarity of words. These methods mostly use the scale for emotions discussed in the previous section and base classification on the general structural analysis of text. Although unsupervised learning generally includes much more than this technique, Pang and Lee (2004) state that in the context of Sentiment Analysis it can be equated with the lexical methods. The second main approach tries to utilize techniques from the disciplines of machine learning and artificial intelligence for Sentiment Analysis.

3.4.1 Knowledge-based Techniques

All the classifiers belonging to this category base their techniques on the more or less thorough analysis of the syntax and semantics of an input entity. They try to accumulate a knowledge base mirroring the knowledge of a human reader and capturing as many subtleties of natural language as possible. Using so-called sentiment lexicons, which are basically lists of terms indicating a positive or negative sentiment, thesauri like WordNet, grammatical and part-of-speech analysis to detect intensifiers, negation and other syntactical characteristics, lexical classifiers strive to integrate both structural features of language as well as the psycho-linguistic background of emotions in written text.

However, few of the sentiment lexicons are publicly available, and most of them are for the English language. This shortage has triggered a flourishing sub-discipline of Sentiment Analysis that is

concerned with the automatic or semi-automatic construction of such resources. A general change in technique led from letting people rate the sentiment of a list of words to numerous methods for automatic extraction, most of which start out with a small list of manually compiled seed words.

While Hatzivassiloglou and McKeown (1997) use a manually annotated set of words as lexicon, most other works turn to linguistic sources such as WordNet (see Section 4.3.3) as a basis for semi or fully automatic creation of sentiment lexicons. Examples are Kamps, Marx, Mokken, and De Rijke (2004) or Godbole, Srinivasaiah, and Skiena (2009) who induce graph-related distance measures and the analysis of synonym and antonym sets to expand small lists of positive and negative words into full sentiment lexicons. In order to determine the polarity strength of the candidate terms and eliminate ambiguous terms, sentiment-alternation hop counts are used. Similar ideas can be found in works of Andreevskaia and Bergler (2006) or Esuli and Sebastiani (2006a) as well as Strapparava and Valitutti (2004) who classify word-to-synset relations. Also on the basis of WordNet, Esuli and Sebastiani (2006b) propose a method for the analysis of glosses and associated synset which lead to the creation of SentiWordNet (see Section 4.3.4). Other possibilities of extending lexicons include the extraction of new words from existing corpora based on co-occurrence as presented by Takamura, Inui, and Okumura (2005).

Although the knowledge based approach is the most intuitive one and seems to be easy to implement, it has several drawbacks: Apart from the obvious fact that the development of such a classifier requires a great amount of manual work, the creation of the lexicon as well as the implementation of all rules reflecting syntax demands for an in-depth knowledge of the language it is constructed for. A good example for the linguistic approach and formalizations of semantics is the work of Moilanen and Pulman (2007). They split the classification of a complex constituent into the classification of its component constituents and operations on these that resemble the usual methods of compositional semantic analysis.

Another concern is that the dependence on the lexicon restricts the application of the classifier to the domain it has been created for (i.e. movie reviews). Switching to another domain means to redesign the lexicon in order to adapt to a different use of language. It is easy to see that lexicons are not only hard to collect, but even harder to maintain. Qiu, Liu, Bu, and Chen (2009) present an iterative approach to extract sentiment words for a new domain from an existing corpus as well as a way to assign polarities to them. The method uses a small seed lexicon to extract new sentiment words. This enhanced lexicon is the base for a new iteration. Iteration continues until no more new words can be found. The extraction rules are designed based on relations described in dependency trees. While most of the research concentrates on movie or product reviews, one of the rare works in the domain of the financial sector is Devitt and Ahmad (2007). Apart from exploring a computable metric of positive or negative polarity in financial news text which can be used in a quantitative analysis of news sentiment impact on financial markets, they also give a good overview of the aspects of news texts that affect the markets in profound ways, impacting on volumes of trades, stock prices, volatility and even future firm earnings.

Apart from problems with the generation of a sentiment lexicon, even extensive domain specific lexicons face one more problem: The emotional content a term carries is often tied to the context it appears in. This context dependency is hard to tackle, and to the authors best knowledge

there is no approach with traditional lexical classifiers that take this into account. SentiWordNet determines varying polarities for all different senses a term can have and thus provide a ground for context sensitive Sentiment Analysis. However, any lexicon-based technique making use of the context must inevitably include means to disambiguate word senses in a given context, which entails a number of other problems as described in Section 6.5.

Denecke (2008) is one of the few works making use of SentiWordNet, however, no word sense disambiguation (WSD) is done. For all sentiment words discovered in a text, simply the sum of the polarity values of all synsets belonging to it is averaged. Another approach to circumvent WSD is proposed by Verma and Bhattacharyya (2008). They base the choice of the correct word sense and corresponding synset on the guess that people express their sentiments strongly and therefore choose words with high polarity value. For their score calculation, they take into account the average maximum of all positive and negative scores of SentiWordNet, the maximum over all senses, and their weighted average.

The portability of a lexical classifiers is a big issue. Due to the linguistic nature of the construction, they are only applicable to the language they have been developed for. If classification of content written in any other language is needed, all lexical resources need to be created from the scratch again. Multilingual approaches as described in Denecke (2008). Bautin, Vijayarenu, and Skiena (2008) or Mihalcea, Banea, and Wiebe (2007) mostly concentrate on generating lexicons for the new language by leveraging on translation tools and resources available in English. As a bridge between languages, dictionaries or parallel corpora are used. Boyd-Graber and Resnik (2010) on the other hand present a procedure they call multilingual supervised latent Dirichlet allocation (MLSLDA), a probabilistic generative model that allows insights gleaned from one language's data to inform how the model captures properties of other languages.

3.4.2 Machine Learning Techniques

More recent studies move away from the rule-based approach and towards machine learning methods. Advances in computer science have lead to a development of a number of techniques for text classification. The importance of this area for sentiment analysis stems from the fact that quite a lot of problems can be formulated as applying classification, ranking or regression to given textual units. Making a decision of how positive a particular document or a body of documents is, ranking a number of documents according to their positivity, or predicting the positivity of a new text sample given a number of determined samples by utilizing the relationship between samples are just a few options of how questions of sentiment analysis can be formulated to fit into the concept of classification.

Some of the most common techniques include *decision trees* (DT), *naive Bayes networks* (NB), *support vector machines* (SVM) and *maximum entropy models* (ME). Details about the theory behind these methods can for example be found in Safavian and Landgrebe (1991), Ren, Lee, Chen, Kao, Cheng, and Cheung (2009), Hsu, Chang, and Lin (2003) or Jaynes (1957), respectively. The main aspect of all four approaches, however, is the extraction of features from a labeled corpus. These features can be anything from numerical values to boolean expressions. In the context

of sentiment analysis, a feature could for instance be how often the word "great" occurs in an input entity in relation to its occurrence in the whole corpus. The labels simply indicate to which category an input entity belongs, i.e. a posting can be labeled *positive*. If a particular feature or feature combination occurs very often, the model will learn that it is a good indicator of the category denoted by the label. Given a new input sample without a label, the algorithm is able to deduce a category by relating the features of the new entity to the ones derived from the training before. All of the four previously mentioned examples for machine learning techniques fall into the category of supervised classification. While they all have the main drawback of needing an annotated corpus for training models, they do have some important advantages compared to lexical classifiers:

1. Independence from the language. Classifiers can be used on any language by just switching the training corpus.

2. Independence from a certain domain (under the assumption that there is a training corpus available).

3. No in-depth knowledge about the syntax or semantics of the language is necessary.

Unfortunately, many of the potential applications of sentiment analysis are currently infeasible due to the huge number of features found in standard corpora. However, the selection of appropriate features is crucial to the accuracy of a classifier. To date, only a limited amount of research has been done in the area of feature selection for sentiment analysis. In their paper, O'Keefe and Koprinska (2009) evaluate a range of feature selectors and feature weights with both NB and SVM classifiers and introduce two new feature selection methods as well as three new feature weighting methods. Another interesting and very sophisticated approach by Abbasi, Chen, and Salem (2008) combines *information gain* (IG) and *genetic algorithms* (GA) in a new algorithm called *entropy weighted genetic algorithm* (EWGA) to improve feature selection.

Numerous researchers have used supervised methods of the machine learning discipline, and especially SVMs attract a lot of attention due to their good performance. One of the pioneers in using supervised learning in sentiment analysis were Pang, Lee, and Vaithyanathan (2002). In order to find out whether sentiment analysis could be treated as a special case of topic-based categorization with the two topics *positive* and *negative*, they test NB, ME and SVM classifier. Features are unigrams and bigrams, with unigrams giving the better results. Both feature frequency (how often does a feature occur in the given input entity) and feature presence (does it occur) are evaluated, with unigram feature presence leading to the best accuracy of 82.9%.

Mullen and Collier (2004) use SVMs to bring together diverse sources of potentially pertinent information, including the emotional measures of Osgood et al. (1971) for phrases and adjectives and where available, knowledge of the topic of the text. These features are further combined with unigram models of Pang et al. mentioned above as well as with lemmatized versions of them. Another work making use of SVM is by Na, Sui, Khoo, Chan, and Zhou (2004). While they use the same methodology, they are some of the few researchers that test on another domain than the movie review corpus of Pang et al.. Accuracies up to about 79% compared to around 91% for the movie domain show the impact of a data set that is a bit less clean. *Sequential*

minimal optimization (SMO) is a technique used by Whitelaw, Garg, and Argamon (2005) to reach accuracies around 90%. Sehgal and Song (2007) compare NB and ME in connection with a trust value symbolizing the relevance of an input entity to study the correlation between the sentiment, the trust value and the corresponding stock value and are able to achieve an prediction accuracy up to 81%.

3.4.3 Making Use of Unlabeled Data

The assumption of independence from domain and language as mentioned in the previous section only holds when multiple corpora are available for training. Supervised classifiers are bound to a certain domain and language by the available training data, which is one of the major disadvantages: they all need a domain specific labeled corpus. Since it is not always possible to extract user rated (stars, smilies, etc) content for a certain domain from the Web, for example Amazon reviews, it is inevitable that a lot of researchers fall back to the few domains where labeled corpora already exist. Unfortunately, only very few such corpora have been created and are freely available. Therefore, in order to explore beyond the limited domains, the need to develop methods to make use of unlabeled data is obvious. Most of the techniques in this context focus on semi-supervised learning and the automatic or semi-automatic construction of labeled information.

Liu, Li, Lee, and Yu (2004b) for example include unlabeled data by creating a representative document from the sentiment lexicon for each classification category. For each unlabeled input text, they calculate the cosine similarity to all the representative documents and assign the category with the highest similarity. Evaluative results show that training a Naive Bayes classifier on these pseudo-labeled samples achieves better results than using just the lexicon by itself.

Joachims (1999) exploits unlabeled data with a modified support vector machine (TSVM), which constructs a linear separator in a low density-area of data in such a way that the margin over both labeled and unlabeled data is maximized. Apart from having more unlabeled than labeled data available, another common case in natural language processing is that there is plenty labeled data for various domains but none or little for the domain in question. This problem is considered by Daumè and Marcu (2006), who formalize it in terms of a simple mixture model. They propose an a method using maximum entropy classifiers and their linear chain counterparts as well as inference algorithms for this special case based on the technique of conditional expectation maximization.

Hatzivassiloglou and McKeown (1997) propose a method for assigning a sentiment values to adjectives by clustering documents into same-oriented parts and then manually label the clusters positive or negative. Popescu and Etzioni (2005) develop a system called OPINE that uses term clustering for determining the semantic orientation of an opinion word in combination with other words in a sentence. All these methods provide means to include unlabeled data. There are no many studies, however, if this integration benefits the accuracy of the classification.

3.5 A Hybrid Approach

Both of the discussed approaches, knowledge based as well as machine learning, have their shortcomings. Lexicon based methods tend to be non-adaptive and restricted to language, domain and linguistic knowledge of the developer, whereas learning techniques do not effectively exploit prior knowledge and, in case of supervised learning, require an existing annotated corpus for training. While a number of researchers such as Zhou and Chaovalit (2008) or Pang et al. (2002) investigate the difference between single classifier methods and conduct comparative studies between accuracies, only few works concentrate on a hybrid approach. Especially in the first half of the last decade evidence of people combining methods is scarce. However, in the last couple of years there seems to be an increasing interest in exploiting the advantages of various techniques to overcome the disadvantages.

Kennedy and Inkpen (2005) present a hybrid where lexical knowledge, so-called valence shifters (negations, intensifiers and diminishers) contribute to the features used in an SVM and show a slight improvement in accuracy. Prabowo and Thelwall (2009) combine rule-based classification and supervised learning into a new method. The combination is basically a cascade, in which each classifier can contribute to the result of other classifiers in order to maximize the level of effectiveness. Results from tests on four different data sets show that a hybrid classification can improve the accuracy in terms of micro- and macro-averaged F1[1]. König and Brill (2006) construct a hybrid classifier that utilizes human reasoning over automatically discovered text patterns to complement machine learning. Using a standard sentiment-classification dataset and real customer feedback data, they demonstrate a significant reduction of the human effort required to obtain a given classification accuracy.

One of the few works concentrating on the financial sector also falls into the hybrid category: Das and Chen (2007) develop a system comprising of different classifiers coupled together by a voting scheme, which may be used to assess the impact on investor opinion of management announcements, press releases, third-party news, and regulatory changes. The five classifiers used range from simple lexical analysis to Vector Distance and Naive Bayes. Accuracy levels are similar to widely used Bayes classifiers, but false positives are lower and sentiment accuracy higher. Additionally to the standard test setup, they relate the detected sentiment to the Morgan Stanley High Tech Index (MSH) in order to check for a causality between message board discussion and market economics.

Table 3.1 gives an overview of the accuracies achieved by selected approaches presented in the literature. The examples roughly cover the period of the last ten years and range from the evaluation of purely knowledge-based methods to supervised techniques and hybrid systems.

[1] F1 is a measure that takes both the precision and recall of a classifier's effectiveness into account.

Author	Corpus	Number of Categories	Approach	Classifiers	Cross-validation	Domain	Samples/ Category	Accuracy (%)
Turney (2002)	Epinions[2]	2	Knowledge based	Lexical	-	Automobiles	75	84
						Banks	120	80
						Movies	120	65,83
						Travel Destinations	95	70.53
Pang et al. (2002)	Movie Reviews[3]	2	Machine Learning	NB, ME, SVM	3-fold	Movies	700	82,9
Mullen and Collier (2004)	Movie Reviews	2	Knowledge based, Machine Learning	SVM/Lexical (Turney values, Osgood values, lemma models)	10-fold	Movies	700	86
Na et al. (2004)	Product Reviews[4]	2	Machine Learning	SVM	3-fold	Mobile Phones	900	79,33
Kim and Hovy (2005)	DUC 2001 Corpus[5]	2	Knowledge based	Lexical	-	Politics & Social	50	81
Kennedy and Inkpen (2005)	Movie Reviews	3	Hybrid System	Hybrid (Lexical, SVM)	10-fold	Movies	1000	86,2
Whitelaw et al. (2005)	Movie Reviews	3	Machine Learning	Sequential Minimal Optimization (SMO)	10-fold	Movies	1000	90,2

Table 3.1: Accuracies of sentiment classification systems presented in the literature.

[2]Epinions Reviews: http://www.epinions.com
[3]Movie Review Corpus: http://www.cs.cornell.edu/people/pabo/movie-review-data/
[4]Review Centre: www.reviewcentre.com
[5]Document Understanding Conference 2001 Corpus: http://www-nlpir.nist.gov/projects/duc/

Author	Corpus	Number of Categories	Approach	Classifiers	Cross-validation	Domain	Samples/ Category	Accuracy (%)
Wilson et al. (2005)	MPQA Corpus[6]	4	Machine Learning	BoosTexter	10-fold	Politics	49	75,9
König and Brill (2006)	Movie Reviews	2	Hybrid System	Hybrid (Pattern-based, SVM)	5-fold	Movies	1000	> 91
	Microsoft Internal Customer Feedback					Customer Feedback	N/A	< 72
Das and Chen (2007)	Stock Market Forums	2	Hybrid System	Hybrid (Lexical, Vector Distance, NB)	N/A	Investor discussions	7536	47,75
						Investor discussions, reduced amount of ambiguous postings	1548	49,55
							252	52,77
McDonald et al. (2007)	Product Reviews	2	Machine Learning	Hidden Markov Model (Viterbi)	10-fold	Car seats for children	300	81,5
						Fitness equipment		81,9
						Mp3 players		87,2
Ding and Liu (2007)	Amazon Reviews[7]	2	Knowledge based	Lexical	-	Electronic products	370	79 (F-Score)

[6]Multi-Perspective Question Answering Corpus: nrrc.mitre.org/NRRC/publications.htm, see also Stoyanov, Cardie, and Wiebe (2005)
[7]Extracted from Amazon: http://www.amazon.com

Author	Corpus	Number of Categories	Approach	Classifiers	Cross-validation	Domain	Samples/ Category	Accuracy (%)
Zhou and Chaovalit (2008)	Movie Reviews	2	Knowledge based	Lexical, Rubryx	3-fold	Movies	700	66,27 to 85,54
Abbasi et al. (2008)	Movie Reviews	2	Machine Learning	SVM with Genetic Algorithms (GA), Information Gain (IG), IG + GA for feature selection	10-fold	Movies	1000	91,7
	English & Arabic Web Forums					Extremist Groups		94,72
O'Keefe and Koprinska (2009)	Movie Reviews	2	Machine Learning	SVM/NB with different feature selection techniques	10-fold	Movies	1000	87,15
Prabowo and Thelwall (2009)	Movie Reviews	2	Pseudo-hybrid System	Rule-based & SVM in a cascade	10-fold	Movies	1000	91
	Market Sentinel Corpus[8]					Product Reviews	180	83,33

[8]Proprietary Corpus from Market Sentinel: http://www.marketsentinel.com

3.6 Research in the Financial Domain

According to the early research on the movements of the stock market, the prices are mostly driven by new information rather than present and past prices. Stock prices are directly reacting to news, and keep changing without delay, thus reflecting all events and news (Fama, 1965; Fama, 1970). Following that theory of random walk and the so-called *efficient market hypothesis* (EMH), a prediction is not possible since news is unpredictable. However, research in recent years have critically examined EMH and question the validity of the hypothesis. Several studies have been conducted that show that some indicators can be extracted from social media such as blogs or forums to predict changes (Gallagher and Taylor, 2002; Qian and Rasheed, 2007). Taking these studies into account, to base trading strategies only on known information therefore present no advantage for the investors, making trend forecasts and prediction a necessity in the decision making process.

Although most of the research on sentiment analysis traditionally concentrated on specific products on popular product review sites, interest in the stock market domain has gradually increased in the recent years. As early as 2002, Fung, Yu, and Lam investigated in how news articles might influence changes of stock trend and their immediate impact on market movement. Specifically, they employ various data mining techniques such as support vector machines to cluster trends into two categories (rise/drop) and employ time series in order to predict future changes.

Antweiler and Frank (2004) use naive bayes and support vector machine classifiers to analyze the contents of 1.5 million messages posted on Yahoo! Finance and Raging Bull in the Dow Jones Industrial Average, and the Dow Jones Internet Index. They find evidence that the messages help predict market volatility both at daily frequencies and also within the trading day, but not stock returns. Thus, stock messages reflect public information extremely rapidly. According to that, they conclude that Internet data from social media may be helpful in studies of insider trading and event studies.

A more detailed overview of research in text mining surrounding stock market movements and prediction can be found in Mittermayer and Knolmayer (2006). This paper roughly covers the period from the early 2000s up until 2006 and provides a good review of the most important prototypes developed in this time as well as their performance.

Zhang and Skiena (2010) analyze blogs as well as news articles to determine how a company's reported media frequency, sentiment polarity and subjectivity anticipates or reflects its stock trading volumes and financial returns. In fact what they do is compare the results of their sentiment analysis to the daily, monthly and annual return of the stock in question. Their experiments suggest that the media correlates with the stock information. The paper focuses mainly on developing a trading strategy based on the sentiment of social media; sentiment analysis itself is done by a relatively simple lexicon based approach developed in one of their previous projects (Godbole, Srinivasaiah, and Skiena, 2009; Bautin, Vijayarenu, and Skiena, 2008).

Bollen, Mao, and Zeng (2011) investigate whether measurements of collective mood states derived from large scale Twitter feeds are correlated to the value of the Dow Jones Industrial Average

(DJIA) over time. They analyze the text content of daily Twitter feeds by two mood tracking tools, namely OpinionFinder that measures positive vs. negative mood and Google-Profile of Mood States (GPOMS) that measures mood in terms of 6 dimensions. They use a fuzzy neural network and granger causality analysis to investigate the hypothesis that public mood states, as measured by the OpinionFinder and GPOMS mood time series, are predictive of changes in DJIA closing values. Experiments indicate that the accuracy of DJIA predictions can be significantly improved by the inclusion of specific public mood dimensions but not others.

Additionally, a few commercial tools have emerged in this area. Table 3.4 gives an overview over what they analyze and which features they provide. Unfortunately, not all of the tools provide an insight into what methods they use exactly. Neither do they provide much information via publicly released test series, research papers or conclusive evidence of their accuracy. Therefore it is difficult to evaluate their performance. What is clear though is that they mostly concentrate on financial news releases.

Only one tool, *The Stock Sonar*[9] actually includes social media and also backs up the product with scientific research. In their paper, Feldman, Rosenfeld, Bar-Haim, and Fresko (2011) present the mechanisms underlying their tool. As one of only few researchers, they design a hybrid system that integrates sentiment lexicons, phrase-level compositional patterns, and predicate-level semantic events. They classify in-text sentiment as well as summaries for a given stock. To enhance the precision, they extract business events from news articles. Overall, their method reaches an accuracy of about 62%. Figure 3.2 shows how the results are graphically presented to the user.

Another tool named *Opfine*[10], on the other hand, uses only the most simple lexicon based approach by assigning positive/negative scores to words found in a news article. From these scores, ratings of news compiled into a time/mood graph as depicted in Figure 3.3.

Compared to the two tools mentioned above, the *Dow Jones News Analytics*[11] offers an extensive coverage of analyzed companies. On the website[12], some research papers are provided, although they appear to be more about market movements and influence of news in general and do not introduce the methods used for implementation.

