Opinion Mining in Newspapers for a Media Response Analysis

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Thomas Scholz

aus Meerbusch

Düsseldorf, Oktober 2013

Aus dem Institut für Informatik
der Heinrich-Heine Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Stefan Conrad
Koreferent: Prof. Dr. Martin Mauve
Tag der mündlichen Prüfung: 04.02.2014

Dedicated to

Thorsten, Margrit, and Stefan

Acknowledgements

The results presented in this thesis are the outcome of my three years of research at the Databases and Information Systems Group of the Department of Computer Science at the Heinrich Heine University of Düsseldorf.

First of all, I would like to say thank you to my supervisor, Prof. Dr. Stefan Conrad. His door was always open for me to come in with questions or problems from the time when I was an undergraduate and needed a schedule for my subsidiary subject to the time when this thesis was nearly printed. I knew very early, that I would like to work in his group one day. Also, I would like to thank Prof. Dr. Martin Mauve for his interest in my research and his willingness to be the second assessor. Likewise, I thank the third reviewer Prof. Dr. Jörg Scheidt for the same reasons.

We would like to expand my compliments to Dr. Johanna Vompras and to Dr. Sadet Alcic, who supervised my first steps in academic research. Special thanks go to my first room neighbour Dr. Tim Schlüter, who taught me a lot of practical things about working in a research department. I would also like to say thank you to my colleagues Ludmila Himmelspach (our rumour generator), Jiwu Zhao (the karate bowler), Dr. Katrin Zaiß (Doctrine), and my new colleagues Magdalena Rischka (redundancy creates security), Daniel Braun (inventor of so many things whose names include his surname), Michael Singhof (the organiser), and Robin Küppers (we are still waiting for his sarcasm blog). Last but not least, I say thank you to Sabine Freese and Guido Königstein for time, patience, support, suggestions, and their positive spirit.

All of you are the reasons, why I will miss my work at the University of Düsseldorf. The time was to short.

And I would like to thank Dr. Isabel Wolters for our many discussions at the beginning of the ATOM project and before. I will never forget the day, when we received the funding approval, and our evenings in the steakhouse.

Finally, I would like to thank everybody, who has supported me in the last three years, especially my family and my friends.

Düsseldorf, Germany
October, 2013 *Thomas Scholz*

Abstract

A part of the broad research domains *Knowledge Discovery* and *Information Retrieval* deals only with *Data Mining* in texts: *Text Mining*. In general, Text Mining tries to obtain knowledge by identifying patterns in textual data. One of its most important areas is *Opinion Mining*, which is the main topic of this thesis.

Opinion Mining is a far-reaching research area, because it is potentially interesting for many different fields of application as well as its results are very valuable: Opinions are analysed in reviews of products, services, etc. to create very detailed reports about the subject of the reviews or to identify fake or spam reviews. Furthermore, contributions for Opinion Mining in Social Media try to discover opinions in these networks such as Twitter, Facebook, and Youtube. We concentrate on Opinion Mining in news articles, because automatically extracted opinions from news have a high economic value, especially for media monitoring services, but at the same time, this domain has been rather neglected by approaches for Opinion Mining.

Thus, we complement this research area by tasks of a *Media Response Analysis*, which includes the extraction of statements, the classification of the tonality, and the determination of viewpoints. To establish these tasks within the Opinion Mining community, we published an own dataset of a Media Response Analysis (MRA).

A major challenge is the extraction of statements for an MRA. In this step, the text parts of a news article have to be identified, which are most relevant for analysis objects and contain opinions, even if the tonality of the opinion is neutral.

The classification of the tonality for a given text or text part represents the most difficult task for almost every Opinion Mining approach. Many contributions involve only this step and apply a broad spectrum of techniques to tackle this problem: The creation of sentiment dictionaries, the analysis of contextual information, machine learning, heuristic rules, profoundly linguistic analyses, and many more. During this thesis we investigate many characteristics for the determination of tonality in our domain in contrast to recent research and propose a very well working approach for the tonality classification of statements in newspaper articles, which is adjusted to the requirements of a practical solution and achieves better results for our task than current state-of-the-art techniques.

Extracted and rated statements are difficult to assess for MRA, if they do not contain any information about the viewpoint. To complete a fully automated solution of Opinion Mining for a Media Response Analysis, we explain and evaluate our ontology-based approach for the determination of viewpoints.