[9] The Stock Sonar: http://www.thestocksonar.com
[10] Opfine: http://www.opfine.com
[11] Dow Jones News Analytics: http://ravenpack.com/services/rpna_dj.htm
[12] Dow Jones News Analytics, Research Papers: http://ravenpack.com/research/resources.htm

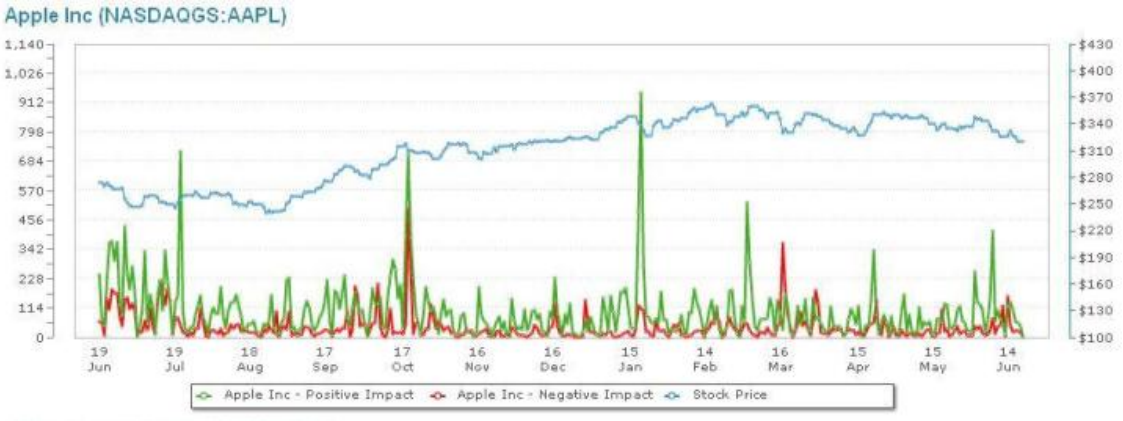

Figure 3.2: Sentiment graphs for companies in The Stock Sonar. Source: The Stock Sonar

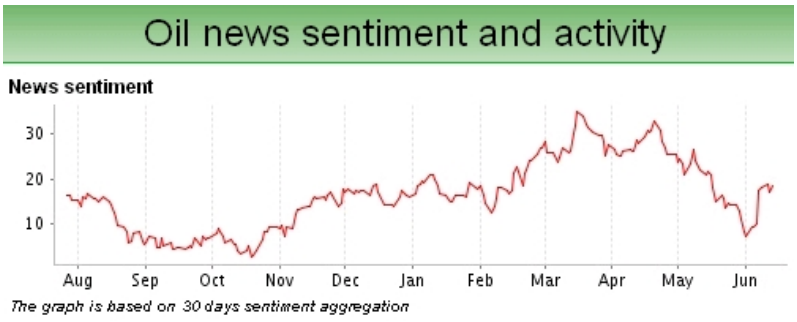

Figure 3.3: Sentiment graphs for commodities in Opfine. Source: Opfine

Tool	Media	Methods	Features	Classification categories
Opfine	News articles	Simple lexicon based techniques	Scanning and rating of news articles from over 3000 companies with hourly updates on the mood. Graphs for individual companies, grouped by commodity or by country as well as a macro view of the whole market.	Positive, negative
The Stock Sonar	News articles, blogs, press releases	Hybrid system (lexicon based, phrase-level patterns, semantic events). Note: All of the methods used in the hybrid are linguistic based. No machine learning techniques are involved.	Sentiment scores for public companies trading on the US stock market, including a daily overall score, a positive as well as a negative score (weighted sum of all positive/negative nuances and events extracted from that day's news articles)	Positive, negative
Dow Jones News Analytics	News articles	Possibly hybrid, at least several classifiers for different tasks including simple lexicon based techniques	Quantifies Dow Jones News in terms of company-specific sentiment (28.000 companies), relevance (does the information concern the company in question), and novelty (new information or update). Detection of 330 different market-moving events.	Positive, negative, neutral

Table 3.4: Commercial tools

3.7 Summary

Based on the review of previous literature and conclusions, several important research gaps can be identified. Firstly, there has been limited previous sentiment analysis work on data retrieved from real web forums. Few works integrate the resource of SentiWordNet and even then, little emphasis has been placed on context sensitivity as described in Section 3.4.1. Most studies have focused on evaluating and comparing single methods. While knowledge-based methods have the distinct advantage of an easy concept and implementation, they lack the capacity to model more complex structures. There are several suggestions as to how these approaches could be made more sensitive, but all of them require in-depth knowledge of linguistics and thus the constructed classifiers are limited to the one language they have been modeled for. Furthermore the lexicons used are very dependent on the domain they have been built for, thus rendering the classifier useless when applied to another domain. On their own, these techniques are not likely to achieve high accuracies.

Machine learning approaches seem to be better suited to the challenge due to their use of statistical techniques. These methods do not take any syntax or semantics into account but focus on the general statistical features of the input data. On their own, they perform well, but have the drawback of needing an annotated corpus. Results of classifiers from these two approaches can be combined to further improve the accuracy of the result. Hybrid methods introduced by the research community so far are usually based on integrating lexical knowledge into the features of a machine learning technique. The few hybrids that include more than two classifiers use simple voting strategies to combine results, thus dismissing important information about the degree of confidence in the correctness of a single classifier.

Furthermore, most of the previous work has been conducted under the assumption that all input is either clearly in favor of the topic or clearly not and as a result focused on two categories only: *positive* and *negative*. Contrary to that, the stock market sector demands for three categories to correctly mirror investor behavior: *buy*, *sell* and *hold*. Additionally to the restriction to two categories, hardly any studies move away from the few available corpora in the domain of movie and product reviews. These corpora are further discussed in Section 5.4 and compared to the new corpus for the stock market domain introduced in this thesis.

Last of all, although some properties of the financial sector are very similar to other domains such as the movie review, there are numerous characteristics such as the need for something other than a binary approach, the high volatility of the market as well as the fact that the quality of the data available for the domain is not great that are quite unique. Regardless, only minimal research has been conducted in the stock market domain. The project designed and implemented for this thesis tries to address these issues. The next chapter gives an overview of the tasks required of the system, its design and architecture, as well as an introduction to the approaches chosen for implementation and to the tools and libraries used.

4 Design and System Architecture

As can be concluded from the analysis of research in the specific area of stock market data, there is definitely a need for experiments with more complex systems. Researchers as well as the commercial tools mostly use very simple lexicon based approaches with the main concern of speed in order to cope with the flood of information. Additionally, there seems to be a major tendency towards analysis of news articles and similar data. Only few experiments include social media.

The goal of this project is therefore to focus entirely on social media, particularly on posts in forums and to test whether a more complex system is able to outperform simpler approaches by combining the "knowledge" of various different techniques and increase the overall accuracy of such a classification.

4.1 Requirements and Basic Concept

The goal of this thesis is to design, implement and evaluate a system for sentiment analysis of stock market forum data that:

- mirrors the structure and unique requirements of the stock market domain;
- allows to filter the corpus for relevant postings;
- provides the possibility to preprocess the data to identify sentence boundaries, negation, part of speech and spelling mistakes;
- makes use of available resources such as lexicons or thesauri;
- evaluates and tests a range of different classifiers;
- combines the classifiers to a hybrid system; and
- applies different ways of combination of evidence.

Adapting to the unique structure of the stock market domain requires moving from the traditional binary classification good/bad towards a more fine-grained approach. This leads to three categories for classification: *buy*, *sell* and *hold*. The traditional and most widely used *positive*/*negative* separation does not seem sufficient to correctly mirror the sentiment regarding a stock. In a broader sense, a *hold* recommendation could be interpreted as a positive opinion, however, a suggestion to not sell a stock is definitely not equal to a *buy* sentiment. Classification is done on

document level, although the system is designed to allow for an easy extension to classification of the overall sentiment for a single stock.

The basic idea is to implement, test and evaluate a number of classifiers suitable for sentiment analysis. These classifiers range from very simple to more sophisticated and complex supervised and unsupervised methods. Taking the most promising classifiers, the goal is to design a hybrid system out of the ones with the best performance, where performance might not only denote the accuracy of the classification, but also the speed and memory consumption. There are numerous ideas as to how to best combine single classifiers in the hybrid system and how to signify their importance, starting from a simple voting system over a weighted sum to more advanced methods such as Dempster-Shafer theory. For an overview over the combination of evidence from multiple sources refer to section 4.2.5 as well as section 8.2 for a more detailed description. Conducting a thorough evaluation will show whether it is a valid assumption that a hybrid system can significantly improve the accuracy of the classification reached by single classifiers and produce acceptable results even when the single classifiers do not do exceptionally well on their own. As methods with similar behavior might result in similar classification results (similar false positives and negatives), the hybrid should ideally consist of classifiers that have a wide range of variety, meaning that all the classifiers used in the system implement an approach that differs from the others. A greater variety in approaches might improve the overall quality.

4.2 Conceptual Architecture

As depicted in Figure 4.1, the hybrid system is composed of several modules. The basis of all operations is the London Forums data corpus. The postings have been stored in a PostgreSQL database after performing a number of necessary preprocessing steps which are described further in Section 4.2.2. A module of single classifiers forms the core of the hybrid system. They are responsible for deciding to which of the three categories *buy*, *sell* and *hold* a posting belongs to. Since every classifier provides their own set of probabilities, all these values have to be combined to a single answer. A separate module responsible for handling the combination of evidence provides various options, from simple voting systems to complex Dempster-Shafer theory.

All classifiers except for the lexical one are self-contained and do not require additional input. The implementation of the lexical classifier splits into two variations. The first one uses SentiWordNet and has complex underlying methods, the second variation works with traditional sentiment lexicons. Several of these lexicons with slightly varying characteristics have been acquired for testing in order to determine which one yields the best results and how the different properties influence the accuracies.

4.2.1 Data Management

In order to implement the proposed system, only a very simple database design is necessary. The core of the database is the table depicted in 4.2 that stores the data from the London Forums Corpus (a detailed description of the corpus can be found in chapter 5). One entry represents one

posting of the forum, with a reference to the stock it belongs to. The third and last table stores the sentiment of a certain posting as predicted by a classifier. Although this thesis only conducts experiments concerning the sentiment classification of single postings, the database is set up to enable future work like the classification of a whole stock by relating each posting to a certain stock. For more ideas on future work please refer to chapter 10.

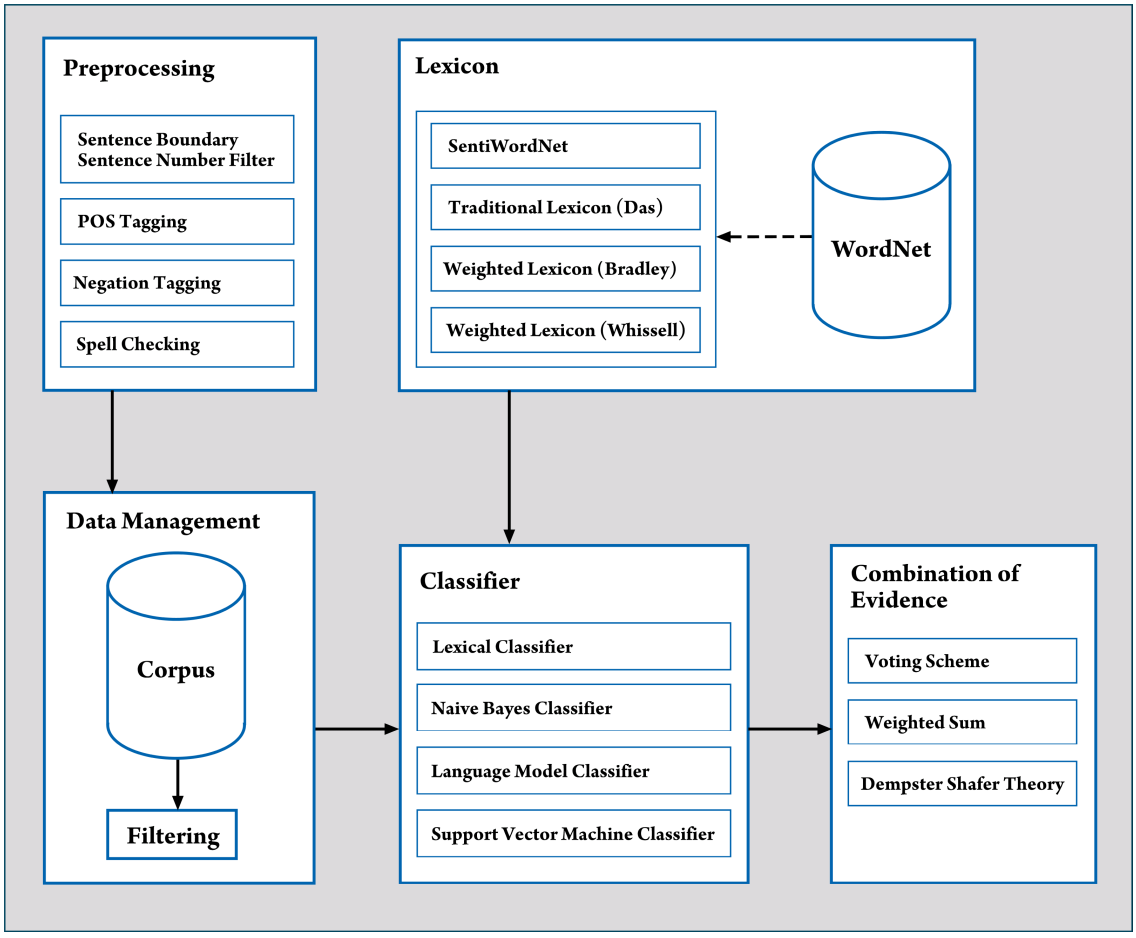

Figure 4.1: A hybrid system - design overview.

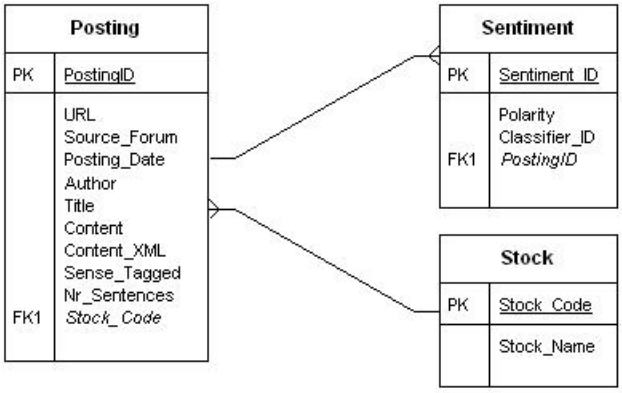

Figure 4.2: An ER model of the London Forums database.

4.2.2 The Preprocessing Module

Various preprocessing steps need to be performed, either on the fly for every posting during classification or beforehand. In order to cut down the time demands of the classification process, all preprocessing is done beforehand and results are saved in the database. Due to reasons detailed in Section 5.3.1, a *sentence filter* is applied. Using LingPipe (see Section 4.3.8) for *sentence boundary detection*, all postings with less than two sentences are discarded in order to ensure that only texts providing enough context for classification get stored in the database. The number of sentences is saved additionally for later use, such as being able to have varying training runs according to the samples' length.

Researchers such as Benarama, Cesarano, and Reforgiato (2007), Rittman and Wacholder (2008) and Hatzivassiloglou and Wiebe (2000) suggest that some parts of speech (POS) carry more sentiment than others, especially adjectives and adverbs, sometimes nouns. Therefore *POS tagging* is performed on every posting and stored in the database additionally to the raw text. This allows to filter out any part of the text that does not contribute to the identification of a category.
In the context of sentiment analysis, *negation* can have a big influence on accuracy. According to Wiegand et al. (2010), traditional lexicon based methods as well as supervised approaches both benefit from taking negation into account. Therefore negations in the raw text are identified and tagged to allow for later processing. More details about the importance of negation tagging can be found in Sections 6.3.1 and 7.6. The lexical classifier using SentiWordNet requires *spell checking* for the two underlying Word Sense Disambiguation services. Although this is actually done during classification and not beforehand as the other preprocessing operations, it still belongs the same module.

4.2.3 The Classifier Module

The classifier module consists of four different types of classifiers. Each of them has several different variations, with the one yielding the best results being selected for the hybrid system. Chapter 6 deals with the lexical classifiers, while all other approaches are discussed in Chapter 7. The following classifiers are part of the module:

Traditional Lexical Classifier
This classifier makes use of a so-called sentiment lexicon to identify words and phrases carrying sentiment (Pang and Lee, 2008). Using more or less complex aggregation functions, the positive and negative indicators are summed up to provide the probabilities of the posting belonging to a certain category. Four different lexicons have been acquired to test and evaluate, one of which has been selected for the hybrid system.

Naive Bayes Classifier
The first of three classifiers in the area of supervised learning, the naive Bayes (NB) classifier, has been implemented using LingPipe. The naive Bayes classifier technique is based on based on applying Bayes' theorem with strong (naive) independence assumptions and is particularly suited when the dimensionality of the inputs is high (Zlotnick, 1970). Independence assumptions

means that the classifier assumes that the presence of a particular feature of a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Despite its simplicity, naive Bayes can often outperform more sophisticated classification methods.

Language Model Classifier
Originally coming from the area of speech recognition, *statistical language models* (SLM or LM) become more and more important in various other natural language applications such as machine translation, part-of-speech tagging or text to speech systems. The goal of SLM is to build a model that is able to estimate the distribution of natural language as accurately as possible, where the model is a probability distribution P(s) over tokens S (i.e. a word) that attempts to reflect how frequently a token S occurs as an input entity (i.e. a sentence). A great introduction into the topic of language models in general and in the context of information retrieval in particular can be found in Zhai (2008). Contrary to the traditional lexical approaches to sentiment analysis, a LM classifier expresses various language phenomena in terms of simple parameters in a statistical model instead of trying to build upon knowledge of the linguistic structure. Thus, LMs provide an easy way to deal with the complexities of natural language. Both the LM and the NB classifier have been trained and evaluated with two different input tokens: whole words, and single characters.

Support Vector Machine Classifier
Support vector machines (SVM) are based on the concept of decision planes that define decision boundaries and can be used for classification, regression or other tasks. More formally, a SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space that optimally separates the data into two categories (Hsu, Chang, and Lin, 2003). The idea of using a hyperplane to separate the feature vectors into two groups works well when there are only two target categories, but gets more difficult when the target variable has more than two categories as is the case for this project. Several approaches have been suggested, but two are the most popular:

1. *"one against many"* where each category is split out and all of the other categories are merged; and,

2. *"one against one"* where k*(k-1)/2 models are constructed (k ... number of categories).

For this project an implementation called LibSVM (see Section 4.3.9) has been chosen, as it has the capability of multi-class classification.

4.2.4 The Lexicon Module

Consisting of four lexicons, this module handles all the input and operations necessary for the lexical classifier. For the variation of the classifier using SentiWordNet, *word sense disambiguation* (WSD) is necessary. Although the actual WSD happens during preprocessing due to the excessive time and memory consumption of the acquired tools, it still belongs to this module. It logically belongs to and was planned as part of the classification process in order to provide the input for

the lexical classifier and can be used that way in case options for a faster WSD emerge. Until then, WSD is done beforehand to save time during experiments.

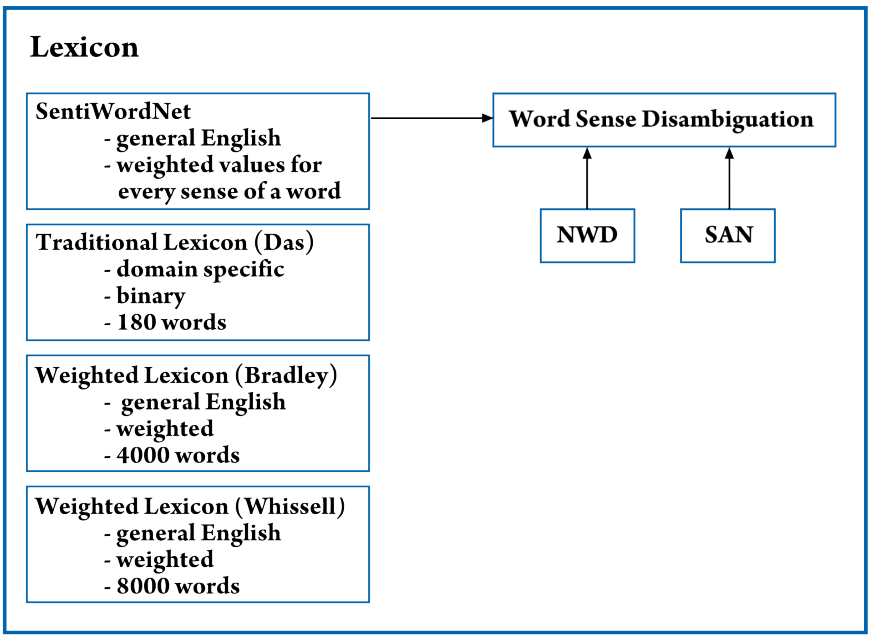

Figure 4.3: The selection of lexicons for unsupervised classification.

4.2.5 Combination of Evidence

As depicted in Figure 4.4, this module provides three different approaches to the combination of the results of the classifiers. The most intuitive method is a voting scheme, where each classifier simply states which category a posting belongs to, where the one with the most votes wins. This has several drawbacks as detailed in Section 8.2.1 which enforce a limit of two categories. Discarding the category hold leaves only buy and sell, thus submitting to the limitation and making a voting system possible. Since the system implemented within this thesis is unique in having three categories instead of a simple binary positive/negative, other means of aggregation are necessary. As most of the classifiers except for some variations of the lexical classifier provide actual probabilities for each category instead of simple "yes" and "no" answers, it might be a good idea to include this level of detail in order to improve the accuracy of the result.

The next intuitive step is to assign weights to the classifiers symbolizing their trustworthiness. One possible way to determine these weights is to check how well the do on their own and use that accuracy as a degree of importance. Weights could be improved by applying Linear Regression as it would optimize the weights for a certain collection of data (Montgomery, Peck, and Vining, 2001). Though this would most likely result in an improved overall accuracy of the hybrid classifier when tested on the corpus it has been trained on, the idea has been discarded due to reasons stated in Section 8.2.2.1. Since this project aims to test the system's performance in an environment that is as close as possible to real world data, using linear regression would contradict this requirement.

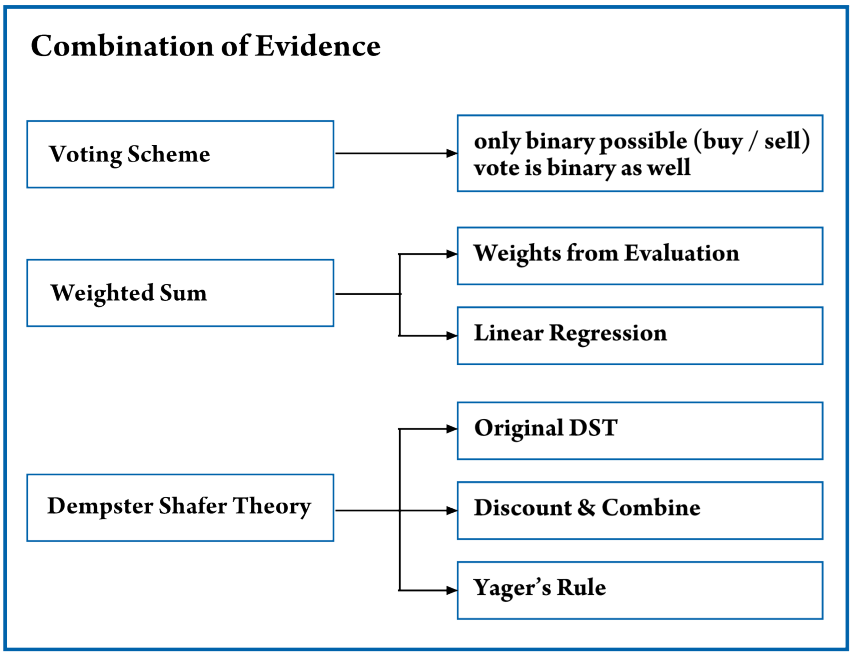

Figure 4.4: Methods for the combination of evidence.

The third approach to combine results is the most promising and complex one. Using variations of the Dempster-Shafer theory, no assumption regarding the probability of the individual constituents of the set or interval is required. This makes it a valuable tool for the combination of the results of different classifiers, where it is not possible to obtain a precise answer with all the classifiers agreeing on one category or a certain classifier even having an assumption about a certain category at all. For an introduction into the topic refer to Sentz and Ferson (2002).

4.3 Tools, Libraries and Services

A number of freely available sources have been used for the implementation of the system as described in the previous sections. The following subsection gives an overview of all tools, libraries and services that contributed to the project.

4.3.1 PostgreSQL

PostgreSQL[1] is an object-relational database management system (ORDBMS) which is released under an MIT-style license and is thus free and open source software. As with many other open-source programs, PostgreSQL is not controlled by any single company; a global community of developers and companies develops the system. It has a strong reputation for reliability, data integrity, and correctness and runs on all major operating systems, including Linux, UNIX, and Windows. PostreSQL is fully ACID compliant, includes most SQL:2008 data types and provides

[1] PostgreSQL: www.postgresql.org

native programming interfaces for numerous languages including for example C/C++, Java, .Net, Perl, Python or Ruby. For these reasons as well providing a good documentation, PostgreSQL has been chosen for this project to store the raw data of the corpus and the preprocessed and classified postings.

4.3.2 Hibernate

For mapping from Java objects to the database, Hibernate[2] has been used. Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. It solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions. The primary features are mapping from Java classes to database tables (and from Java data types to SQL data types), session management, data query and retrieval facilities. Hibernate is free software that is distributed under the GNU Lesser General Public License.

4.3.3 WordNet and JWNL

WordNet[3] is a large lexical database of English, developed under the direction of George A. Miller. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. In this project it is used to determine all possible meanings for a given word. These meanings are needed to allow the disambiguation of different meanings of the word in a given context. This sense tagging of words is necessary to identify the polarity of a given input text using SentiWordNet. JWNL[4] provides a Java interface for WordNet.

4.3.4 SentiWordNet

SentiWordNet[5] is a lexical resource for opinion mining. It annotates of all WordNets synsets with three numerical scores indication how positive, negative or objective (neutral) the terms contained in the synset are. Different senses of the same term may thus have different opinion-related properties. Each of the three scores ranges in the interval [0:0; 1:0], and their sum is 1:0 for each synset. Figure 4.5 shows a graphical representation of these opinion-related properties of a term. The idea of distinguishing between different senses a word might have in different contexts has the potential to increase the accuracy of a traditional lexical classifier by far. For example, the term "cancer" can be associated with very different meanings. If we are talking about

[2] Hibernate: www.hibernate.org
[3] WordNet: http://wordnet.princeton.edu
[4] Java Word Net Library: http://sourceforge.net/projects/jwordnet
[5] SentiWordNet: sentiwordnet.isti.cnr.it

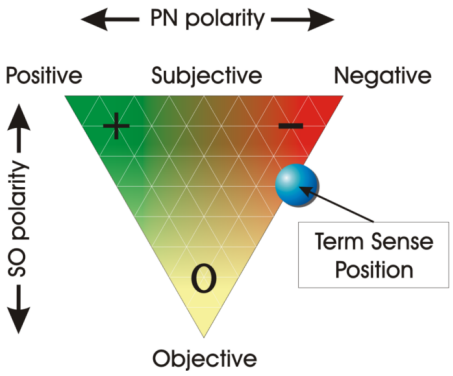

Figure 4.5: The graphical representation adopted by SentiWordNet for representing the opinion-related properties of a term sense. Source: Esuli and Sebastiani (2006b)

the animal, the word is rather neutral, however, if we are talking about the illness, the word is associated with a lot more emotion. For this thesis, a lexical classifier using SentiWordNet as input has been implemented. Version 1.01 has been used as version 3.0 has not been available at that time. SentiWordNet is further described in the papers of Esuli and Sebastiani (2006b) and Esuli, Baccianella, and Sebastiani (2010).

4.3.5 George Tsatsaronis' SAN WSD

For the disambiguation of word senses, a tool developed by Tsatsaronis, Vazirgiannis, and Androutsopoulos (2007) has been integrated into the system. It uses *spreading activation networks* to determine the sense of a given term depending on its context. Disambiguation is necessary for the lexical classifier using SentiWordNet in order to retrieve the correct sentiment values.

4.3.6 Wilson Wongs NWD

As a second tool for word sense disambiguation, Wong, Liu, and Bennamoun (2008) have kindly provided their implementation of the *normalized Web distance*. This approach is completely different from Tsatsaronis' method. The idea is to combine the results of two different techniques to increase a correct labeling of the input terms.

4.3.7 JOrtho

JOrtho[6] (Java Orthography) is an open source spell-checker entirely written in Java. Its dictionaries are based on the free Wiktionary project and can therefore be updated for virtually any language. JOrtho is used to check the spelling of terms before doing Word Sense Disambiguation on them. This is necessary because both tools mentioned above give incorrect or no results if the word is unknown or does not exist.

[6] JOrtho: http://jortho.sourceforge.net

4.3.8 LingPipe

LingPipe[7] is a collection of Java libraries for processing text using computational linguistics. It can be used to implement tasks like *named entity detection* (finding the names of people, organizations or locations in news), *text classification* (by language, character encoding, genre, topic or sentiment), *part-of-speech tagging*, *sentence boundary detection* or *spell checking*. The latter three features have been used for preprocessing in the project, and different versions of language model and naive Bayes classifiers have been implemented using LingPipe as a basis. Apart from providing a foundation for training models, it also provides a framework for evaluation.

4.3.9 LibSVM

LiSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification as required for this project and provides a simple interface where users can easily link it with their own programs. Main features of LibSVM include different SVM formulations, support of cross validation for model selection, probability estimates, various kernels and weighted SVM for unbalanced data. The latter might be used to integrate all of the highly unbalanced London Forums Corpus in future work. Both C++ and Java sources are available as well as interfaces to various other languages. Apart from that, a GUI demonstrating SVM classification and regression and a tool for automatic model selection which can generate contour of cross validation accuracy are an asset to users not overly familiar with SVMs.

4.4 Summary

The design of the system strives to cope with the research gaps identified in the previous chapter. Most of the work done in the specialized area of the financial market are focusing on news articles, and research as well as the commercial tools available use simple lexicon based or machine learning methods. Not many try to combine techniques into a more complex system in order to exploit the advantages of single classifiers to overcome the disadvantages of others. Those who do usually use very simple voting mechanisms, and do not test whether more complex forms of combination of evidence could lead to an improvement of accuracy.

The architecture of the system has been designed to be as modular as possible. Minimization of dependencies between separate modules and classifiers allow to easily extend the system with new classifiers, integrate different preprocessing steps and include any number of classifiers into the hybrid system in order to cope with future expansions and modifications. Each classifier can be trained, tested and evaluated separately as well as part of a combined classifier. Several tools, libraries and services have been evaluated and selected for use in the practical part of this project.

[7] LingPipe: alias-i.com/lingpipe

The area of traditional lexical classifiers has been given particular consideration, especially the combination of SentiWordNet and word sense disambiguation. The methods chosen for implementation range from traditional lexicons to a number of supervised methods including naive Bayes, language models and support vector machines, thus ensuring to meet the demand for variety.

5 Corpus

The term corpus in the context of data mining and natural language processing denotes a usually very large and structured collection of texts that are used for hypothesis checking, training of models and statistical analysis. The prototype of the hybrid classifier implemented in this thesis includes various methods that require the training of models and their statistical evaluation. The use of a corpus is therefore unavoidable.

5.1 Use of a Corpus in Unsupervised Approaches

According to Pang and Lee (2008), the term unsupervised in the context of sentiment analysis describes mainly lexicon-based approaches. These methods focus on the syntax of the text by scanning it for words indicative for a certain class. For example, the word "good" could be taken as an indicator for the category *positive*. Small refinements like taking negation ("not good") into account or applying some sort of weighting scheme improve these methods further. Still, none of these approaches are in need of training and therefore do not require labeled corpora. Any corpus consisting of domain specific texts would be enough. However, to be able to compare all methods used in this project, all of them work with the same corpus as described in Section 5.3.1.

5.2 Use of a Corpus in Supervised Approaches

The term supervised learning summarizes methods that use a corpus in order to generate a mapping between input and output data, where the latter are the desired results. The learning process is supervised insofar as that for every input sample (in this case the raw text posting) there is a result (the opinion it expresses, the class it represents), the so-called label which can be used to measure the error of the model. Supervised approaches require a corpus to have a certain attribute: it has to provide these results - it has to be annotated. Annotating a corpus means that for a certain classification scenario, each sample (e.g. text) in the collection has to be assigned a label declaring which class it belongs to. Taking the stock market forum domain as an example, each posting has to get annotated by a label stating whether it expresses a *buy*, *sell* or *hold* opinion.

Annotation can be done either automatically, semi-automatically or manually. Whereas it is obvious that using the last option takes a lot of work and time in order to provide a sufficiently large corpus, it is also clear that it most likely provides the highest quality. Automatic annotation on the other hand is a quick way to get a huge corpus, but can be error-prone for certain annotation

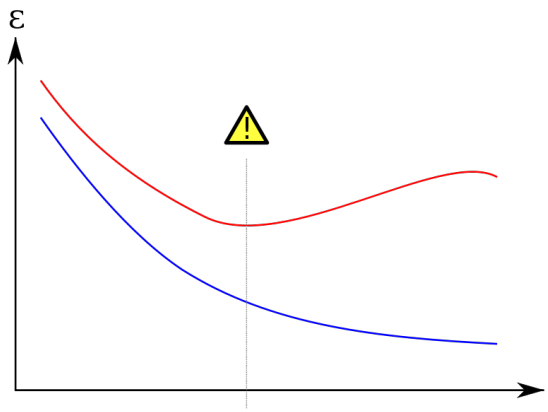

Figure 5.1: Overfitting. The red curve is the error on the validation set over several epochs. The blue curve is the error of the training set. Source: Wikipedia

tasks. As a compromise, semi-automatic approaches combine automatic labeling with manual checking or use a small manually labeled base to expand by automatic processes.

5.2.1 Training, Testing and Validation in Supervised Approaches

Supervised methods train models by feeding a number of samples into them and using the labels to update the values of certain attributes describing a class. In order to achieve a well-trained model, several points have to be kept in mind (Whiteson, Tanner, Taylor, and Stone, 2011):

1. The samples used in training ideally cover the whole domain. This is not always easy to provide, but it is essential to avoid overfitting. Furthermore, training with samples from outside the domain usually reduce the accuracy of classification significantly.

2. Overtraining should be avoided. The term overtraining or overfitting describes the problem that occurs when a statistical model describes random error or noise instead of the underlying relationship. Usually a model is trained using some set of training samples. It is assumed to reach a state where it will also be able to predict the correct output for unknown samples, thus generalizing to situations not presented during training.
   By training too long or using only rare samples as discussed before, the model may adjust to very specific random features of the training data that have no causal relation to the target function. In this process of overfitting, the performance on the training samples still increases while the performance on unseen data becomes worse as depicted in Figure 5.1.

In order to test and verify the results of a classifier, the corpus is usually split into two or three parts. The larger one is used to train a model, the smaller one to test the trained model. In some cases it is interesting to have a third portion of the corpus available for verification of the test results. This splitting of the corpus is called *cross-validation* (Kohavi, 1995) and guarantees that the classifiers are tested on samples not seen during training. This is necessary to correctly measure its performance and its ability to generalize on unknown samples. For the measurements

of the accuracy of the classifiers used in this project the corpus has been split into a training and testing section while the verification part has been left out in order to avoid diminishing the small amount of samples even further. Using a 10-fold cross validation, 90% of the corpus has been used for training, leaving the remaining 10% for testing, respectively.

5.3 The London Forums Corpus

Theoretically, any corpus consisting of natural language texts would suffice. Considering that the methods chosen for the prototype and in need of training are all supervised, in practice the corpus is required to be annotated. Henry Leung from the Business School of the University of Sydney kindly provided a corpus of stock market forum data for this research. The corpus is a collection of postings, retrieved from forums on the London stock market, covering a timespan from 1998 to 2008 and consisting of two subsets: A number of postings from the III Forums (see Section 5.3.1), and a number of postings from the ADVFN Forums (see Section 5.3.2).

5.3.1 Postings from the III Forum

The III[1] part of the data is annotated with labels for each posting. Possible sentiment labels are *strong_buy*, *weak_buy* (positive), *hold* (neutral), *strong_sell*, *weak_sell* (negative), and *null* (no label). The labeling has been done manually by the author of the respective posting and is optional, which means there are a lot of postings with sentiment *null*. Postings with sentiment *null* have been considered to be objective and have been discarded, as they might pollute training or at least do not contribute to it in any way due to their missing classification. However, when implementing a module to do subjectivity detection as described in Section 2.2, these postings might come in handy. Although five categories might be great to have for refinement of classification in future work, for this project *strong_buy* and *weak_buy* have been accumulated to a category *buy*, while *strong_sell* and *weak_sell* form a general category *sell*. This reduces the complexity to a more manageable size and at the same time guarantees a sufficient number of training samples per category.

Correctness of the Labels

The sentiment labels do not always correctly reflect the content of a posting. An example for such a problematic label can be seen in Figure 5.2. While a human might still be able to interpret that a repetition of "888" in combination with the stock code "888.L" as an enthusiastic exclamation in favor of the stock, a computer program has no ability to come to such conclusions. Using a lot of input samples such as this in training might significantly decrease accuracy of a model. Unfortunately, there is not much that can be done about wrong or problematic sentiment labels other than manual correction. In view of the short period of time available for the project, this drawback had to be accepted with the conviction that the amount of wrongly labeled postings is

[1] III forum: http://www.iii.co.uk

```
<posting>
    <source>III</source>
    <stock_code>888.L</stock_code>
    <author>JenningsSmyth</author>
    <title>Dawn of the new age</title>
    <content>888 888 888 888 888 888</content>
    <sentiment>Strong BUY</sentiment>
</posting>
```

Figure 5.2: Example for a posting with a problematic label

```
<posting>
        <source>III</source>
        <stock_code>888.L</stock_code>
        <author>rogk</author>
        <title>Monday</title>
        <content>Monday will be interesting.</content>
        <sentiment>Strong BUY</sentiment>
</posting>
```

Figure 5.3: Example for a short posting

not big enough to have a significant effect on the classification results of the system.

Sentence Filtering

As it is the nature of postings in forums, the corpus contains a great number of very short postings that are labeled but do not provide enough context for training or classification. An example of such a posting can be seen in Figure 5.3. Using these postings could significantly decrease the accuracy of supervised classifiers when used in training. Therefore, postings with less than two sentences have been discarded using LingPipe for Sentence Boundary Detection (SBD). This serves not only the purpose of cleaning the data of very short and insignificant postings, but might have also had the useful side effect to get rid of some of the postings written in very heavy slang, as the model for SBD is more likely to be unable to identify a sentence structure in such a case.

Experiments with, for example, dynamic language models and naïve Bayes classifiers show that using only longer postings significantly improve the classification accuracies. Evaluation with 1000 samples per category using a varying minimum number of sentences per sample result in an increase of accuracy the longer the postings are (the more context is provided in the training). More on these test results can be found in Section 7.

Data Balance

The corpus has a very bad distribution over the three categories buy, sell and hold. There is a huge overhead of buy instances compared to sell and hold instances combined. The amount of negative samples is significantly smaller than the number of postings in the other two categories.

Of the 112907 samples, there are:	
76916 samples	68% for category buy
26078 samples	23% for category hold
9913 samples	9% for category sell

After filtering the corpus for all postings with at least eight sentences, the following numbers of samples are left:

Of the 6002 samples, there are:	
4363 samples	70% for category buy
1111 samples	21% for category hold
528 samples	9% for category sell

In order to get correct results, all the training and testing has to be done using balanced sets of data, meaning an equal amount of instances for every category. Balancing the sets as well as using sentence filtering reduces the size of the corpus significantly. Using a balanced set with the above mentioned sentence filter leaves a subset of 1029 samples per category from a corpus of over 112000 samples. An even smaller subset of the corpus might not be able to suffice to get well-trained models any more. That means although accuracy might be improved further by raising the number of sentences when filtering, the number of instances would decrease to a level where no meaningful evaluation can be conducted. The data is stored in a PostgreSQL database and accessed via Hibernate (see also Sections 4.3.1 and 4.3.2).

5.3.2 Postings from the ADVFN Forum

The ADVFN[2] part of the corpus is in the same format as the III data, but does not contain any sentiment labels. It has not been used in this project, but it might be of interest for future research that employs semi-supervised methods such as Generative Models like Hu, Lu, Chen, and Duan (2007a) or Transductive Support Vector Machines. Wang, Shen, and Pan (2007) give a detailed overview over the TSVM's and which problems it can be applied to, while Joachims (1999) introduces them for text classification and provides a detailed analysis of their suitability for this specific task.

In their paper, Xu and Zhou (2007) propose a method based on a TSVM for personalized spam filtering showing that the results of filtering are better than using a traditional SVM. Due to the restrictions and shortcomings of the labeled part of the corpus as mentioned in Section 5.3.1, its size after all the necessary filtering is quite small. It is quite possible that including the unlabeled part of the corpus using methods stated above might improve results significantly.

[2]ADVFN forum: http://www.advfn.com

5.4 Comparison to Other Available Corpora

Labeled corpora available for the task of sentiment analysis are scarce. Only very few have been created so far and to the best knowledge of the author, none of them except for the London Forums corpus covers the financial domain. As mentioned in Section 5.2.1 it is crucial to train on a corpus fitting the classification domain. Few of the research in sentiment analysis specializes in the stock market area, Das and Chen (2007) and Devitt and Ahmad (2007) being some of the best known examples. Most of the research in the field of opinion mining seems to focus on the context of movie or product review domains. This might result from the lack of annotated corpora in other domains as well as the challenge to create one. In the product review domain there are a lot of web forums or similar systems that provide a sufficiently large number of user-annotated reviews. This manually provided rating tends to make it easier to collect labeled data. To the authors best knowledge, the corpus that seems to be used in almost all projects is the freely available Movie Review database[3] introduced by Pang, Lee, and Vaithyanathan (2002) at the 2002 Conference on Empirical Methods on Natural Language Processing.

In Table 5.1 the Movie Review database is being compared to the London Forums corpus in order to illustrate the main differences as well as the problems that may arise from them. The London Forums corpus is special in several ways: Unlike other corpora, it does not assemble a world with perfect samples ideal for training but consists of real world postings from existing forums. Instead of providing an optimal environment, it challenges with harsh conditions. Working with such a corpus creates several problems that are difficult to overcome, but on the other hand offer the opportunity to get results that mirror the performance of a system in a much more realistic way.

5.5 Summary

When training and testing the machine learning classifiers, special care needs to be taken to avoid overfitting and thus loose the ability to generalize on unknown samples. The quality of the corpus used for training is essential for the accuracy of a classifier. Quality in that context means that:

- there is a sufficient number of samples,
- all samples are labeled with the correct class,
- all samples are in proper English without slang words or other grammatical or spelling errors,
- all samples are sufficiently long to provide enough context for feature extraction, and
- ideally, none of the samples use irony or sarcasm.

According to these criteria, the London Forums corpus that will be used in this project has very low quality. Especially a comparison to the Movie Review corpus used in most other projects highlights this. However, it also shows the advantage of providing unbiased real world data allowing a realistic measurement of accuracy, even though this leads to a number of problems which are difficult to

[3] www.cs.cornell.edu/People/pabo/movie-review-data/

overcome and in succession most likely to a decreased performance in classification. Several of the drawbacks of the corpus have been illustrated as well as possibilities and ideas to reduce their impact discussed.

Whereas other corpora in the area of sentiment analysis are fully labeled, the London Forums corpus provides an additional part consisting of unlabeled samples. Given the scarcity of high quality labeled corpora, this represents a potential that could be utilized with semi-supervised approaches. Such techniques might be able to successfully utilize that extra information contained in the unlabeled data and thus enhance the overall quality of the corpus. However, this is not part of the scope of this thesis.

Moving on from the corpus, the next two chapters offer a detailed description of both unsupervised and supervised approaches making use of the labeled part of the London Forums data, discussing what methods were selected for evaluation and the results achieved.

Difference	Movie Reviews	London Forums	Problem
Number of categories	Two categories: *positive* and *negative*.	Three categories: *buy* (positive), *sell* (negative) and *hold*.	Using three (or potentially more) categories instead of a simple binary positive/negative classification means there are two borders that the system has to model and take into account. This increased complexity further complicates the classification process and makes it harder to achieve high accuracies.
Length of samples	Long texts consisting of a lot of sentences.	Mostly very short texts consisting of less than 3 sentences.	The shorter the training samples, the less context they provide for training. This lack of context causes the models to be less well trained as they do not get enough information to generalize patterns.
Use of language	All the samples are in proper English.	A lot of samples contain grammatical and spelling errors. Worse, they use inexistent (slang) words.	Training on samples using slang and wrong language decreases the accuracy of a model tremendously. The use of incorrect English prevents the system from providing a number of essential information such as part-of-speech to the model, causing it to rely on too few characteristics to be able to generalize correctly.
Use of irony and sarcsm	No use of either of them.	Heavy use of both of them.	Sarcasm and irony make it incredibly difficult to train a model as it is very hard to find characteristics to distinguish between a serious statement and an ironic one that might express the exact opposite opinion using syntactic and semantic features.
Correctness of labels	Labels are 100% correct.	Labels do not always correlate with the content of the texts, see 5.3.1	Incorrect labeling obviously decreases the performance of the classifiers.