Zusammenfassung

Ein Teil der weitreichenden Forschungsgebiete von *Knowledge Discovery* und *Information Retrieval* beschäftigt sich nur mit *Data Mining* in Texten: *Text Mining*. Allgemein versucht man beim Text Mining durch Mustererkennung Wissen aus textuellen Daten zu ziehen. Eines der bekanntesten Gebiete in Text Mining ist *Opinion Mining*, das grundlegende Thema dieser Arbeit.

Opinion Mining ist ein weitreichender Forschungszweig, weil Opinion Mining für viele Anwendungsgebiete interessant ist und gleichzeitig die Resultate potentiell wertvoll sind: Meinungen können in Bewertungen zu Produkten, Dienstleistungen, etc. untersucht werden, um detaillierte Berichte über den Gegenstand der Bewertung zu erstellen oder um nicht glaubwürdige oder nutzlose Bewertungen zu identifizieren. Im Bereich soziale Netzwerke versucht man Meinungen z. B. bei Twitter, Facebook, Youtube, etc. zu entdecken. Wir konzentrieren uns auf Opinion Mining in Zeitungsartikeln, weil automatisch extrahierte Meinungen aus Zeitungen von großem wirtschaftlichen Wert sind, besonders für Medienbeobachter und ihre Kunden. Zugleich ist Opinion Mining in Zeitungen von aktuellen Arbeiten eher vernachlässigt worden.

Deshalb komplettieren wir dieses Forschungsgebiet um die Herausforderungen einer *Medienresonanzanalyse*, die eine Aussagenextraktion, eine Tonalitätsklassifikation und eine Perspektivbestimmung umfasst. Wir veröffentlichten einen eigenen Datensatz einer solchen Analyse um diese Herausforderungen noch weiter zu etablieren.

Eine Schlüsselaufgabe ist die Extraktion von Aussagen für eine Medienresonanzanalyse (MRA). In diesem Schritt müssen die Abschnitte von Zeitungsartikeln identifiziert werden, die relevant für die Analyseobjekte sind und eine Meinung beinhalten, selbst wenn die Tonalität dieser Meinung neutral ist.

Die Klassifikation der Tonalität für einen gegebenen Text oder Textteile ist meistens die schwierigste Aufgabe innerhalb einer automatischen Meinungsanalyse. Viele Ansätze drehen sich nur um diesen Schritt und schlagen ein breites Spektrum an Techniken für dieses Problem vor: Generierung von Tonalitätswörterbüchern, Analyse des Kontextes, maschinelles Lernen, heuristische Regeln, tiefgehende sprachliche Analysen und vieles mehr. Innerhalb dieser Arbeit stellen wir viele Besonderheiten für die Tonalitätsbestimmung in Zeitungen im Vergleich zu aktuellen Arbeiten heraus und entwickeln einen sehr gut funktionierenden Ansatz für die Klassifikation der Tonalität in Aussagen aus Zeitungsartikeln, der an die Voraussetzungen für einen Einsatz in der Praxis angepasst ist und bessere Resultate erzielt als der aktuelle Stand der Technik.

Extrahierte und mit Tonalität versehene Aussagen sind allerdings immer noch schwierig zu bewerten innerhalb einer MRA, wenn keine Informationen über die Perspektive verfügbar sind. Deshalb vervollständigen wir unsere automatische Lösung für eine Medienresonanzanalyse um eine Perspektivbestimmung durch einen ontologiebasierten Ansatz.

Contents

1

Introduction

Opinion Mining as the identification and evaluation of opinionated text is a challenging, but at the same time a scientifically and economically interesting question. Opinion Mining is a branch of Text Mining and involves several techniques from the research areas of Knowledge Discovery, especially Data Mining, and Information Retrieval [AZ12]. The issue has grown importance in the light of recent developments in the increasing availability of textual data [PL08, Liu10], which contain potentially valuable opinions such as reviews [PL08, Liu10]. But at the same time, also the number of news texts in the internet is rising as well as their consumption [MB10, HF12]. So, a continuous trend is that an increasing number of news is accessible as textual data and it covers more and more parts of traditional media.