Table 5.1: Comparison of the Movie Review Corpus and the London Forums Corpus

6 Unsupervised Methods

6.1 Overview and Definitions

According to the definitions of Pang and Lee (2008), unsupervised methods in the context of sentiment analysis are generally lexicon-based approaches that use some aggregation function on the occurrence of so-called sentiment words in order to determine the category of an input sample. The methods of aggregation range from a very simple counting of occurrences of indicative words to more sophisticated ideas with modified counting based on dependencies and weighting of words.

Examples of such functions can be found in Lu, Tsou, and Kwong (2008), Hatzivassiloglou and Wiebe (2000) and Turney (2002). Other interesting variants of this general technique can be found in Hu and Liu (2004), where the polarity of the previous sentence is used for a decision when the function does not give a definitive classification for the current sentence, or in Sindhwani and Melville (2008) and Beineke, Hastie, Manning, and Vaithyanathan (2004), where information from labeled data is incorporated. As a basis, lexicon-based methods all use so-called sentiment lexicons, which are generally lists of positive and negative terms and phrases. The identification of such words carrying sentiment has been subject of many experiments. Further discussion on this topic can be found in Section 6.2.

Methods based only on lexicons are perhaps the most intuitive and hence rudimentary approach to sentiment analysis. Using simple counting functions that take the frequency of occurrence of words that are supposed to indicate a certain polarity, these techniques are not the most sophisticated solution. By itself, this approach might not yield highly accurate results as can be seen from the evaluation of experiments in Section 6.6. Combined with other methods where the lexicon serves as a basis or as one of several classification modules, however, it still might be able to contribute to an improvement of accuracy.

Apart from using the lexicon directly for classification, some researchers have used lexicons along with unlabeled data in a semi-supervised learning setting. Liu, Li, Lee, and Yu (2004a) for example treat the whole list of positive (negative) words as a "representative document" and use the cosine similarity as a metric to calculate the distance of each text to classify to this representative instance. Each unlabeled document gets assigned the class with the highest similarity. Using this method, they get a corpus of labeled pseudo-samples that are in turn used for supervised training instead of trying for a direct classification.

Section 6.2 provides an overview over how to create lexicons manually or in a semi-automated manner. Section 6.3 compiles suggestions on the improvement of simple lexicons, and finally Sections 6.4, 6.5 and 6.6 discusses the methods chosen for implementation of a lexical classifier as well as the experimental results.

6.2 Creation of a Sentiment Lexicon

A number of papers deal with the problem of creating a traditional sentiment lexicon. Lexicons usually concentrate on adjectives, adverbs and sometimes nouns, since these parts of speech have been found to be the best indicators for sentiment. While Hatzivassiloglou and Wiebe (2000) state that adjectives are good indicators of polarity, Turney (2002) suggests that, although an isolated adjective might indicate a polarity, there may be insufficient context to determine the semantic orientation. Consider the following adjective in different contexts: "`unpredictable steering`" in an automotive review versus "`unpredictable plot`" in a movie review. Turney therefore strongly suggests to use tuples consisting of adjectives combined with nouns and of adverbs combined with verbs instead of isolated adjectives. More information on that topic can be found in Hatzivassiloglou and Wiebe (2000), Turney (2002), Rittman and Wacholder (2008) or Benarama, Cesarano, and Reforgiato (2007).

6.2.1 Manual Creation of Lexicons

In order to determine the sentiment of a word, various methods have been proposed. Early works usually use a manually generated lexicon, where people were asked to list positive and negative words. Some of them, as for instance Whissell's or Bradley and Lang's lexicons (see Section 6.4) consist of words rated in several different categories, thus providing even weighted values instead of a binary positive/negative assessment.

6.2.2 Semi-automatic Construction of Lexicons

To skip or at least reduce this tedious and time consuming work, several semi- or fully automatic approaches have been developed. The basis for most semi-automatic method are the so-called seed lists, which are expanded into a sufficiently large lexicon. The seed lists are lists of words and phrases annotated with their polarity with a size that takes only a reasonable amount of manual work for generation. By propagating the labels of the seed words to terms that co-occur with them in general text or dictionary glosses, or have some specified relations to them in thesauri, the polarity of new words can be determined. Examples of such methods can be found in the following sections.

6.2.2.1 Thesaurus-based Approach

A thesaurus most often used for this kind of expansion is WordNet, whose database consists of nodes (words) connected by edges (relations). Usually the members of the synonym and/or

antonym sets are taken as the related set. Whitelaw, Garg, and Argamon (2005) use several thesauri for the expansion. Taking into account only synonyms, new words are added to a candidate list, even if they are already on it. After completing the expansion, all candidate terms get ranked by their frequency of occurrence in order to provide a coarse ranking of relevance. This enables a more efficient manual selection, since uncommon words, unrelated words or words arising from an incorrect sense of the seed term will tend to occur less frequently and can be automatically discarded, which leaves a smaller list for manual inspection.

Kamps, Marx, Mokken, and De Rijke (2004) do not only determine a simple positive/negative polarity, but use more fine-grained levels of distinction. In order to do so, they use Osgood, Suci, and Tannenbaum's (1971) dimensions of appraisal. Kamps et al. (2004) define the Minimum Path Length (MPL) as a distance metric between words, which counts the number of edges on the shortest path. To estimate the polarity of a particular word and the polarity strength, respectively, they compare the MLP of that word towards the positive and the negative ends. Those ends are represented by the prototype words "good" and "bad". Godbole, Srinivasaiah, and Skiena (2009) expand their seed collections through antonyms and synonyms in WordNet. After creating a graph from the relations, they perform a thorough analysis of the paths from a word to the seed words. They enhance Kamps et al.'s (2004) metric by taking into account what they call *flips*. These flips describe apparent sentiment alternations along the path.

6.2.2.2 Web-search Approach

The use of co-occurrence of words in a general corpus with a small list of seed words is a method employed by a number of researchers. Initially, it was developed by Turney (2002), whose idea was to compare whether a word or phrase has a greater tendency to co-occur within a certain window with a predetermined positive word ("excellent") than with a predetermined negative word ("poor"). The data on which his computations are made come from the results of particular types of Web search-engine queries. For each word in the lexicon, two queries are issued in an arbitrary search engine: one query that returns the number of documents that contain the word from the lexicon close to the chosen positive word (close defined as "within a 10 word distance"), and a similar query with the chosen negative word. If the lexicon word is found more often in the same context as the chosen positive word, it is considered to indicate a positive orientation. Much of the work above focuses on identifying the polarity of isolated terms or phrases. Such prior identification is necessary in order to be able to further determine the contextual polarity, meaning the polarity of sentences, or on higher level, documents.

6.2.2.3 Game-based Approach

A very different approach to the creation of a lexicon is presented by Weichselbraun, Gindl, and Scharl (2011). In their paper, they propose a semi-automatic system which assigns sentiment values to sentiment terms via crowd-sourcing. This game based approach is realized with a simple Facebook game where users can earn points for rating individual words for its positivity. That way through a large number of players, a good evaluation and scoring of the lexicon words can

be achieved with only a minimum effort of the individual gamer. The big advantage is that through the large number of evaluating people, a good average opinion can be determined. The game characteristics with a simple reward system makes it easy to create sentiment lexicons for various languages and domains, as the only input required from the researchers is a list of selected representative words.

Furthermore, a bootstrapping process is introduced, which operates on unlabeled domain documents to extend the created lexicons, and to customize them according to the particular use case. This process considers sentiment terms as well as sentiment indicators occurring in the discourse surrounding a particular topic. Initially, a base sentiment lexicon is created by crowd-sourcing the task of annotating vocabulary with sentiment values. As a second step, this lexicon is used to perform sentiment detection on a number of unlabeled Web reviews. After identifying representative examples of reviews with a positive and negative sentiment and using them to create a corpus of such reviews, sentiment indicators and terms are extracted from this corpus and the terms merged into the basic lexicon. This process is then repeated until a satisfactory level of customization and accuracy is reached. Using this method, their experiments show that the created lexicons yield a performance comparable to professionally created language resources.

6.3 Improvement of a Lexicon

Since the lexicon-based approach does not usually achieve a very high accuracy, it is worth thinking about improvements to the basic lexicon. In the following sections, a few ideas are pointed out.

6.3.1 Negation Tagging

When it comes to the use of unweighted conventional sentiment lexicons as well as a weighted ones such as SentiWordNet, it is worth thinking about taking negation into account. Consider the following example:

"`This movie is funny but it is not at all good`"
"`This movie is not at all funny but it is good`"

These two sentences contain same words with same frequency but are of opposite sentiment polarity. Using only the words occurring in the lexicon, both statements would most likely be assigned the same polarity. Tagging negations in this contexts means to tag words with a $\_not$ (i.e. $good\_not$) if there is a negation found that corresponds to the word in question. This tagged word is then regarded as having reverse polarity.

There are different ways to reach that goal. One proposed by Pang, Lee, and Vaithyanathan (2002) to just tag every word following a negation as negative until the end of the sentence is reached. An approach at modeling negation more accurately is suggested by Na et al. (2004). They look at specific part of speech patterns and tag only words fitting one of them. An example for that is to search for negations and tag every following word as negative until an adjective is reached.

Taking the example above, the first method would tag the first sentence correctly, while it would not be sufficient to solve the second sentence, as it tags the word good as negated as well. The second method, however, is able to tag the negations in the second sentence correctly. Regardless of this outcome, the first method has been used in this thesis for the sake of simplicity.

Although it seems logical to include some sort of negation modeling into the classification process, it is not always as easy to determine negated parts as in the example above. Irony and sarcasm are an example of such a complication in the detection of negations. Further discussion about the complexity of negation in natural language can be found in Wilson, Wiebe, and Hoffmann (2005).

6.3.2 Further Modeling of Syntactic Features

Especially for lexical classifiers that concentrate on only the syntactic features of a sentence or text, a more detailed modeling than just negation might be of use. Adapting the tagging of negations to fit the syntactical structure of a sentence rather than negating all words until the next punctuation mark might improve the quality of the classification.

Additionally including other syntactical characteristics such as taking into account modifiers could lead to better results. An example for such a modifier would be "good" versus "very good". While an ordinary lexical classifier just takes the occurrence of the word "good" as a positive indicator into account, an improved version could also model the strength of sentiment. Although weighted sentiment lexicons as SentiWordNet or Whissell's lexicon do provide some measurement of sentiment strength, they still do not provide any inclusion of modifiers and might benefit from that.

6.4 Implementation of a Traditional Lexical Classifier

Using a lexicon as described above, an implementation of a traditional lexical classifier has been made. As aggregation function a simple sum of all values of the detected sentiment words has been divided by the number of all these words. The result is then interpreted in order to determine the category the sample in question belongs to. Evaluation has taken place with the following three different lexicons that have been kindly provided by the authors for use in this project:

1. **Bradley and Lang (1999)'s lexicon**: This is a manually generated lexicon, consisting of 1034 words of general English. The words have been labeled by an unknown number of human annotators in three categories: pleasure, arousal and dominance (Osgood, Suci, and Tannenbaum, 1971). The one most relevant for Sentiment Analysis would be pleasure, although the other two might be able to contribute to an improved classification accuracy. The words are rated on a scale from 1 to 8. That means this dictionary can be viewed as a simpler version of SentiWordNet, each word having a degree of positivity (weighted instead of binary *positive/negative*) and only one possible sense. Important to note is that, while the lexicon itself might be quite good, it is a lexicon for general English, whereas people commenting on topics of the stock market sector mostly use a very domain specific

vocabulary. Therefore this dictionary might not give as good results as a domain specific dictionary could probably give.

2. **Das and Chen (2007)** kindly provided the lexicon they used in their paper. Although it consists of only a small number of words, it is one of the few that have been specifically designed for the financial sector and the stock market domain and might therefore prove a lot more useful for this project. Since it has been manually built, it is also very small. Consisting of only 141 words as well as providing only binary classification in *positive* or *negative*, it is a good candidate to determine if the size is actually important for a good performance or if the advantage of the domain outweighs the size.

3. **Dr. Whissell (1989)'s "Dictionary of Affect in Language"** is not a domain specific lexicon but general English again, but it consists of over 8000 words. Every word has been rated in three categories: pleasure, activation and imagery on a scale from 1 (positive) to 3 (negative).

Bradley's and Whissell's lexicons provide weights for words just as SentiWordNet does. The difference is that they do not distinguish between several meanings a word might have depending on the context it appears in. They can be seen as a simpler version of SentiWordNet, on the one hand having the disadvantage of loosing detail but on the other hand gaining the advantage of not being dependent on other methods such as WSD.

6.4.1 Normalization of the Polarity Values

Since all three lexicons use different scales, it is necessary to normalize their polarity information if they are to be combined with each other or compared in any way. Choosing a range from 0 (negative) to 1 (positive), the normalized values represent a weighted indication as to what sentiment a word in a posting might have.

6.5 Implementation of a Lexical Classifier using SentiWordNet

After implementing a classifier that uses a traditional lexicon, exploring other options within the same class of methods seems a logical step. Some of the lexicons such as Whissell's already provide weighted values, but only one value per word. Depending on the context a word appears in though, it can have different meanings and express different opinions. Take for example the word "cancer". In the context of describing an animal, it does not carry much sentiment, whereas it carries decidedly negative connotations when talking about the illness cancer. Therefore the next step would be a classifier who does not only operate on lists of terms and phrases alone, but takes the context of words into account as well.

SentiWordNet as described in Section 4.3.4 is based on the WordNet thesaurus and assigns three sentiment scores to each synset of WordNet: positivity, negativity and objectivity. By providing these weighted polarity values for each sense of a word, SentiWordNet presents the ideal lexicon

for an improved and more complex classifier. Depending on the sense the values can vary very much. It is therefore not sufficient to just detect a sentiment word. In order to make use of SentiWordNets unique information, the correct sense has to be determined. This determination of a words sense depending on its context is called Word Sense Disambiguation (WSD).

6.5.1 Word Sense Disambiguation (WSD) using LingPipe

LingPipe[1] provides an online tutorial on WSD where they claim to reach around 62-68% accuracy for a number of classifiers. Mihalcea, Banea, and Wiebe (2007) provides the freely available corpus of the Semcor[2] project on her web page, which is basically the Brown Corpus[3] annotated with WordNet senses. Implementing WSD along the lines of LingPipes tutorial using the Semcor corpus turned out to be not practical for the following reasons: What happens in the tutorial is that actually one language model for every single sense synset is trained, which in turn leads to a huge number of models, each trained with a *very* small amount of samples, depending on the corpus used for training. Since there is no vast sense tagged corpus available, this results in severely undertrained models and does indeed present a problem. Secondly, there is no way to classify word senses that do not occur in the training corpus. Considering all these arguments, the idea of using LingPipe classifiers to do WSD has been discarded and other options have been explored instead.

6.5.2 WSD using Wilson Wongs NWD (Normalized Web Distance)

Wong, Liu, and Bennamoun (2008) presents an implementation of the so-called *Normalized Google Distance* (NGD) invented by Cilibrasi and Vitanyi (2007). They basically use Google queries with the two terms that should be compared to find out their distance. The distance is calculated by aggregating the page counts of the search terms. Details on this aggregation can be found in their paper. This method could be used for WSD easily by just querying for the current words synsets as compared to the text the word has been found in (its context).

6.5.2.1 Example

The word that should be disambiguated is the verb **"activate"**. The first step is to look at the text that this word has been found in.

> Do you know what it is , and where I can get one ? We suspect you had seen the Autospade , which is made by Wolf Tools . It is quite a hefty spade , with bicycle - type handlebars and a sprung lever at the rear , which you step on to **activate** it . Used correctly , you should n't have to bend your back during general digging , although it wo n't lift out the soil and put in a barrow if you need to move it ! If gardening tends to give you backache , remember to take plenty of rest periods during the day , and never try to lift more than you can easily cope with .

[1] LingPipe homepage: http://alias-i.com/lingpipe/
[2] Semcor project: http://www.cse.unt.edu/~rada/downloads.html#semcor
[3] Brown Corpus: http://www.sscnet.ucla.edu/issr/da/index/techinfo/M0911.HTM

Depending on how much context is used, the first term of the query could for example be "`step on to activate it`" (window of three surrounding words). The second term of the query is in turn each of the sense synsets of the word `activate` that have been found in WordNet:

1. "`to initiate action in; make active.`"
2. "`in chemistry, to make more reactive, as by heating.`"
3. "`to assign (a military unit) to active status.`"
4. "`in physics, to cause radioactive properties in (a substance).`"
5. "`to cause decomposition in (sewage) by aerating.`"

That means for the disambiguation of the verb "`activate`", five queries have to be processed. The first term of the query is the word plus n surrounding words as found in the text. The second term of the query is the synset of the word. It should be noted that the query that is set is not looking for pages that contain all of the words anywhere on the page, but for the whole connected phrase. The resulting probabilities are usually within a quite small interval.

This approach to WSD seems like a very primitive idea, but due to the huge corpus (which means everything Google has indexed) it yields reasonable though far from perfect results. Cilibrasi provides a free tool implementing the NGD, but unfortunately Google allows only so many queries in a short amount of time before the IP address gets banned. Therefore it is not practical to use NGD in this project as it means either stretching the queries over a long period of time or trying to get Google to increase the limit of allowed queries for a certain IP address. Wong, Liu, and Bennamoun (2008) have implemented their own version of the NGD that uses Yahoos web search instead of Google, as they are much more generous with their limit. Their approach has therefore been used as one of two tools to do WSD for this thesis.

6.5.2.2 Problems and Difficulties

One of the difficulties is to balance exactly how much context to use for the word to disambiguate. Obviously, using the whole posting as context is too much, even using the whole sentence the word occurs in would still be too much. In the above example, should the context be "`handlebars and a sprung lever at the rear , which you step on to activate it`" or rather only "`step on to activate it`"? Using the longer context, there is more information provided in order to determine which of the senses is occurring most often with it, but the drawback is: The longer the context, the less pages contain exactly that combination of words in exactly that order that make up the context. If the context gets so long that the number of pages containing it drops significantly, it will distort the calculated probabilities and result in incorrect tagging.

Another problem is caused by slang, abbreviations, spelling errors and other non-existent words that are not part of regular English. I.e. using a query such as "`the zim gov has been out of`

`options for some time`" where ´´zim" means *Zimbabwe* and "gov" *government* will most likely result in zero or very few pages found that contain exactly that phrase. Due to the calculations of the NWD, that causes the probabilities of all senses to be zero or equal, meaning that there is no way to decide which sense the word should be.

As a solution for this particular problem with probabilities equaling zero, the freely available Standard English dictionary JOrtho[4] has been extended to check for a words existence. It is used to look up every word of the context. A word gets added to the context only if it is found in the dictionary, therefore guaranteeing that the queries consist only of correct English words and thus preventing the system from getting zero-results. Although a query can theoretically still return zero if there really is no page containing the phrase, the likelihood of that happening is quite small due to the extremely large number of pages available on the Web. The biggest problem by far though is the fact that it takes up to one and a half hour to sense tag one single posting with eight sentences, depending on how long the sentences are and how many different meanings the words to disambiguate have.

6.5.3 WSD using Spreading Activation Networks (SANs)

The second tool that has been used is Tsatsaronis, Vazirgiannis, and Androutsopoulos (2007) Spreading Activation Network (SAN). They kindly provided a command line tool of their work. Unfortunately that means the tool has to be called as an external application, which is not an ideal solution and does contribute to an even bigger slowdown of the sense tagging. The SAN method is still faster than the NWD approach, but the accuracy of the tagging is only about 49%.

6.5.3.1 Problems and Difficulties

Additionally to the low accuracy, one other problem that arose with Tsatsaronis' tool is that it does not always give results. One possible reason might be if non-existent or slang words occur in the phrase to be disambiguated. Since SAN uses WordNet to build the network from, it might be that a word not present in WordNet might cause the network to fail and return nothing. To solve this problem, JOrtho has been used again as a dictionary to check for the occurrence of the words, as well as accessing WordNet via JWNL to establish certainty about their existence. This has reduced the number of times the tool returns nothing, but did not completely eliminated the problem.

Since the source code of the implementation is not available, it is hard to determine the reason for this behavior. It might be that in case the spreading does not result in a complete connected graph, it returns no senses for any of the words as a default. Another guess would be that it can only take a certain number of words at the same time, with too many words causing a failure.

[4] http://jortho.sourceforge.net/

6.5.4 Combining SANs and NWD

There is no way of knowing how accurate sense tagging with NWD is unless conducting a series of tests, which would require a manually sense tagged corpus to compare the results against. Though such a corpus has been made available by Rada Mihalcea for the Semcor project, time restrictions have made it impossible to thoroughly evaluate the NWD approach. A quick look at the results for a very small number of postings have shown that it does not give great results either. Still, from the manual checks as well as the fact that NWD always gives a result, the NWD seems to be a bit more reliable than SAN. Therefore NWD has been used as main sense tagger, whereas SAN does act as a tie breaker if NWD does not return one sense as the clear winner. The following paragraph briefly sketches how exactly the two methods have been combined for use in this project:

1. Use WSD to get probabilities for every sense of a word.
2. Check the distance between these probabilities.
3. If there is no clear winner (one sense stands out with a probability much higher than the rest):
    - All probabilities within a certain threshold (0.006 deviation from the highest ranking sense) are considered to be candidates for the best sense.
    - Perform SAN WSD.
    - Check if the winner of SAN WSD equals one of the NWD candidates.
    - If so, this one is taken as the best sense.
    - If not, the SAN candidate is ignored and the best of the NWD candidates is taken as the best sense.

Since the disambiguation takes a very long time with either of the two methods combined, the sense tagging cannot be done on the fly for every posting before classification. To avoid the problem of unacceptable waiting times and still be able to conduct meaningful tests and evaluation of the SWN classifier, the entire corpus, or rather the relevant part of it (postings with more than eight sentences), would need to be sense tagged in advance. Due to the fact that only one computer has been available during implementation, this turned out to be impossible. As a compromise, as many postings as possible in the given amount of time have been sense tagged. By the end of the project, 73 postings per category were available for evaluation of the classifier.

The problems of WSD severely affect the usability of this classifier, as they do not only bind the use of it to a certain sense tagged corpus, but also reduce the possibilities to evaluate any hybrid system it is part of. Whatever other classifiers a hybrid consists of, including the SWN classifier limits the amount of test samples for evaluation to the number of available sense tagged instances.

6.6 Experimental Setup and Results

Both implementations of the lexical classifiers presented in this chapter have been tested and evaluated. In the following, the test setup and results of the experiments are explained in detail and decisions for the selection of one of the classifiers for participation in the hybrid system are being discussed.

6.6.1 Evaluation of the Traditional Lexical Classifier

Experiments have been conducted on a balanced dataset of 528 posts per category, where each posting consists of eight or more sentences. The lexical classifier has been run once with every of the three available lexicons. As explained in Section 6.4, the classifier simply sums up the values of the sentiment words as specified in the lexicon, and then divides it by the number of sentiment words. Since the values of all lexicons have been normalized to the interval [0, 1] where 0 is negative and 1 is positive, the outcome of this process lies between 0 and 1 as well.

Since the corpus used in this project does not only contain samples for buy (positive) and sell (negative) but also hold (neutral), it is not sufficient to cut this range in half, where values bigger than 0,5 indicate a positive word and values smaller than 0,5 a negative word. It is necessary to define three intervals in order to correctly represent all three available categories. The most obvious approach - dividing the scope equally into three parts - leads to an interval $[0 - 0.33]$ for sell, $[0.33 - 0.66]$ for hold, and $[0.66 - 1]$ for buy. Despite this being the most intuitive solution, experimental results suggest that adapting these intervals to each lexicon individually as well as having varying interval lengths increase the accuracy. The best thresholds identified for each lexicon throughout testing are shown in Table 6.1. The final intervals have been identified by shifting thresholds and selecting the ones that (1) render the highest accuracy and (2) produce the best possible confusion matrix.

	Das	Bradley	Whissell
Interval for SELL sentiment	[0 - 0.55]	[0 - 0.56]	[0 - 0.413]
Interval for HOLD sentiment	[0.55 - 0.7]	[0.56 - 0.64]	[0.413 - 0.435]
Interval for BUY sentiment	[0.7 - 1]	[0.64 - 1]	[0.435 - 1]

Table 6.1: Intervals defined for the three categories buy, sell and hold.

As can be seen from the results shown in Table 6.2, Das' lexicon outperforms the other two by far. Though the smallest of the three lexicons, it shows that the adaption of a lexicon to a specific domain matters much more than its size.

The confusion matrices in Table 6.3 give valuable information about the distribution of the true and false positives/negatives. Das' lexicon seems to do quite well in distinguishing buy and sell as well as hold and sell, but has problems identifying the difference between buy and hold. Bradley's lexicon on the other hand does not excel in any of the categories but seems only to be able to identify slightly more buy and hold postings correctly than the average random distribution of 33% per category. Whissell's lexicon leans strongly towards positivity, meaning that a lot of postings

	Das	Bradley	Whissell
Total Count	1570	1571	1579
Total Correct	681	601	551
Total Accuracy	**0.434**	**0.383**	**0.349**
95% Confidence Interval	+/- 0.025	+/- 0.024	+/- 0.024
Macro-averaged Precision	0.441	0.387	0.351
Macro-averaged Recall	0.434	0.383	0.349

Table 6.2: Comparison of the three lexicons used for experiments.

have been falsely rated as buy. All these problems might be caused by a wrong choice of intervals. However, after adapting the intervals several times according to the indications of the reference-response matrices did not produce better results. The intervals as stated above are the final ones of this process of iteration and refitting.

	B	S	H
B	227	86	205
S	101	243	184
H	171	142	211

	B	S	H
B	216	111	192
S	159	170	199
H	190	119	215

	B	S	H
B	230	95	199
S	252	106	170
H	215	97	215

Table 6.3: Reference/Response Matrices of the three different lexicons. From left to right: Das, Bradley, Whissell. (B .. BUY, S .. SELL, H .. HOLD)

6.6.1.1 The Influence of Negation Tagging

In order to determine whether or not negation tagging has an influence on results, the lexical classifier has been evaluated with and without the use of this technique. In Table 6.4, a comparison of the tests with all three lexicons can be seen. While the domain-specific lexicon from Das seems to benefit significantly from the simple reversing of the polarities of words occurring in the context of a negation, Bradley's lexicon shows indifference to this approach. Whissell's lexicon even shows a notable decrease in accuracy.

Negation Tagging		Das	Bradley	Whissell
No	Total Accuracy	39,7%	38%	38,5%
Yes		43,4%	38,3%	34,9%
Difference		3,7%	-0,3%	-3,6%

Table 6.4: The influence of negation tagging on the three lexicons used for experiments.

6.6.1.2 Expanding Das' Lexicon

Although by far the smallest, Das' domain specific lexicon seems to outperform the other two. Having a domain specific corpus available, it might be worth to point out the possibility to grow this lexicon in future work. The expansion could be a simple adding of words (i.e. adjectives) in the corpus that occur in close proximity to the words already in the lexicon. Ideas for an

expansion can for example be found Qiu, Liu, Bu, and Chen (2009), who propose a propagation approach that exploits the relations between sentiment words and topics or product features that the sentiment words modify, and also sentiment words and product features themselves to extract new sentiment words.

6.6.2 Evaluation of the SWN Classifier

Due to the severe limitations of the lexical classifier using SentiWordNet and Word Sense Disambiguation (see Section 6.5.4), evaluations had to be conducted on the small number of sense tagged samples available at the end of the project. The dataset for testing consisted of 73 samples per category. The desired result of the classification of a posting using the lexical classifier with SWN is a probability for every of the three available categories buy, sell and hold. SentiWordNet provides a positive and negative value for each word. These values have been used to calculate the buy and sell probabilities by the following simple method:

- Sum up the positive values (and negative values, respectively) of every sense tagged word contained in the posting.
- Divide each of those sums by the number of sense tagged words.
- Normalize these positive and negative total values to sum up to 1.

Calculating Probabilities for hold, buy and sell

To derive the probability for category hold, the distance d between the probabilities for buy and sell has been used: If this distance converges towards 0, the probability for hold obviously goes towards 100%. If this distance converges towards a certain threshold (called *maxDistance* in the following, i.e. 0.5), the probability for hold goes towards 0%. To calculate it, a $\log_{10}$ function has been used:

$$prob(hold) = 1 - log_{10}(1 + x * d), \tag{6.1}$$

where d = distance between buy and sell score and $x = 9/maxDistance$. $\log_{10}$ of values under 1 result in negative values. Using probabilities normalized between 0 and 1, the log function would therefore yield negative values for the hold probability. Adding of 1 to the distance solves this problem. The division of the *maxDistance* by 9 is necessary as (1 + x*$maxDistance$) has to be 10. This is required to achieve a probability of 100% ($\log_{10}(10) = 1$) when $d = maxDistance$. It is possible to use a linear function as well, but experiments have shown that using a log function does improve classification quality.

After calculating the probability for hold, the two scores for buy and sell have to be recalculated in order to guarantee that the sum of all three probabilities is 1. Recalculation is done by normalizing by the following rule:

$$prob(x) = (1 - prob(hold)) * prob(x), where x \in \{buy, sell\} \tag{6.2}$$

Using WSD combined with SentiWordNet's values for positivity and negativity, the following results have been achieved:

Total Accuracy	**0.3973**
95% Confidence Interval	0.3973 +/- 0.0648
Macro-averaged Precision	0.4008
Macro-averaged Recall	0.3973

Table 6.5: Experimental results of the SWN classifier on a dataset of 73 samples per category.

	BUY	SELL	HOLD
BUY	36	24	13
SELL	27	31	15
HOLD	31	22	20

Table 6.6: Confusion matrix of the SWN classifier.

With an overall accuracy of 39-40% the classifier definetely does not live up to the expectations. As can be seen in the matrix in Table 6.6, it is not doing well in distinguishing between buy and sell, and performes even worse in terms of recognizing hold postings. As there is not enough data to do a thorough evaluation, there is not much point in interpreting these results. Until the underlying concepts such as Word Sense Disambiguation have not reached a satisfactory level of accuracy as well as an improvement in speed, this approach to a lexical classifier is not applicable in a hybrid system. Although the method certainly has potential, it is currently not able to perform any better than traditional lexical classifiers.

6.6.3 Conclusions

Taking into account the severe limitations the SWN classifier imposes onto the evaluation of a hybrid system, the traditional lexical classifier is clearly the choice for use in a combined approach. Although in theory SentiWordNet is able to provide a much more detailed and fine-grained assessment of a words polarity than traditional sentiment lexicons, in practice that advantage cannot compensate the low accuracy of current WSD methods. For all their limitations in expressiveness, the simpler lexicons do not depend on still premature methods like WSD. In view of the results discussed in Section 6.6.1, the obvious choice for the hybrid system is a lexical classifier with Das and Chen' lexicon and negation tagging.

6.7 Summary

Two versions of a lexical classifier have been implemented for the prototype. First and foremost, a new idea of a classifier using SentiWordNet has been tested. Unfortunately due to the necessity of word sense disambiguation (WSD) this method has a couple of severe drawbacks. Both WSD tools provided for use in this project perform with a low accuracy around 50% and are extremely slow. Despite the potential of this classifier it is no use until the underlying methods reach a sufficient

level of performance. The experimental results show that at the current state, it does not perform any better than traditional lexical classifiers which have been implemented as a baseline.

Using three different sentiment lexicon provided by Bradley and Lang, Das and Chen and Whissell, moderately better results have been achieved. Introducing further levels of complexity such as negation tagging seems to be helpful in increasing the accuracies. Due to these results, the classifier using SentiWordNet has to be dismissed and the traditional classifier with the highest accuracy is clearly the candidate for the hybrid system. Despite the the fact that lexicon-based methods are certainly the most popular and widely used approaches of all unsupervised methods (Pang and Lee, 2008), there are other unsupervised methods that have potential to be adapted to sentiment analysis.

An example of one such other unsupervised technique is using *generalized expectation criteria* (GEC) to label features instead of labeling whole texts and feeds that directly like that into a model. In their paper, Druck, Mann, and McCallum (2008) use their method to do topic classification of text samples and use specific words as indicators for the different topics (i.e. "puck" would be an indicator for the topic "hockey"). These indicative words are the aforementioned features that get labeled. Basically what they are doing is using lists of indicative words, so called constraints (which can be seen as an equivalent to a sentiment lexicon) as features describing a certain topic.

It could be worth trying to adapt this method to sentiment classification by using buy, sell and hold as topics and i.e. the words contained in SentiWordNet as features, with their positive /negative values indicating to which of the three topics a word belongs. Mallet[5] is a freely available Java library that supports classification with generalized expectations and could be used in future work to test and evaluate this idea.

[5]Mallet homepage: http://mallet.cs.umass.edu/

7 Supervised Methods

As there is labeled domain specific data available, an obvious option for classification is the use of supervised methods. Numerous works cover the machine learning perspective in the context of sentiment analysis, such as Mullen and Collier (2004) or Abbasi, Chen, and Salem (2008). Most problems in the area of opinion mining can be formulated in a way that classification, ranking or regression can be applied. Some common techniques include *decision trees* (DT), *naive bayes networks* (NB), *support vector machines* (SVM) and *maximum entropy models* (ME).

In the area of natural language processing (NLP), text mining and machine learning (ML), a lot of freely available libraries have been developed by the research community that make it easy to quickly implement and test such approaches. For this thesis, three different techniques have been chosen for implementation of the supervised classifiers: language model (LM), naive bayes (NB) and support vector machine (SVM). The libraries described in Section 4.3 have been used in the project.

7.1 Features and Feature Selection

Data-driven methods with regard to text processing call for a representation of the text that makes its most important and salient features available and processable. Usually this representation is a feature vector. The selection of features for machine learning techniques in general as well as for approaches tailored to a specific problem is covered in a great amount of works already (e.g. Simeon and Hilderman, 2008), (e.g. O'Keefe and Koprinska, 2009). Since this is an extensive topic beyond the scope of this work, only the features relevant for the task of polarity classification are described briefly. A key role for the success of a supervised statistical classifier and crucial for the training and classification process is the selection of appropriate features. Pang and Lee (2008) summarize the discussion about features and identify the following ones in their book:

1. **Term Presence vs. Term Frequency**. The phrase term frequency summarizes a number of metrics related to the number of occurrences of a term. In traditional information retrieval, measures such as the tf-idf weighting have been established as standards, however, term presence has been found to be a good indicator in the area of sentiment analysis. Term presence is simply a binary value stating whether a term occurs (1) in a given input entity or not (0). Pang and Lee (2008) state that such binary feature vectors form a more effective basis for review polarity classification than real-valued vectors representing term frequency. This might indicate a difference between the classification of factual texts and sentiment:

While a topic is more likely to be emphasized by frequent occurrences of certain keywords, sentiment may not usually be highlighted through repeated use of the same terms.

2. **N-Grams**. An n-gram is a subsequence of n items from a given sequence, where the items can be phonemes, syllables, letters, words or base pairs depending on the application. An n-gram of size 1 is referred to as a *unigram*, size 2 as a *bigram* and size 3 as a *trigram*. Anything beyond that is called *n-gram*. Consider the statement "`The respite in the European debt crisis is over.`": Unigrams in this case would be simply all single words of the sentence, whereas "`the respite`", "`respite in`", and so on would be the bigrams of the sentence. Whether n-grams with higher-order than unigrams are useful features seems to cause some disagreement. Pang et al. (2002) report that unigrams clearly outperform bigrams, Dave, Lawrence, and Pennock (2003) on the other hand find that bigrams and trigrams yield better accuracies.

3. **Parts of Speech**. Parts of speech (POS) are the basic types of words a language has. The exploitation of information regarding POS is quite common in the area of polarity classification, both in lexical classifiers as well as in the machine learning context. Certain POS have been considered to be more indicative of sentiment than others: adjectives and adverbs. Adjectives have been employed as features by a number of researchers (Hatzivassiloglou and Wiebe, 2000). A high correlation between the presence of adjectives and subjectivity as well as the inclusion of adverbs described by Benarama, Cesarano, and Reforgiato (2007) has led to an increased interest in the presence or polarity of adjectives and adverbs in classification. This holds especially in the unsupervised setting, but also supervised methods have been evaluated with a focus on certain parts of speech (Pang, Lee, and Vaithyanathan, 2002). However, this does not imply that other parts of speech do not contribute to expressions of opinion or sentiment. For example, Wiebe and Riloff (2005) specifically observe that the extraction of subjective nouns like "`anger`" might contribute to an improved classification.

4. **Syntax**. The inclusion of syntactic relations into the feature set requires a deeper linguistic analysis. Modeling valence shifters such as negation, intensifiers or diminishers (Kennedy and Inkpen, 2005) as well as collocations and more complex syntactic patterns have been found to be useful for sentiment analysis and subjectivity detection (Wiebe, Wilson, Bruce, Bell, and Martin, 2004).

5. **Negation**. The appropriate handling of negation can be of importance when it comes to classification using features. Consider the statement "`This cell phone has a good user interface`" versus "`This cell phone has no good user interface`": The feature representations of these sentences are likely to be considered very similar by most similarity measures, however, the simple negation "`no`" forces them into opposite categories. In Section 6.3.1, an overview of attempts to model negation has been given.

6. **Topic-Oriented Features**. While the system developed for this thesis does not explore these options, researchers like Pang and Lee (2008) further explore interactions between topic and sentiment. In their opinion these might be relevant for opinion mining. As an example they give two statements: "Wal-mart reports that profits rose" and "Target reports

that profits rose". Since Wal-mart and Target are two rivaling companies, these sentences can indicate completely different types of news regarding the subject of the document (i.e. if the document is a report regarding Wal-mart stocks).

7.2 Preliminary Data Processing

Based on the features discussed in the previous section and their selection, a number of preprocessing steps have been considered necessary for the implementation this project:

Sentence Boundary Detection. SBD is a necessary precondition for POS tagging. In order to guarantee a correct labeling of the input with parts of speech, the text has to be split up to single sentences first. Additionally, SBD allows to filter posts where no conclusive structure can be detected as well as very short texts. Details about the use of SBD and sentence filtering regarding the data used in this project can be found in Section 5.3.1.

Part-Of-Speech Tagging. POS tagging allows to filter for the parts of the text that potentially carry sentimental value. Especially adjectives and adverbs, but also nouns and verbs have been found to provide more information about a writers opinion than other parts of speech. In this project, both SBD and POS tagging is done by using available Markov models from LingPipe.

Negation Tagging. For the sake of simplicity, a very simple method of tagging negation has been chosen for this thesis. The method as proposed by Pang et al. (2002) just tags every word following a negation as negative until the end of the sentence is reached. Negations are detected by looking for occurrences of "not", "n't", and so on. While this detection ignores more subtle hints at negativity such as negative adverbs (`"I would never go to that concert"`) or negative pronouns (`"Nobody came to the concert"`), it is considered sufficient for testing the influence of the tagging on the accuracy of a classifier.

N-Gram Profile and Feature Vector Generation. For the training and evaluation of supervised classifiers, the construction of an n-gram profile is necessary. The profile and its feature vector, simply provides all features selected for the representation of a text as extracted from the labeled corpus. For instance, a profile can consist of all unigrams occuring in the data set. Its feature vector consists of the subset of unigrams selected as significant for representing a document. How the relevance of features is measured is of great importance to the success of the classifier. More on the topic of feature selection can be found in O'Keefe and Koprinska (2009). The two mechanisms for selection chosen for implementation in this project are further described in Section 7.5.1.

7.3 Language Models

Originally coming from the area of speech recognition, *statistical language models* (SLM or LM) become more and more important for a variety of other natural language applications: machine translation, document classification and routing, information retrieval, handwriting recognition,

spelling correction, and many more. In the area of sentiment analysis, too, researchers are are slowly starting to make use of SLMs. Hu, Lu, Chen, and Duan (2007b) estimate both the positive and negative language models from training collections and test by computing the *Kullback-Leibler divergence* between the language model estimated from test document and the two trained sentiment models. They assert the polarity of a test document by observing whether its language model is close to the trained positive or the negative model. Unigrams and bigrams are employed as the model parameters, and correspondingly maximum likelihood estimation and some smoothing techniques are used to estimate these parameters. They claim that experiments show an improvement of precision and robustness compared to a support vector machine.

Awadallah, Ramanath, and Weikum (2010) concentrate on detecting opinions in political discussions. They extract topic and sentiment related unigrams and bigrams as the basis for constructing a pro and a con query. Following standard LM techniques, they classify a document as positive or negative depending on which of the query likelihoods is higher. In contrast to training one LM, Jeong, Kim, Kim, Myaeng, and Oh (2009) build a total of 162 different models based on syntactic categories and combine them with a regression classifier. Their experiments, too, show that this approach outperforms a SVM classifier.

7.3.1 Language Models for Sentiment Analysis

The goal of SLM is to capture regularities of natural language for the purpose of improving estimating the probability distribution of various linguistic units, such as words, sentences, and whole documents. SLM employs statistical estimation techniques using language training data (i.e. texts). Because of the categorical nature of language, and the large vocabularies people naturally use, statistical techniques must estimate a large number of parameters, and consequently depend critically on the availability of large amounts of training data. With the availability of large amounts of text on the Web, it was possible to increase the quality of language models significantly. According to Rosenfeld (2000), who provides a good introduction into the fundamental concepts, the most successful SLM techniques use very little knowledge of what language really is. The most popular language models use n-grams and take no notice of the fact that what is being modeled is language. Exactly this can be exploited for sentiment analysis, where a knowledge-based approach quickly reaches its limits due to the complexities involved in expressing opinions.

7.3.2 Experimental Setup and Results

Experiments have been conducted by implementing and evaluating a LM classifier using LingPipe. Two versions of the classifier using input tokens have been created:

Process LM (PLM): This implementation is a dynamic conditional process language model which normalizes probabilities for a given length of input (characters). It represents a generative language model based on the chain rule which estimates the probability of characters given previous characters; the maximum likelihood estimator is smoothed by linear interpolation with the next lower-order context model:

$$P'(c_k|c_j,...,c_{k-1}) * P_{ML}(c_k|c_j,...,c_{k-1}) + (1 - \lambda(c_j,...,c_{k-1})) * P'(c_k|c_{j+1},...,c_{k-1}) \quad (7.1)$$

where P_{ML} are maximum likelihood estimates based on the term frequency:

$$P_{ML}(c_k|c_j,...,c_{k-1}) = count(c_j,...,c_{k-1},c_k)/extCount(c_j,...,c_{k-1}) \quad (7.2)$$

Count is the number of times a given string has occurred in the data, *extCount* is the number of times an extension to the string has occurred. *P(c_k)* is interpolated with the uniform distribution PU, with interpolation defined with the argument to lambda being the empty (i.e. zero length) sequence:

$$P(d) = \lambda() * P_{ML}(d) + (1 - \lambda()) * PU(d) \quad (7.3)$$

The uniform distribution PU only depends on the number of possible characters used in training and tests:

$$PU(c) = 1/alphabetSize \quad (7.4)$$

where *alphabetSize* is the maximum number of distinct characters in this model.

Token LM (TLM): This implementation is a dynamic sequence language model which models token sequences with an n-gram model, and whitespace and unknown tokens with their own sequence language models. The token n-gram model itself uses the same method of counting and smoothing as the PLM. Probabilities assigned to a character sequence are factored as follows:

$$P(cs) = P_{tok}(toks(cs)) \prod_{t-in-unknownToks(cs)} P_{unk}(t) \prod_{w-in-whitespaces(cs)} P_{whsp}(w) \quad (7.5)$$

where

- P_{tok} is the token model estimate and where *toks(cs)* replaces known tokens with their integer identifiers, unknown tokens with -1 and adds boundary symbols -2 front and back;
- P_{unk} is the unknown token sequence language model and unknownToks(cs) is the list of unknown tokens in the input; and
- P_{whsp} is the whitespace sequence language model and whitespaces(cs) is the list of whitespaces in the character sequence.

	Accuracy for D1 (%)	Accuracy for D2 (%)
PLM 5-gram	55,72	**58,40**
PLM 8-gram	57,66	57,39
PLM 11-gram	56,52	-
TLM 5-gram	49,40	**53,47**
TLM 8-gram	48,47	50,57
NB 5-gram	54,57	57,64
NB 8-gram	53,90	**57,64**

Table 7.1: Accuracy reached by various versions of LM and NB classifiers. Data set D1: 1251 posts, 5 sentences. Data set D2: 528 posts/category, 8 sentences

Tests on the two LM classifiers have been performed on datasets with 528/1251 input samples per category with 10-fold cross validation. The resulting accuracies listed in Table 7.1 are not the best ones achieved but averaged over the ten folds. While the PLM generally performs better than the TLM, results also indicate that an n-gram size of five as well as a bigger data set increases accuracy moderately. Experiments with different n-gram lengths suggest that accuracy increases with a size up to eight and then slowly decreases with higher orders. Generally, the classifiers trained on a dataset with a sentence filter of 8 clearly outperforms the ones trained on the bigger dataset with shorter postings. The best results are highlighted; the PLM classifier using 5-grams will be used for the hybrid system.

7.4 Naive Bayes

The *naive bayes* classifier (NB) technique is based on the Bayesian theorem (e.g. Zlotnick, 1970) with strong (naive) independence assumptions and is particularly suited when the dimensionality of the input entities is high like in sentiment analysis. NB greatly simplifies learning by assuming that features are independent of each other. Although independence is generally a poor assumption, in practice NB often competes well with more sophisticated classifiers, and occasionally outperforms them. Independence assumptions means that the classifier assumes that the presence of a particular feature of a class is unrelated to the presence of any other feature. Abstractly, the probability model for a classifier is a conditional model over a dependent class variable C with a small number of outcomes (categories like buy, sell, hold), conditional on several feature variables f_1 through f_n. Using Bayes' theorem and the independence assumption, the model (see Equation 7.6) is feasible even if the number of features n is large or when a feature can take on a large number of values.

$$P(C|f_1, ..., f_n) = \frac{1}{Z} P(C) \prod_{i=1}^{n} P(f_i|C) \tag{7.6}$$

where Z is a scaling factor dependent only on f_1, ..., f_n.

The NB classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or MAP decision rule. The corresponding classifier is the function *classify*:

$$classify(f_1, ..., f_n) = \max_c P(C = c) \prod_{i=1}^{n} P(F_i = f_i | C = c). \tag{7.7}$$

Information on the theoretical background of NB classifiers can be found in Ren et al. (2009); a thorough study of the data characteristics which affect the performance of NB is provided by Rish (2001). NB classifiers have a long history in text classification and have been explored for sentiment analysis early on. Researchers such as Pang and Lee (2004) employ NB in subjectivity detection, whereas Tan, Cheng, Wang, and Xu (2009) use it to create cross-domain classifiers. They propose an adapted, weighted transfer version of a naive bayes classifier to gain knowledge from the new domain data. In their paper, Melville, Gryc, and Lawrence (2009) present a unified framework for where they combine background lexical information in terms of word-class associations, and refine this information for specific domains using available training data and a NB classifier. In many papers dealing with sentiment analysis, NB classifiers are used as a baseline to compare other methods against (Paltoglou, Gobron, Skowron, Thelwall, and Thalmann, 2010) or generally as one of more supervised classifiers. While many (e.g. Pang, Lee, and Vaithyanathan, 2002)) report that NB does not perform as well as for example SVMs, results of this project with three categories and a different data set indicate that their performance is comparable.

7.4.1 Experimental Setup and Results

For this project, a NB classifier has been implemented using LingPipe. It is a trainable naive bayes text classifier with tokens as features. The token estimator is a unigram token language model with a uniform whitespace model and an optional n-gram character language model for smoothing unknown tokens. Naive bayes applied to tokenized text results in a so-called "bag of words" model where the words are assumed to be independent of one another:

$$P(tokens|cat) = \prod_{i<tokens.length} P(tokens[i]|cat) \tag{7.8}$$

Like the language models before, the NB classifier has been tested on two different sized data sets using 10-fold cross validation: one with 528 samples per category (sentence filter: eight sentences), one with 1521 samples (sentence filter: five sentences). Again, accuracies are averaged over all folds of the 10-fold cross validation. Results as shown in Table 7.1 indicate that both LM and NB classifier reach similar accuracies. Surprisingly, it does not seem to make any difference whether 5-grams or 8-grams are used for the NB classifier, so a classifier using 8-grams will be used for the hybrid system.

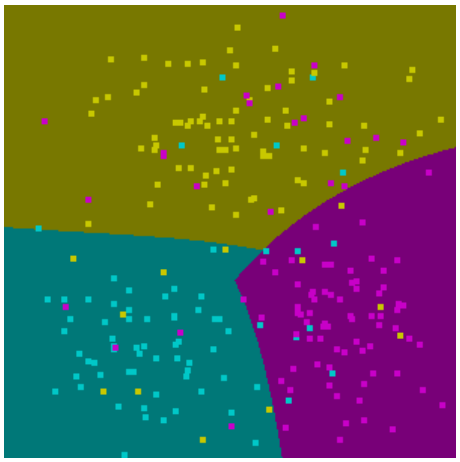

Figure 7.1: SVM hyperplanes: A two-dimensional example with three categories.Generated with LibSVM.

7.5 Support Vector Machines

Due to their general good performance, *support vector machines* (SVM) are still one of the most widely used methods for text classification. That is also true for the sub-discipline of sentiment analysis. While many researchers describe how to use a SVM in that area, its high accuracies as reported by O'Keefe and Koprinska (2009) or Abbasi, Chen, and Salem (2008) also make it ideal for use as a baseline to compare a new method against. The main concept of a SVM is the construction of a hyperplane or set of hyperplanes in a high or infinite dimensional space that optimally separate the data into n categories. Every data sample is represented by a set of features, which can be anything from numerical values to boolean expressions. In the context of sentiment analysis, a feature could for instance be how often the word "`great`" occurs in an input entity in relation to its occurrence in the whole corpus. High occurrence of a feature is interpreted as a good indicator for the class a training entity belongs to.

When confronted with an input entity without a label, the trained model is able to deduce a category based on the relation of the features to the support vectors obtained during training. The goal of training an SVM model is to find the optimal hyperplane that separates clusters of vectors in such a way that samples with one category (i.e. *buy*) of the target variable are on one side of the plane and samples with the other category (i.e. *sell*) are on the other size of the plane. The vectors near the hyperplane are called support vectors. Figure 7.1 illustrates the hyperplane process in a two-dimensional space. Details about the theory and concepts behind SVMs and possibilities for their application in text classification can for example be found in Tong and Koller (2002). Furthermore, Hsu, Chang, and Lin (2003) provide a practical guide to classification, including theory as well as parameter configuration, which can have a great influence on classification results, and examples.

7.5.1 Feature Selection

For this project, LibSVM has been used to implement a classifier (see Section 4.3.9. A number of feature combinations are evaluated: unigrams vs. bigrams, term frequency vs. term presence, all POS vs. adjective + adverb. For feature selection, the two variations described in the next sections are used.

7.5.1.1 Occurrence Counting

Occurrence counting (OC) is a very simple method suggested by Pang, Lee, and Vaithyanathan (2002). Any n-gram that occurs less often than a certain threshold in the whole corpus is discarded and not considered for the feature vector. In their paper, Pang et al. state that a threshold of 4 is sufficient to reduce the number of features to 16000 - 20000. For the evaluation of the SVM classifier with the London Forums Corpus, different thresholds are tested (see Table 7.3).

7.5.1.2 Categorical Proportional Difference

Categorical proportional difference (CPD) is a metric introduced by Simeon and Hilderman (2008), which measures the degree to which a word contributes to differentiating a particular category from other categories. The CPD for a word in a particular category in a text corpus is a ratio that considers the number of documents of a category in which the word occurs and the number of documents from other categories in which the word also occurs. In this project, it is used to find n-grams that occur mostly in one class of documents or the other, by using the buy, sell and hold document frequency of an n-gram. For example, if an n-gram occurs predominantly in buy documents, its CPD value will approach 1, whereas if it is uniformly distributed throughout all categories, its PD will decrease towards -0.33. The formula for determining the CPD value of a given n-gram for two classes is shown in Equation 7.9. When more than two classes are involved, the value for every single class is calculated and the maximum selected as illustrated in Equation 7.10.

$$CPD(w,\ c) = \frac{A - B}{A + B}, \tag{7.9}$$

where A is the number of times word w and category c occur together, and B is the number of times word w occurs without category c.

$$CPD(w) = max_i \left\{CPD(w, c_i)\right\} \tag{7.10}$$

Feature selection using CPD can decrease the number of features significantly. Using the whole London Forums corpus with 112.907 postings for feature extraction, enormous amounts of unigrams are produced. CPD can reduce them by 40.000, but it is obvious that this approach alone is not sufficient. There are two ways to reduce features further: (1) Take as input only the part of

the corpus that is used for training and evaluation (528 posts per category), or (2) Combine CPD with OC. While a corpus with 528 posts per category yields only 12.324 features for unigrams, CPD can reduce the amount to 9023 with a threshold of 1. However, the obvious high occurrence of unigrams with value 1 indicate that there are a lot of terms that occur very rarely. That means a meaningful reduction of features can only be done in combination with occurrence counting as described in the previous section. Table 7.2 shows the number of features selected with CPD for different thresholds. The integration of occurrence counting induces a significant drop by around 9000 in the feature count. Whereas OC alone requires a threshold of 100 to 500 to cut down features to 4000 to 1000, a combination with CPD lowers that threshold to 3 and still gets the same results.

Threshold	CPD: 528 posts	CPD: 112907 posts (whole corpus)	CPD + OC (min. 3x): 528 posts
0	12.324	158.438	3581
0,125	11.177	148.817	2987
0,25	10.715	145.719	2525
0,375	9909	137.429	1719
0,5	9691	132.242	1501
0,625	9247	124.426	1057
0,75	9104	121.123	914
0,875	9029	119.355	839
1	9023	119.184	833

Table 7.2: Number of selected features by CPD as feature selector as well as CPD + OC for various selection thresholds. The extracted features are unigrams.

7.5.2 Experimental Setup and Results

Various tests have been conducted in order to find the best setup for SVM, all of which use 10-fold cross validation. A *radial basis function* (RBF) kernel has been chosen for the implementation due to slightly better results than for example a linear or a polynomial kernel (see also Hsu, Chang, and Lin, 2003). Furthermore, a basic SVM is capable of distinguishing between two categories. Therefore, for a classification problem with n categories, there are in total n binary classifiers whose results are combined. The rules for combination can vary, but for this project, the following rule has been implemented:

1. For the k-th binary classifier, examples of category k are used as positive examples and all other examples are used as negative examples.

Various experiments have been conducted to test a range of different settings and features such as various OC and CPD thresholds, or unigrams versus bigrams. Accuracies with several different thresholds for OC alone have been determined as shown in Table 7.3. These tests have been performed with unigrams and bigrams with 528 posts per category used for cross-validation.

As the CPD value of an n-gram converges towards -0.33 when it is uniformly distributed throughout all three categories, a threshold of -1 means that no features are cut off by CPD. The evaluative results shown in Table 7.4 indicate that a threshold of zero combined with an OC cutoff at minimum three occurrences yields the most promising features for unigrams. Comparison to the results achieved OC alone (Table 7.3) makes it clear that feature selection using both CPD + feature presence as presented by Simeon and Hilderman (2008) and OC improves the accuracy for the corpus used in this project significantly. As with all other tests, the tests on CPD + OC thresholds have been conducted on the two versions of n-gram profiles described before. As expected, the profile with 528 posts performs better, as all the n-grams relate to the posts used for testing. When comparing unigrams to bigrams and using CPD + OC ((see Table 7.4 for unigram results, Table 7.5 for bigrams), it is clear that bigrams are the better choice for the London Forums corpus. The SVM classifier selected for the hybrid will therefore be trained on bigrams; feature selection will be based on CPD with a threshold of 0,125 and OC with a threshold of 3.

OC Threshold	Unigrams, 528 posts		Bigrams, 528 posts	
	Features	Accuracy	Features	Accuracy
min. 1x	32905	**45,49**	224463	43,91
min. 3x	11375	43,73	27530	**44,74**
min. 10x	4510	42,97	5322	42,97
min. 20x	2588	41,83	2114	39,67

Table 7.3: Accuracies for various OC thresholds on unigrams and bigrams. Two versions of n-gram profiles have been created: one using the whole corpus as source, the other using only the 528 posts per category used for training.

CPD Threshold	528 posts: CPD + OC (3x)		528 posts: CPD + OC (10x)	
	Features	Accuracy	Features	Accuracy
-1.0	11417	44,80	4518	42,59
0.0	7781	**55,90**	1753	**50,06**
0.125	5201	51,84	838	48,36
0.5	2526	-	235	-

CPD Threshold	Whole corpus: CPD + OC (3x)		Whole corpus: CPD + OC (10x)	
	Features	Accuracy	Features	Accuracy
-1.0	45524	45,75	19681	46,71
0.0	44122	**47,65**	19230	**50,19**
0.125	40325	47,28	18180	49,18
0.5	23750	40,18	0.58775	45,75

Table 7.4: Accuracies for various CPD and OC thresholds on unigrams. Two versions of n-gram profiles have been created: one using the whole corpus as source, the other using only the 528 posts per category used for training.

CPD Threshold	528 posts: CPD + OC (3x)		528 posts: CPD + OC (10x)	
	Features	Accuracy	Features	Accuracy
-1.0	27572	47,50	5241	43,88
0.0	20776	62,59	2103	54,85
0.125	14835	**70,32**	936	57,84
0.5	6018	**76,27**	128	-

Table 7.5: Accuracies for various CPD and OC thresholds on bigrams. Two versions of n-gram profiles have been created: one using the whole corpus as source, the other using only the 528 posts per category used for training.

CPD Threshold	528 posts: CPD + OC (3x)		528 posts: CPD + OC (3x)	
	Features	Accuracy	Features	Accuracy
-1.0	11631	51,55	1877	36,97
0.0	7938	62,08	1284	54,85
0.125	5472	**66,16**	852	-
0.5	2593	-	309	-

Table 7.6: The influence of simple negation tagging combined with various CPD thresholds on unigrams on a data set of 528 posts per category.

7.6 The Influence of Negation Tagging

If a tagging of negations is necessary for SVMs depends on what features are used. Generally, Pang, Lee, and Vaithyanathan (2002) claim that using bigrams is enough to capture the context and make negation tagging unnecessary. Therefore, negation has not been integrated into the evaluation of all bigram classifiers. However, if using unigrams it is recommended to include negations. Evaluations on the two best performing unigram classifiers (Table 7.6) as compared to two classifiers with the same settings but including negation tagging suggest that including negation decreases the performance tremendously. It could be that a finer modeling of negation would achieve better results, but at least the very simple method described Pang, Lee, and Vaithyanathan (2002) (tag every word following a negation as negative until the end of the sentence) is not helpful. Anyway, the bigrams outperform unigrams by far, so there is no need to investigate further into negation tagging for unigrams.

7.7 Summary

The results of all classifiers presented in this chapter clearly show the superiority of the machine learning approach compared to the knowledge-based techniques. While the lexical classifiers that have been evaluated barely make it past the 33% random choice limit, all three supervised methods selected for testing reach accuracies from around 50% up to 76% from the support vector machine. Although this is not nearly as high as results presented in the literature, the fact that there are three categories instead of two and the corpus quality is much lower than the ones used by other researchers makes these accuracies quite acceptable.

Especially the performance of the support vector machine stands out. Further experimenting with different kernel functions and settings might lead to an even better result. Additionally, a more thorough evaluation of feature selection could possibly increase the classifiers accuracy further.

Naive bayes and language model classifier perform with a similar accuracy. This comes as a surprise, as the expectations were that language models will do better. It seems that although naive bayes is based on very simple assumptions, it is still able to extract the most useful characteristics of each category and generalize well.

The following chapter discussed methods for the combination of the different classifiers selected in this and the last chapter and summarizes the results of the experiments conducted on the hybrid system.

8 Hybrid Systems

A number of research and development in the area of sentiment analysis concentrates on a single classifier for classification with resulting accuracies up to over 80% (see Section 3.4). What should be noted though is that many of the researchers limit the training and test data to a certain domain, a guaranteed grammatical correctness and 100% correctly labeled instances. Unlike the movie review corpus used by many researchers and which consists of only correctly labeled samples in correct English, classification of real life data from blogs or a forums cannot rely on these advantages (see Chapter 5). The heavy use of slang words, incorrect grammar, very short replies with no real content and repetitions like quoting of previous postings have a significant impact on the results of the classifiers. The results of the experiments conducted for this thesis show that a single classifier on it's own struggles to exceed the 70% threshold even despite the preprocessing steps taken to improve the corpus quality (i.e. discarding postings shorter than eight sentences).

Therefore the next step is to try to combine several different classifiers into a hybrid systems to investigate if that improves performance. After implementing, testing and evaluating a number of classifiers ranging from very simple to more sophisticated and complex supervised and unsupervised methods, the goal is to design a hybrid system consisting of the ones with the highest accuracies and the most diverse approaches.

8.1 Selection of Classifiers

Ideally, a hybrid system should consist of classifiers that have a wide range of variety, meaning that all the classifiers used in the system implement an approach that differs from the others. Methods with similar behavior might result in similar classification results (similar true/false positives and negatives), and a greater variety in approaches might improve the overall quality. For example, a joint system consisting of three lexical classifiers will not very likely result in any better accuracy than using just the one that performs best, as they all operate on the same aspect of natural language – the occurrence of certain indicative words and the syntactical structure of a text – and are therefore likely to produce very similar false positives and false negatives. Therefore when combining several classifiers, it is important to choose them in such a way that the coverage of correct classifications is maximal. That implies the classifiers in question should be as different as possible; their methods of classification should use very diverse aspects of the text.

One idea how to get the best coverage might be to do a visualization for the different classifiers on a fixed amount of test instances. Figure 8.1 briefly sketches this method: Every square represents a

Figure 8.1: Selection of classifiers for a hybrid system: One image for each classifier, to be overlaid with the images of other classifiers.

test sample, every true/false positive/negative will be colored accordingly. Applying this primitive methods on every classifier, an overlay could be done in order to get an idea of what areas different classifiers may cover and what aspects of natural language some classifiers handle better than others. Using this information it might be easier to find out which of the classifiers work best together. Due to the time restrictions of the project, this approach has not been implemented. The classifiers to be combined have been chosen by the different aspects of natural language they use for processing text without testing whether they really actually result in different true positives and true negatives.

8.2 Combination of Evidence

A lot of different methods to combine evidence from multiple sources exist. This section gives an overview over a few of them that can be of use in a hybrid system like this, where several classifiers with varying reliability contribute to a joint scored classification of an input sample.

8.2.1 Voting Scheme

The most intuitive way to combine the results of classifiers is a simple majority voting scheme (e.g. Das and Chen, 2007). Every classifier gets one vote as to which category the input in question belongs to. The category with the most votes wins and is the final classification for a test sample. For all the advantages of this easily understandable and intuitive approach, this method has several significant drawbacks.

Firstly, it completely ignores the certainty with which a classifier associates a category. For example, if a classifier thinks the input is category buy with a 41% certainty, category sell with a 39% certainty, and category hold with a 20% certainty, the voting scheme only takes into account that the classifier voted for *buy*, but not that it is almost as likely that it is *sell*. This complete ignorance of probabilistic evidence discards all additional knowledge returned by the classifiers implemented for this project, as they all return a certainty value for every category. Using a voting scheme where every classifier simply has one vote for one category equals setting the probability for this category to 1 and the probability for all other categories to 0.

The second and more inconvenient problem is that the number of classifiers in the system and the number of categories have to be chosen such that it is impossible to get a tie. Since it is not

possible to guarantee a 100% avoidance of ties for any number of categories greater than two, this method is limited to a binary approach. Thirdly, in order to guarantee an unambiguous decision for a binary scenario, the number of classifiers has to be uneven. In this project not only positive (buy)/negative (sell) has been taken into account as in other works in this area, but neutral (hold) as well. Hence, this method cannot be applied in this project unless ignoring the hold category and reducing the problem to a binary buy/sell scenario.

8.2.2 Mixing or Averaging (Weighted Sum)

As the name implies, results could be combined by summing them up according to their weights (i.e. used by Kennedy and Inkpen, 2005). The weight or importance that a classifier should have in the voting could for example be determined by the accuracy it achieved during the evaluation experiments conducted for every single classifier previously. Another approach to get weights assigned might be linear regression. The formula for the actual combination is simple:

$$m_{1\ldots n}(A) = \frac{1}{n}\sum_{i=1}^{n} w_i n_i \tag{8.1}$$

where m_i is the probability assigned to A by a source i, w_i the weight representing the reliability of a source, and n the number of sources. In a way, this method can be seen as a variation of the Dempster-Shafer rule. Sentz and Ferson (2002) point out that mixing is a generalization of averaging for probability distributions: "*If one applies the mixing operation to these inputs, the result will be a Dempster-Shafer structure all of whose masses are also at single points. These masses and points are such that the Dempster-Shafer structure is equivalent to the probability distribution that would have been obtained by mixing the probability distributions, that is, by simply averaging the probabilities for every point.*"

For this project, the weights of each classifier have been assigned according to their evaluation. Table 8.1 in Section 8.3 shows the distribution of weights as determined by the experiments.

8.2.2.1 Determining Weights with Linear Regression

Another method to retrieve weights, or rather, to adapt the initial weights to enable the method to yield the best results possible, could be *linear regression* (LR). LR is a method to determine the relationship of two variables X and Y, where X is independent (so-called predictor) and Y is dependent (predicted). X can be seen as input, producing Y as an output. The variables X and Y both consist of a number of concrete values x1, ..., xn and y1, ..., yn, respectively. An example could be to check if there is a causal relation between the characteristic "height" and the characteristic "weight" of a person (Does a taller person usually weigh more?). Based on available data, LR tries to fit a straight line describing that minimizes the error and thus allows for the best possible prediction of a value of Y based on a value of X.

$$Y = a + b * X \tag{8.2}$$

where X and Y are the variables, b is the slope of the regression line and a is the intercept point of the regression line with the y axis (the value of Y when X = 0). Equation 8.3 and 8.4 show the Slope and Intercept function, respectively.

$$b = (N * \sum XY - \sum X \sum Y)/(N * \sum X^2 - (\sum X)^2) \tag{8.3}$$

$$a = (\sum Y - \sum X)/N \tag{8.4}$$

where

- N = Number of values or elements.
- X = Value of the first variable X.
- Y = Value of the second variable Y.
- $\sum XY$ = Sum of the product of the values of the first and second variable.
- $\sum X$ = Sum of the values of the first variable.
- $\sum Y$ = Sum of the values of the second variable.
- $\sum X^2$ = Sum of square of the values of the first variable.

A detailed introduction to LR both in traditional and modern applications can be found in Myers (1994). In case of finding the optimum weights for the mixing method, X would be the weights, and Y could be seen as the accuracy of the hybrid classifier. Thus, LR can be used to adapt X in order to achieve the best possible Y. The major drawback of using LR to get the best weights as possible is that those weights are fitted to a certain corpus of data – the corpus that is used to find the regression line. Therefore the resulting weights will most likely only yield the best possible results for the classification of postings that were already used as the input corpus, or at least for data that is very similar to the data in this corpus. That means as soon as another data set is used as an input for classification, the weights will not necessarily fit the data and therefore might cause the hybrid system to produce bad results. Hence, if using a significantly different corpus, the weights might have to be recalculated accordingly. As the goal of this thesis is to design a classifier that can generalize on any data in the used domain, the idea to use LR has been discarded.

8.2.3 Combination of Evidence in Dempster-Shafer Theory

The *Dempster–Shafer theory* (DST) owes its name to work by Dempster (1968) and Shafer (1976) and is a theory of belief functions which are a generalization of the Bayesian theory. Whereas the

Bayesian theory requires probabilities for each question of interest, belief functions allow to base degrees of belief for one question on probabilities for a related question. It offers an alternative to traditional probabilistic theory for the mathematical representation of uncertainty and allows to combine evidence from different sources (i.e. classifiers) to arrive at a degree of belief that takes into account all the available evidence. The significant innovation of this framework is that it allows for the allocation of a probability mass to sets or intervals. DST does not require an assumption regarding the probability of the individual constituents of the set or interval, which makes it a potentially valuable tool for the combination of the results of different classifiers, where it is not possible to obtain a precise answer with all the classifiers agreeing on one category or a certain classifier even having an assumption about a certain category at all.

An important aspect and source of criticism is the modeling of conflict between evidence. The original conception of the theory by Dempster and Shafer strongly emphasizes the agreement between sources and does not take conflicting evidence into account at all, which can cause counterintuitive results when there is significant contradiction. However, there are several variations of the same general approach, adapted to better fit specific kinds of situations and aimed at handling conflicts in evidence better. A good overview over all these variations can be found in Sentz and Ferson (2002).

8.2.3.1 Formal Definition of the Original Dempster-Shafer Theory

A detailed description of the basics behind the DST as well as a definition of the terms *belief* and *plausibility* can be found in Dempster (1968). The most important part of the original conception of Dempster-Shafer theory is the rule how to combine evidence of various sources.

The measures of belief and plausibility are derived from the combined basic assignments. As stated in Sentz and Ferson (2002), the rule combines multiple belief functions through their basic probability assignments (weights m). Basic assignments in the case of combining classifiers for sentiment analysis are the probabilities that a classifier assigns to a certain category (buy, sell, hold). The belief functions are defined on the same frame of discernment, but are based on independent arguments or bodies of evidence.

The issue of independence is a critical factor when combining evidence and is an important research subject in the DST. The Dempster rule of combination is purely a conjunctive operation (AND). The combination rule results in a belief function based on conjunctive pooled evidence (Shafer, 1985). Specifically, the combination (called the joint m_{12}) is calculated from the aggregation of two basic probabilities m_1 and m_2 in the following manner:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - K}, forall A \neq \phi \tag{8.5}$$

where ϕ denotes that a source has no assumption about the category in question. For the actual calculation, whenever the assumptions of two sources are to be combined and one of them has no assumption about the category in question, the result of the combination is 0 (see Equation 8.6).

$$m_{12}(\phi) = 0 \tag{8.6}$$

$$K = \sum_{B \cap C = \phi} m_1(B) m_2(C) \tag{8.7}$$

The factor K represents the basic probability mass that is associated with conflict. This is determined by the summing the products of the basic probabilities of all sets where the intersection is null. Using 1-K as a normalization factor causes complete ignorance of conflict by attributing any probability mass associated with conflict to the null set, which is the main point of criticism in Yager et al. (1994). Consequently, this operation yields counterintuitive results when confronted with significant conflict.

Example

This example with two sources has been taken from Sentz and Ferson (2002) and modified slightly to show that significant conflict leads to very bad results.

	Classifier 1	Classifier 2
Weight for category:	m_1(buy) = 0.99	m_2(buy) = ϕ
Weight for category:	m_1(sell) = 0.01	m_2(sell) = 0.01
Weight for category:	m_1(hold) = ϕ	m_2(hold) = 0.99

A ... buy, B ... sell, C ... hold

		m_1(A)	m_1(B)	m_1(C)
		0.99	0.01	ϕ
m_2(A)	ϕ	m_1(A) m_2(A) = 0	m_1(B) m_2(A) = 0	m_1(C) m_2(A) = 0
m_2(B)	0.01	m_1(A) m_2(B) = 0.0099	m_1(B) m_2(B) = 0.0001	m1(C) m_2(B) = 0
m_2(C)	0.99	m_1(A) m_2(C) = 0.9801	m_1(B) m_2(C) = 0.0099	m_1(C) m_2(C) = 0

Using Equations 8.5 – 8.7:

1. Calculation of the combined basic probability assignment for a particular cell: multiplication of the masses from the associated column and row.

2. Where the intersection is nonempty (both classifiers have an assumption about a certain category), the masses for a particular set from each source are multiplied, e.g., m_{12}(B) = (0.01)(0.01) = 0.0001

3. Where the intersection is empty (the two classifiers do not both provide a value for a category), this represents conflicting evidence and should be calculated as well. For the empty

intersection of the two sets A and C associate with classifier 1 and 2, respectively, there is a mass associated with it. m_1(A) m_2(C)=(0.99)(0.99) =(0.9801).

4. Summarization of the masses for all sets and the conflict.

5. The only nonzero value is for the combination of B, m_{12}(B) = 0.0001. In this example there is only one intersection that yields B, but in a more complicated example it is possible to find more intersections to yield B.

6. For K, there are three cells that contribute to conflict represented by empty intersections. Using equation 8.7, K = m_1(A) m_2(B): (0.99)(0.01) + m_1(B) m_2(C): (0.99)(0.01) + m_1(A) m_2(C): (0.99)(0.99) = 0.9999

7. Using Equation 8.5, calculate the joint, m_1(B) m_2(B) = (0.01)(0.01) / [1-0.9999] = 1

Due to the fact that there is highly conflicting evidence, the conflict is almost 1, which corresponds to a probability (B) = 1. This is the result of normalizing the masses to exclude those associated with conflict. This points to the inconsistency when Dempster's rule is used in the circumstances of significant relevant conflict that was pointed out by Zadeh (1986).

8.2.3.2 A Variation of the Dempster-Shafer Rule: Discount & Combine

Numerous other rules of combination have been developed in order to deal with the particular problem illustrated by example 8.2.3.1 or are tailored to very specific scenarios. A good overview is given in Sentz and Ferson (2002). One of the rules that is particularly applicable for this project is called *discount & combine*, since this rule extends the original rule additionally taking the level of trust in a classifier into account (weights). Introduced by Shafer (1976), this particular rule deals with conflicting evidence just how the name declares. In this scenario, all of the sources are discounted, meaning they are assigned a level of trust. After including this discount, the resulting functions are combined with either Dempster's rule or any other rule as preferred. In this project, the original rule is being used after discounting. The so-called *discounting function* represents something like an expert who is qualified to distinguish between sources of information and determine their reliability. Shafer applies this function to each *belief* (see Dempster, 1968) as defined in Equation 8.8.

$$Bel^{\alpha_i}(A) = (1 - \alpha_i)Bel(A) \tag{8.8}$$

where $Bel(A)$ is the belief in category A, $1 - \alpha_i$ is the discounted degree of reliability of a source ($0 \leq \alpha_i \leq 1$) and i is the index to specify the particular discounting function associated with a particular belief measure. The average of all belief functions associated with category A is then used to obtain the combined belief of all sources in category A (see Equation 8.9).

$$\overline{Bel}(A) = \frac{1}{n}(Bel_1^{\alpha}(A) + ... + Bel_n^{\alpha}(A)) \tag{8.9}$$

Consequently, the discount and combine method uses an averaging function as the method of combination. This can be used to eliminate the influence of any strongly conflicting single belief function as is highly likely with the classifiers used in this project. The fact that it is possible to assign a level of trust to every single source makes this rule interesting for experiments in this thesis.

8.3 Experimental Setup and Results

For the hybrid classifier, all classifiers with the best accuracies from the different approaches have been selected as identified in the previous experiments. Specifically, these are:

1. Lexical classifier using Das lexicon (see Section 6.6)
2. Naive Bayes classifier trained on 8-grams (see Section 7.4.1)
3. Language model classifier trained on 8-grams, character based (see Section 7.3.2)
4. SVM classifier trained on bigrams (no negation) (see Section 7.5.2)

The methods chosen for combination are mixing, Dempster-Shafer theory with the original rule and with a modification, the discount & combine rule as described in the previous sections. For the weights needed for mixing, the distribution determined by the evaluation of the single classifiers are used (see Table 8.1). The classifiers used for the hybrid system are highlighted. The original Dempster-Shafer rule, of course, is excluded from all experiments with varying accuracies, as it does not use any weighting.

Classifier	Weight set A1 (%)	Weight set A2 (%)	Weight set A3 (%)
Lexical Classifier (Das)	**42,93%**	**40%**	**20%**
Lexical Classifier (Bradley)	38,52%	-	-
Lexical Classifier (Whissell)	32,39%	-	-
Lexical Classifier (SentiWordNet)	39,73%	-	-
Naive Bayes Classifier	**54,72%**	**55%**	**20%**
DLM Classifier (char. based)	**59,12%**	**55%**	**20%**
DLM Classifier (token based)	52,20%	-	-
SVM Classifier	**68,15%**	**60%**	**20%**

Table 8.1: Weight distributions (accuracies) of all single classifiers.

Since these weights are heavily imbalanced and indicate an almost insurmountable domination of the SVM classifier, several tests with modified weights have been conducted. The second column of Table 8.1 shows a more moderate distribution, the third completely balanced weights. Both distribution A2 and A3 have been manually selected in order to investigate what impact such a single strong and dominating source can have and if it is better to moderate its influence. Secondly, as with all experiments on single classifiers before, two test series have been conducted: One on a dataset of 1251 samples with at least 6 sentences in each of the postings, and one on 528 samples and at least 8 sentences. The resulting accuracies of the hybrid classifier using different methods

Method of Combination	Accuracy A1 (%)	Accuracy A2 (%)	Accuracy A3 (%)
Mixing	56,88	56,88	**57,24**
DST (Original)	57,47	-	-
DST (Discount & Combine)	**58,82**	58,82	55,66

Table 8.2: Accuracy of the hybrid classifier for different methods for the combination of evidence on data set D1 (1251 samples, 6 sentences).

Method of Combination	Accuracy A1 (%)	Accuracy A2 (%)	Accuracy A3 (%)
Mixing	66,88	**68,15**	67,52
DST (Original)	67,52	-	-
DST (Discount & Combine)	67,52	66,88	**68,15**

Table 8.3: Accuracy of the hybrid classifier for different methods for the combination of evidence on data set D2 (528 samples, 8 sentences).

for the combination of evidence on the first dataset are listed in Table 8.2, accuracies for the second dataset in Table 8.3.

As expected, data set D2 yields much better results, due to the fact that also all single classifiers do a lot better with that set. As can be seen, choosing a more balanced distribution such as A2 or A3 does not have a great impact. In fact, for mixing, a slightly more balanced weight distribution (A2) seems to be the best choice, whereas for discount & combine, a completely flat distribution (A3) yields the best result. Both methods, however, do not give conclusive evidence in preference of a certain weight distribution. The differences of the resulting accuracies are simply too small to indicate any significance concerning certain weights.

Classifier	Accuracy for D1 (%)	Accuracy for D2 (%)
Lexical Classifier (Das)	42,93%	-
Lexical Classifier (Bradley)	38,52%	-
Lexical Classifier (Whissell)	32,39%	-
Lexical Classifier (SentiWordNet)	39,73%	-
Naive Bayes Classifier	53,90%	57,64%
DLM Classifier (char. based)	55,72%	58,40%
DLM Classifier (token based)	49,40%	53,47%
SVM Classifier	-	70,32%
Hybrid Classifier Mixing	57,24%	68,15%
Hybrid Classifier Original	57,47%	67,52%
Hybrid Classifier Discount & Combine	58,82%	68,15%

Table 8.4: Overview of all classifier accuracies.

Overall, the results of the experiments with the hybrid system are not encouraging. Table 8.4 provides an general overview of the results of all experiments done in this project. Comparing the accuracies of all single classifiers, as well as the best of them versus the hybrid classifier, it is obvious that none of the combination efforts enable the hybrid to exceed the accuracy of its best member, the SVM classifier. One of the problems might be that the selected classifiers each produce similar true/false positives/negatives although their approach is very different. A more thorough analysis of the coverage of each of them as discussed briefly in Section 8.1 might give an insight whether a different selection could possibly yield better results. Even though the discount

& combine rule of DST actually takes the degree of belief of each classifier into account, the accuracy does not notably vary from the original rule or the mixing method. A reason for that could be that the probabilities for the categories produced by any of the classifiers is within a very small interval. For example, the SVM classifier gives the following probabilities for a test sample:

Rank	Category	Score (%)
1	SELL	33,76
2	BUY	33,22
3	HOLD	33,17

Table 8.5: Probabilities for each category given by the SVM classifier for a test sample.

As can be seen, the classifier gives the probabilities for the categories within a small range. That way, the belief in winning category is not significantly higher than for any of the other categories. Looking at the example of the SVM given in Table 8.5, the probability for the winning category *sell* is only 0,5% higher than the probability for *buy*, and 0,7% higher than for *hold*. Thus, with such minimal differences, the DST is not able to assign much mass to a category and fails to achieve the expected results. For future work, these problems can be a first step for improvements and a new set of evaluations. For example, the focus could go towards finding a way to emphasize the probability of the winning category, or towards analyzing the coverage of true/false positives/negatives of each classifier in order to find classifier techniques that are better suited for being combined.

Another idea might be design a completely different hybrid system. It could be a possibility to take the best classifier, the SVM, and define some threshold of certainty. If the threshold is not reached, then the classifier is fairly sure which category is the winning one. If the threshold is reached, it means the classifier is not able to decide on its own. Other classifiers could then act as tie-breakers, where each of their votes adds a factor to the base certainty. When taking the example probabilities shown in Table 8.5, the difference between the classes are quite distinct. Taking a threshold value of 0.05, in this case, no tie-breaker might be needed as the difference is 0.07. If however the value for the winning category would be lower, the language model or naive bayes classifier could be used to cast their votes.

8.4 Summary

The theory that a hybrid system combining more classifiers in order to compensate the disadvantages of the single classifiers has not been proved. In fact, the combination yielded worse results than the best single classifier, even if the results were not bad. However, this version of the system was just a first attempt at combination. Solving some of the problems mentioned in the last section as well as implementing some more ideas described in this chapter might be a step towards an improvement. During research for possibilities for a more sophisticated way of combining evidence, the Dempster-Shafer theory seemed like the best option since it provided exactly what was needed - a way to model uncertainty as well as the level of trust in one of the sources, and a

way to deal with conflicting evidence. The most interesting part about it is that there is a great number of variety. Based on the original rule, numerous adaptations have been created to deal with very specific problems, one of which exactly fitted the problem presented by this thesis: The discount & combine rule. Unfortunately the fact that all classifiers operate within a very small interval for the probabilities and do not really emphasize the winning category, has proven to be a bigger problem than anticipated. It might be worth looking at other variations of the DST rules (i.e. Yager's rule) that could be modeled to better fit the exact problem of the small variations of certainty about the winning category as compared to the other categories. This could prove useful in finding a solution other than trying to find an algorithm to emphasize the winning category. Also an option is to look for an entirely different method for combination of probabilities.

Last, there is always the possibility to completely rethink the strategy of combining all classifiers as more or less equal and try the opposite approach - using one single classifier as basic trusted source, to which all other less trustworthy classifiers only contribute in ambiguous situations where the base classifier is not able to make a clear decision. It came as a surprise that the support vector machine classifier was able to perform so outstandingly well with that bad quality data, since the expectations were the same as for the language model and naive bayes classifier. Therefore, expecting only mediocre accuracies around 50-55% from all classifiers, it was never an option to have this kind of setup where one classifier is the sole source with others just supporting in unclear situations. However, this scenario might be worth investing some time.

Overall, the performance of the hybrid classifier is - although not as good as expected - still better than all other classifiers except the SVM. Nevertheless the theses that a combination of single parts to a more complex system will be able to outperform the single classifiers has not worked out due to the seemingly unimportant problems mentioned above. For future work it is clear that they must be dealt with in order to get satisfactory results.

9 Lessons Learned

During the development and implementation of the system and the experiments with various setting, several points that seemed quite easy to overcome or that did not even seem like any problem at all turned out to be quite hard to deal with. That, in the end, helped a lot for relativizing and reassesing the related work and theoretical research done on the topic of sentiment analysis before and during the practical part of the thesis. These are some of the conclusions and valuable insights gained during the work on this project:

- Sentiment analysis is still a long way from being applicable in real world market research.
- The great amount of subtlety involved in expressing opinions in natural language is hard to overcome.
- The machine learning approach might be the more promising way to go when it comes to sentiment analysis. Feature selection might provide the potential for improvement.
- Integrating unlabeled data into the training is necessary to overcome the restriction to available human-labeled corpora and allow research to venture beyond the few domains for which a corpus is available.
- Forum posts in the stock market domain often include lengthy citations of news articles. While such posts usually provide a lot of text, they also bias the results with their factual content. It might be inevitable for future improvement to filter the data for subjective content.
- As is the nature of postings in forums, the corpus contains a great number of very short postings that are labeled but do not provide enough context for training or classification.
- When using traditional lexical classifiers, having a lexicon adapted to the domain matters much more than its size.
- Word sense disambiguation is not yet sophisticated enough to be a reliable source of information. Until such concepts have not reached a satisfactory level of accuracy as well as an improvement in speed, a classifier making use of SentiWordNet is not applicable even though in theory it has a lot of potential.
- Despite the the fact that lexicon-based methods are certainly the most popular and widely used approaches of all unsupervised methods, there are other unsupervised methods that have potential to be adapted to sentiment analysis and could be well worth trying.

- It might be worth to invest more time into feature selection for the support vector machine.
- Although the Dempster-Shafer theory seems perfectly suitable for the combination of single classifiers, the results indicate that the certainty interval each classifier has for the categories in question must be more diverse for it to work.
- The design of the hybrid system will probably need to be overthrown completely and turned into a system where the strongest classifier is only supported by the other classifiers when the probabilities are inconclusive and ambiguous.

10 Conclusions and Future Work

In the course of this thesis, a prototype for sentiment analysis of content from social media platforms in the area of the stock market has been developed. After a thorough analysis and comparison of works in the research area, several gaps have been identified and used to define the basic requirements that a new approach has to fulfill in order to overcome them. These gaps range from the data taken for training and testing to the methods used for classification and the combination of single classifiers to a complex hybrid system. Additionally, in order to reflect the nature of stock market trading more closely, three categories instead of two have been selected for classification: buy, sell and hold. While the traditional lexical approach to sentiment analysis usually focuses entirely on the indicator words as defined in the lexicon, a novel method taking into account the meaning of such a word has been developed. This required venturing into the field of word sense disambiguation in order to determine which meaning is the most likely one depending on the context surrounding the term in question. Two different techniques have been employed for this task - one involving the Google Web distance, where the number of results to a Google query is used to calculate a likelihood for a certain word sense from WordNet, the other one using a spreading activation network built from WordNet, where the distance between synonym sets are used for determining the correct meaning. Additionally to this new approach, several kinds of existing lexica, ranging from a small domain specific to a extensive general English lexicon have been compared and evaluated for their applicability in the stock market domain.

Apart from the knowledge based part of the implementation, several supervised learning methods have been tested for their potential. While the Naive Bayes network, which is often used for classification in the area of sentiment analysis, performed according to the expectations, the support vector machine showed a much greater promise. After mediocre initial results, a thorough investigation of the topic of feature selection improved the accuracy tremendously. A technique usually used in the field of speech recognition, token and character based language models, have been adapted for use in this prototype. The classifier is comparable to the Naive Bayes network in its results. The single classifiers have been combined to a hybrid system and tested with several different approaches to the combination of probabilistic evidence. While the theory of these comination methods differ significantly, the variation in the outcome was not as high as expected.

On the technical side of this thesis, there are numerous opportunities for future work and improvements. Starting with the data available for the research, a possibility to venture beyond the limitations of the few available labeled corpora is to explore the use of unlabeled data. Since it is not always possible to extract rated content for a certain domain from the Web, it is inevitable that a lot of researchers fall back to the few domains where labeled corpora already exist, thus

limiting the research in sentiment analysis in general. Especially for the financial domain, next to no corpora exist. By making use of unlabeled data (i.e. the data from the ADVFN forum as described in Section 5.3.2), new techniques for sentiment analysis could be explored, or at the very least, the amount of labeled data could be increased, thus leading to better quality data and increased accuracies of supervised classifiers.

Even though it would be a possibility to create a better sentiment lexicon which is tailored to the domain and contains a bit more terms than Das' lexicon, all experiments indicate that the whole lexicon based approach is not very suitable for the task. Thus, it seems more valuable to concentrate on improving the machine learning methods. One option to enhance the performance of the strongest single classifier, the support vector machine, could be a further investigation of feature selection. To date, only a limited amount of research has been done in the area of feature selection for sentiment analysis. Experiments in this project, however, clearly show that even with only a rudimentary approach to feature selection, a great increase in accuracy can be achieved.

The experiments with the hybrid classifier give rise to a number of theses that could be put to a test in future work. One of them is that the very small probability intervals of the single classifiers cause the Dempster-Shafer combination (DST) to perform worse than expected due to the fact that none of them emphasize the winning category clearly enough. Therefore it might be worth looking at other variations of the DST rules that could be modeled to better fit the problem. Other possibilities include finding an algorithm to emphasize the winning category or looking for an entirely different method for combination of probabilities.

Also on the conceptual side of the project, various ways of expanding the original scope come to mind. Throughout all the experiments conducted for this thesis, sentiment analysis has been performed on document level (single postings) only. While that is an important step towards judging the overall opinion regarding a certain stock, it alone is not sufficient for a thorough analysis of stock market sentiment. It is crucial to combine these single elements and work on an opinion overview over the shareholder market. Merging the analysis of postings into a sentiment summary of a certain stock, or even a sector (e.g. technological stocks). An intuitive approach to such a combination could be a plot of opinions for a certain stock over time. Although that is a very simple idea, this can provide insight into trends of the assessment of the stock or company by the shareholders and potential buyers. Comparing these evaluation to the market rate might be a first step towards an estimation as to if and how much public opinion correlates with the fluctuations of the market price or whether it is able to influence the trading of a certain stock to some extent.

Comparisons of the performance in terms of public opinion of competing companies, measurements of the impact of financial news released by the company or market analysts are just a few of possible future tasks that might offer great benefits to the financial community. Another task could be to analyze social structures of Online communities in the financial sector by observing the reactions of people in the presence of one very dominant person releasing strongly opinionated postings. To what extent does a person like this influence the opinions of others, are there changes in the posting behavior as a response to such an authority? Can this change the public opinion in such a significant way that a change in opinion regarding a certain company can be induced?

All these possible application scenarios are of great commercial interest. Improvement in quality control, word-of-mouth-marketing, quicker response to market changes and analysis of trends over time are just a few of the possibilities opening up with automated sentiment analysis. Both companies and private persons could benefit greatly from the application of opinion mining, be it in the area of marketing or customer service. Information is becoming a more and more precious commodity, and techniques to single out the relevant parts, analyze and interpret them are in high demand. Sentiment analysis as it is now is certainly not even close to being able to replace manual work, but it can act as a supplement. However, opinion mining stays a highly active field of research and although it might take a while, some time in the future it might reach a status where it can function on its own and yield reliable results.

Bibliography

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Transactions On Information Systems*, *26*: 12:1–12:34. ISSN 1046-8188.

Amazon Mechanical Turk (2012). A crowdsourcing Internet marketplace for work. Retrieved from https://www.mturk.com/mturk/.

Andreevskaia, A. and Bergler, S. (2006). Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *In Proceedings from EACL '06: 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Antweiler, W. and Frank, M.Z. (2004). Is All That Talk Just Noise ? The Information Content of Internet Stock Message Boards. *Journal of Finance*, 1259–1294.

Awadallah, R., Ramanath, M., and Weikum, G. (2010). Language-model-based Pro/Con Classification of Political Text. In *Proceeding from ACM SIGIR '10: The 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, 747–748. ACM, New York, NY, USA. ISBN 978-1-4503-0153-4.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley, 1st edition. ISBN 020139829X.

Bautin, M., Vijayarenu, L., and Skiena, S. (2008). International Sentiment Analysis for News and Blogs. In *Proceedings from ICWSM '08: The International Conference on Weblogs and Social Media*.

Beineke, P., Hastie, T., Manning, C., and Vaithyanathan, S. (2004). Exploring sentiment summarization. In *Proceedings of the AAAI '04: Spring Symposium onExploring Attitude and Affect in Text*.

Benarama, F., Cesarano, C., and Reforgiato, D. (2007). Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone. In *Proceedings from ICWSM '07: International Conference on Weblogs and Social Media*. Boulder, Colorado, USA.

Boiy, E., Hens, P., Deschacht, K., and Moens, M.F. (2007). Automatic Sentiment Analysis in On-line Text. In *Proceedings from ELPUB '07: Conference on Electronic Publishing*. Vienna, Austria.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*: 1–8.

Boyd-Graber, J. and Resnik, P. (2010). Holistic Sentiment Analysis Across Languages: Multilingual Supervised Latent Dirichlet Allocation. In *Proceedings from EMNLP '10: Conference on Empirical Methods in Natural Language Processing*, 45–55.

Bradley, M.M. and Lang, P.J. (1999). Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings. Technical report, Center for Research in Psychophysiology, University of Florida, Gainesville, Florida.

Butcher, D. (2009). Mobile Social Networking Blowing Up. Research Survey. Forrester Research. Retrieved from http://www.mobilemarketer.com/cms/news/research/4458.print.

Carbonell, J. (1979). *Subjective Understanding: Computer Models of Belief Systems.* Phd. dissertation, Yale University, Conneticut.

Cilibrasi, R.L. and Vitanyi, P.M.B. (2007). The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, *19* (3): 370–383.

Crocker, M.W. (1996). *Computational Psycholinguistics: An Interdisciplinary Approach to the Study of Language.* Kluwer Academic, first edition.

Cutler, D.M. and Poterba, J.M. (1989). What moves stock prices. *Journal of Portfolio Management*, *79*: 223–260.

Das, S. and Chen, M. (2007). Yahoo! For Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, *53* (9): 1375–1388.

Daumè, H. and Marcu, D. (2006). Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, *26*: 101–126.

Dave, K., Lawrence, S., and Pennock, D.M. (2003). Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proceedings from WWW '03: The 12th International Conference on World Wide Web*, 519–528.

Dempster, A.P. (1968). A Generalization of Bayesian Inference. *Journal of the Royal Statistical Society*, *30*: 205–247.

Denecke, K. (2008). Using SentiWordNet for Multilingual Sentiment Analysis. In *ICDE Workshops*, 507–512.

Devitt, A. and Ahmad, K. (2007). Sentiment Polarity Identification in Financial News: A Cohesion-based Approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.* Prague, Czech Republic.

Ding, X. and Liu, B. (2007). The Utility of Linguistic Rules in Opinion Mining. In *Proceedings from ACM SIGIR '07: Annual ACM Conference on Research and Development in Information Retrieval.* Amsterdam, Netherlands.

Druck, G., Mann, G., and McCallum, A. (2008). Learning from Labeled Features using Generalized Expectation Criteria. In *Proceedings from SIGIR '08: ACM Special Interest Group on Information Retrieval.* Singapore.

Eckstein, P.P. (2010). *Statistik für Wirtschaftswissenschaftler.* Gabler, second edition.

Engle, R.F. and Victor, K. (1993). Measuring and Testing the Impact of News on Volatility. *Journal of Finance*, *48*: 1749–1778.

Esuli, A., Baccianella, S., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings from LREC '10: The 7th conference on International Language Resources and Evaluation.* European Language Resources Association (ELRA), Valletta, Malta. ISBN 2-9517408-6-7.

Esuli, A. and Sebastiani, F. (2006a). Determining Term Subjectivity and Term Orientation for Opinion Mining. In *Proceedings of EACL '06: Conference of the European Chapter of the Association of Computational Linguistics*. Trento, Italy.

Esuli, A. and Sebastiani, F. (2006b). SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings from LREC'06: The 5th Conference on Language Resources and Evaluation.Proceedings from LREC'06: The 5th Conference on Language Resources and Evaluation*. Genoa, Italy.

Fama, E.F. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, *38* (1): 34–105.

Fama, E.F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25* (2): 383–417.

Feldman, R., Rosenfeld, B., Bar-Haim, R., and Fresko, M. (2011). The Stock Sonar - Sentiment Analysis of Stocks Based on a Hybrid Approach. In D.G. Shapiro and M.P.J. Fromherz, eds., *Proceedings from IAAI '11: Innovative Applications of Artificial Intelligence Conference*. AAAI.

Fung, G.P.C., Yu, J.X., and Lam, W. (2002). News Sensitive Stock Trend Prediction. In M.S. Cheng, P.S. Yu, and B. Liu, eds., *Proceedings from PAKDD '02: Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*, 481–493. Springer. ISBN 3-540-43704-5.

Gallagher, L.A. and Taylor, M.P. (2002). Permanent and Temporary Components of Stock Prices: Evidence from Assessing Macroeconomic Shocks. *Southern Economic Journal*, *69* (2): 345–362.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2009). Large-Scale Sentiment Analysis for News and Blogs. In *Proceedings from ICWSM'07: International AAAI Conference on Weblogs and Social Media*. Boulder, Colorado, USA.

Hatzivassiloglou, V. and Wiebe, J. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *Proceedings from ICCL '00: The 18th International Conference on Computational Linguistics*. New Brunswick, NJ.

Hatzivassiloglou, V. and McKeown, K.R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings from EACL '97: The 8th Conference on European Chapter of the Association for Computational Linguistics*, 174–181. Association for Computational Linguistics, Morristown, NJ, USA.

Hearst, M.A. (1992). *Direction-based Text Interpretation as an Information Access Refinement*, 257–274. L. Erlbaum Associates Inc., Hillsdale, NJ, USA. ISBN 0-8058-1189-3.

Henrikson, J.U. (2011). The Growth of Social Media: An Infographic. Research Survey. The Search Engine Journal. Retrieved from http://www.searchenginejournal.com/wp-content/uploads/2011/08/20110824SocialMediaBlack.pdf.

Hillmann, K.H. (1994). *Wörterbuch der Soziologie*. Kroener Alfred GmbH + Co., fourth edition.

Hsu, C.W., Chang, C.C., and Lin, C.J. (2003). A Practical Guide to Support Vector Classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings from AAAI '04: Nineteenth National Conference on Artificial Intelligence*. San Jose, California, USA.

Hu, Y., Lu, R., Chen, Y., and Duan, J. (2007a). Using a Generative Model for Sentiment Analysis. *Computational Linguistics and Chinese Language Processing*, *12* (2): 107–126.

Hu, Y., Lu, R., Chen, Y., and Duan, J. (2007b). Using a Generative Model for Sentiment Analysis. *Computational Linguistics and Chinese Language Processing*, *12* (2): 107–126.

Jaynes, E.T. (1957). Information Theory and Statistical Mechanics. II. *Physical Review*, *106* (2): 171–190.

Jeong, Y., Kim, Y., Kim, S., Myaeng, S.H., and Oh, H.J. (2009). Generating and Mixing Feature Sets from Language Models for Sentiment Classification. In *Proceedings from NLP-KE '09: International Conference on Natural Language Processing and Knowledge Engineering*, 1 – 8.

Joachims, T. (1999). Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings from ICML '99: The 16th International Conference on Machine Learning*, 200–209. Morgan Kaufmann Publishers, San Francisco, US, Bled, SL.

Kahneman, D. (1979). Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, *47*: 263–292.

Kamps, J., Marx, M., Mokken, R.J., and De Rijke, M. (2004). Using WordNet to Measure Semantic Orientation of Adjectives. In *Proceedings from LREC '04: The fourth international conference on Language Resources and Evaluation*. Lisbon, Portugal.

Kennedy, A. and Inkpen, D. (2005). Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 110–125.

Kim, S.M. and Hovy, E. (2005). Automatic Detection of Opinion Bearing Words and Sentences. In *Proceedings from IJCNLP '05: The First International Joint Conference on Natural Language Processing*. Jeju Island, Korea.

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings from IJCAI '95: The 14th International Joint Conference on Artificial intelligence*, IJCAI'95, 1137–1143. Morgan Kaufmann Publishers Inc.

König, A.C. and Brill, E. (2006). Reducing the Human Overhead in Text Categorization. In *Proceedings from ACM SIGKDD '06: The 12th International Conference on Knowledge Discovery and Data Mining*, KDD '06, 598–603. ACM, New York, NY, USA. ISBN 1-59593-339-5.

Lipsman, A. (2007). Online consumer-generated reviews have significant impact on offline purchase behavior. Research Survey. comScore Inc. and The Kelsey Group. Retrieved from http://www.comscore.com/press/release.asp?press=1928.

Liu, B., Li, X., Lee, W.S., and Yu, P. (2004a). On Transductive Support Vector Machines. In *Proceedings from AAAI'04: The 19th National Conference on Artifical Intelligence*, 425–430.

Liu, B., Li, X., Lee, W.S., and Yu, P.S. (2004b). Text Classification by Labeling Words. In *Proceedings from AAAI '04: The 19th National Conference on Artifical Intelligence*, 425–430.

Lu, B., Tsou, B.K., and Kwong, O.Y. (2008). Supervised Approaches and Ensemble Techniques for Chinese Opinion Analysis at NTCIR-7. In *Proceedings from NTCIR-7'08: NII Test Collection for IR Systems Workshop*. Tokyo, Japan.

Lyman, P. and Varian, H.R. (2003). How much information? Technical report, University of California, Berkeley.

McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured Models for Fine-to-Coarse Sentiment Analysis. Technical report, Google, Inc.

Melville, P., Gryc, W., and Lawrence, R.D. (2009). Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In *Proceedings from KDD '09: The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1275–1284. ACM, New York, NY, USA. ISBN 978-1-60558-495-9.

Mihalcea, R., Banea, C., and Wiebe, J. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic.

Mittermayer, M.A. and Knolmayer, G. (2006). Text Mining Systems for Market Response to News: A Survey.

Moilanen, K. and Pulman, S. (2007). Sentiment Composition. In *Proceedings from RANLP '07: Conference of Recent Advances in Natural Language Processing*, 378–382.

Montgomery, D.C., Peck, E.A., and Vining, G.G. (2001). *Introduction to Linear Regression Analysis*. Wiley series in probability and statistics. Wiley, New York, NY [u.a.], 3. ed edition. ISBN 0-4713-1565-6.

Mullen, T. and Collier, N. (2004). Sentiment Analysis Using Support Vector Machines with Diverse Information Sources. In *Proceedings of EMNLP '04: Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain.

Myers, R.H. (1994). *Classical and Modern Regression with Applications*. PWS-KENT Publishing Company, Boston, MA, second edition.

Na, J.C., Sui, H., Khoo, C., Chan, S., and Zhou, Y. (2004). Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews. In *Proceedings from ISKO'04: Conference of the International Society for Knowledge Organization*.

O'Keefe, T. and Koprinska, I. (2009). Feature Selection and Weighting Methods in Sentiment Analysis. In *Proceedings from AusDM '08: 7th Australasian Data Mining Conference*.

Opfine (2012). An Online Sentiment Analysis Tool. Retrieved from http://www.opfine.com.

Osgood, C.E., Suci, G.J., and Tannenbaum, P.H. (1971). The Measurement of Meaning. Technical report, University of Illinois Press.

Osgood, C.E. (1974). Probing Subjective Culture Part 1: Cross-linguistic Tool-making. *Journal of Communication*, *24* (1): 21–35. ISSN 1460-2466.

Paltoglou, G., Gobron, S., Skowron, M., Thelwall, M., and Thalmann, D. (2010). Sentiment Analysis of Informal Textual Communication in Cyberspace. In *Proceedings from ENGAGE '10*, 13–25. Zermatt, Switzerland.

Pang, B. and Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings from ACL '04: 2004 Conference of the Association for Computational Linguistics*. Barcelona, Spain.

Pang, B. and Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, *2* (1-2): 1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings from EMNLP '02: Conference on Empirical Methods in Natural Language Processing*. Philadelphia, PA, USA.

Popescu, A.M. and Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews. In *Proceedings from HTL-EMNLP '05: Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 339–346. Association for Computational Linguistics.

Prabowo, R. and Thelwall, M. (2009). Sentiment Analysis: A Combined Approach. *Journal of Informetrics*, *3* (2): 143–157.

Qian, B. and Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence*, *26* (1): 25–33.

Qiu, G., Liu, B., Bu, J., and Chen, C. (2009). Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings from IJCAI '09: The 21st International Joint Conference on Artificial Intelligence*. Pasadena, California, USA.

Ren, J., Lee, S.D., Chen, X., Kao, B., Cheng, R., and Cheung, D.W.L. (2009). Naive Bayes Classification of Uncertain Data. In *Proceedings from ICDM '09: The IEEE International Conference on Data Mining*, 944–949.

Rish, I. (2001). An empirical study of the naive Bayes classifier. In *Proceedings from IJCAI '01: Workshop on Empirical Methods in Artificial Intelligence*.

Rittman, R. and Wacholder, N. (2008). Adjectives and Adverbs as Indicators of Affective Language for Automatic Genre Detection. In *Proceedings from AISB '08: Symposium on Affective Language*. Aberdeen, Scotland.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, 2000.

Sack, W. (1994). On the Computation of Point of View. In *Proceedings of AAAI*, 1488. Student abstract.

Safavian, S.R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *Systems, Man and Cybernetics, IEEE Transactions on*, *21* (3): 660–674.

Sehgal, V. and Song, C. (2007). SOPS: Stock Prediction Using Web Sentiment. In *Proceedings from ICDMW '07: Seventh IEEE International Conference on Data Mining Workshops*, 21–26.

Sentz, K. and Ferson, S. (2002). Combination of evidence in Dempster-Shafer theory. Technical report.

Shafer, G. (1985). Probability Judgement in Artificial Intelligence. In *Proceedings from UAI '85: The First Annual Conference on Uncertainty in Artificial Intelligence*, 91–98. Elsevier Science, New York, NY.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton.

Simeon, M. and Hilderman, R. (2008). Categorical Proportional Difference: A Feature Selection Method for Text Categorization. In *Proceedings from ADCS '09: 14th Australasian Document Computing Symposium*, 201–208. Roddick, J. F., Li, J., Christen, P. and Kennedy, P. J., Eds. ACS.

Sindhwani, V. and Melville, P. (2008). Document-Word Co-Regularization for Semi-supervised Sentiment Analysis. In *Proceedings from ICDM '08: Eighth IEEE International Conference on Data Mining*. Washington, DC, USA.

Soroka, S.N. (2006). Good News and Bad News: Asymmetric Responses to Economic Information. *The Journal of Politics*, *68*: 372–385.

Stoyanov, V., Cardie, C., and Wiebe, J. (2005). Multi-Perspective Question Answering Using the OpQA Corpus. In *Proceedings from HTL-EMNLP '05: Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 923–930. Association for Computational Linguistics.

Strapparava, C. and Valitutti, A. (2004). WordNet-Affect: an affective extension of WordNet. In *Proceedings of LREC*, volume 4, 1083–1086.

Takamura, H., Inui, T., and Okumura, M. (2005). Extracting Semantic Orientations of Words using Spin Model. In *Proceedings from ACL '05: The Association for Computational Linguistics*, 133–140.

Tan, S., Cheng, X., Wang, Y., and Xu, H. (2009). Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In *Proceedings from ECIR '09: The 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, 337–349. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-642-00957-0.

The Nielsen Company (2009a). Global Trust in Advertising and Brand Messages. Research Survey. Retrieved from http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/07/pr_global-study_07709.pdf.

The Nielsen Company (2012b). Global Trust in Advertising and Brand Messages. Research Survey. Retrieved from http://www.fi.nielsen.com/site/documents/NielsenTrustinAdvertisingGlobalReportApril2012.pdf.

The Stock Sonar (2012). An Online Sentiment Analysis Tool. Retrieved from http://www.thestocksonar.com.

Tong, S. and Koller, D. (2002). Support Vector Machine: Active Learning with Applications to Text Classification. *J. Mach. Learn. Res.*, *2*: 45–66. ISSN 1532-4435.

Tsatsaronis, G., Vazirgiannis, M., and Androutsopoulos, I. (2007). Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In *Proceedings from IJCAI '07: 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India.

Turney, P. (2002). Thumbs up or thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings from ACL '02: 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, USA.

Verma, S. and Bhattacharyya, P. (2008). Incorporating Semantic Knowledge for Sentiment Analysis. In *Proceedings from ICON '08: 6th International Conference on Natural Language Processing*. Macmillan Publishers.

Wang, L., Shen, X., and Pan, W. (2007). On Transductive Support Vector Machines. In *Prediction and Discovery*. American Mathematical Society.

Weichselbraun, A., Gindl, S., and Scharl, A. (2011). Using games with a purpose and bootstrapping to create domain-specific sentiment lexicons. In *Proceedings from ACM CIKM '11: The 20th ACM International Conference on Information and Knowledge Management*, CIKM, 1053–1060. ACM, New York, NY, USA. ISBN 978-1-4503-0717-8.

Whissell, C. (1989). The dictionary of affect in language. *Emotion - Theory, Research and Experience*, *4*: 113–131.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using Appraisal Taxonomies for Sentiment Analysis. In *Proceedings from CIKM '05: ACM SIGIR Conference on Information and Knowledge Management*. Bremen, Germany.

Whiteson, S., Tanner, B., Taylor, M.E., and Stone, P. (2011). Protecting Against Evaluation Overfitting in Empirical Reinforcement Learning. In *Proceedings from ADPRL '11: Symposium on Adaptive Dynamic Programming and Reinforcement Learning*.

Wiebe, J. and Bruce, R. (1995). Probabilistic Classifiers for Tracking Point of View. In *Proceedings from AIII '95: Symposium on Empirical Methods in Discourse Interpretation and Generation*, 181–187.

Wiebe, J. and Riloff, E. (2005). Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Proceedings from CICLing '05: International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City, Mexico.

Wiebe, J.M., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, *30*: 277–308.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (2010). A Survey on the Role of Negation in Sentiment Analysis. In *Proceedings from NeSp-NLP '10: The Workshop on Negation and Speculation in Natural Language Processing*, 60–68. Association for Computational Linguistics, Morristown, NJ, USA.

Wikipedia (2012). Overfitting in Training of Supervised Methods. Retrieved from http://en.wikipedia.org/wiki/File:Overfitting_svg.svg.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings from HLT/EMNLP '05: Joint Conference on Human Language Technologies and Empirical Methods in Natural Language Processing*. Vancouver, B.C., Canada.

Wong, W., Liu, W., and Bennamoun, M. (2008). Featureless Data Clustering. In M. Song and Y. Wu, eds., *Handbook of Research on Text and Web Mining Technologies*. IGI Global.

Xu, C. and Zhou, Y. (2007). Transductive Support Vector Machine for Personal Inboxes Spam Categorization. In *Proceedings from CISW'07: International Conference on Computational Intelligence and Security Workshops*, 459–463.

Yager, R.R., Kacprzyk, J., and Fedrizzi, M., eds. (1994). *Advances in the Dempster-Shafer Theory of Evidence*. John Wiley & Sons, Inc., New York, NY, USA. ISBN 0-471-55248-8.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings from EMNLP '03: Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan.

Zadeh, L.A. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, *7* (2): 85–90.

Zhai, C. (2008). Statistical Language Models for Information Retrieval A Critical Review. *Found. Trends Inf. Retr.*, *2* (3): 137–213. ISSN 1554-0669.

Zhang, W. and Skiena, S. (2010). Trading Strategies to Exploit Blog and News Sentiment. In W.W. Cohen and S. Gosling, eds., *Proceedings from ICWSM '10: The International AAAI Conference on Weblogs and Social Media*. The AAAI Press.

Zhou, L. and Chaovalit, P. (2008). Ontology-supported polarity mining. *J. Am. Soc. Inf. Sci. Technol.*, *59*: 98–110. ISSN 1532-2882.

Zlotnick, J. (1970). Bayes' Theorem for Intelligence Analysis. In *Proceedings of the Conference on the Diagnostic Process.*