In this introduction, we illustrate how our research in Opinion Mining differs from recent research in this area, because our focus is oriented on newspaper articles and not on reviews as the most recent approaches. In this way, we complement research in Opinion Mining about tasks of a Media Response Analysis and news in general, because we believe that Opinion Mining in the news is potentially more interesting than in reviews. We explain our beliefs in the next section. Although previous contributions concerned Opinion Mining in the news, they are mainly concentrated on the analysis of single words or quotations. We will show that this is not sufficient for a comprehensive solution for Opinion Mining in newspaper articles. This doctoral thesis is emerged in the context of the ATOM project, which was performed with cooperation partner in the industry. Further details are mentioned in section 1.3. Although this thesis is very practically motivated, we introduce many theoretical aspects and perform numerous experiments to prove our convictions or to recognize assumptions, which are wrong or should not be made.

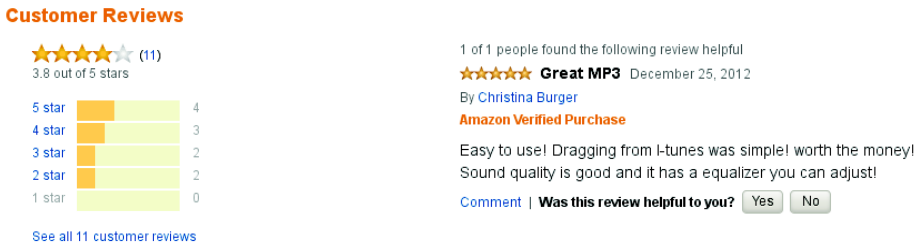

Figure 1.1: A rating overview and a helpful product review for a Sony MP3 player (8GB Sony Walkman NWZ-E374, collected from amazon.com on 16th May 2013).

1.1 Motivation

As mentioned above, academic research concerning Opinion Mining deals primarily with reviews. The connection between Opinion Mining, also known as Sentiment Analysis, and reviews is so pronounced that reviews (product reviews, film reviews, hotel reviews) manage to appear in common definitions of Opinion Mining. The following one is from the textbook "Mining Text Data" [AZ12]:

- "**Opinion Mining from Text Data:** A considerable amount of text on web sites occurs in the context of product reviews or opinions of different users. Mining such opinionated text data to reveal and summarize the opinions about a topic has widespread applications, such as in supporting consumers for optimizing decisions and business intelligence." ([AZ12], page 8)

This is only one sign, that the research to date has tended to focus on Opinion Mining in customer reviews. Another sign is the numerous contributions tackling Opinion Mining in reviews such as [PLV02, DLY08, BES09, DTCY10, SPV11], to name a few. At the same time, far too little attention has been paid to Opinion Mining in newspaper articles, which has become a central issue for media monitoring and media analysis. Today, classical print news are merging with online news. Newspaper publishers present their articles online or write completely new and very up-to-date articles for their Web portal. The amount of potentially useful articles increased substantially in the last years and this trend is continuing.

While research in Opinion Mining is concentrated on analysing product and film reviews, we believe that Opining Mining in news articles is more interesting than in reviews because review systems such as webstores (e.g. amazon), movie review sites (like IMDB), online booking services (e.g. booking.com) already provide a quick overview of the sentiment about products, films, and services by overall scores or selected reviews. The users can also rate reviews as helpful or not and the most helpful reviews are selected. Figure 1.1 depicts an overall rating of a MP3 player taken from amazon.com. The average score is 3.8 of 5 stars, and a table below the score shows the distribution over the different reviews. On the right side, one of the three most helpful

reviews is depicted, which are also shown on page of the product. So, companies, which sell these products, and consumers get a quick overview about their products or the products, in which they are interested. A web crawler system can collect this information from different webstores and combine the results [PL08].

In the field of newspaper articles, the creation of a similar overview requires a big human effort. The offer for such an overview system, which can perform and illustrate a Media Response Analysis (MRA) [WN07], represents a separate business segment for analysis services and is very interesting for companies, organisations such as political parties, associations, or distinguished public figures.

Today, an MRA is carried through with a huge amount of human effort in media monitoring companies. After the news texts, which contain a predefined search term, have been collected by Web crawlers, domain specialists (the so-called media analysts) read these texts and mark statements which are relevant to the customers and set the polarity of the opinion.

In contrast to customer's reviews or some contributions in social media, news items are not as subjective as these [BSK+10]. Furthermore, not all parts of a news article are interesting for an opinion analysis or to put it more succinctly, belong to or affect the results. Therefore, many approaches for newspaper articles work on quotations [PSB07, BSG+09, BSK+10]. In these approaches, one quotation represents opinion and the quotation can be rated in terms of the polarity of sentiment. Besides, also contributions which are concentrated on words and phrases exist [WWH05, WR05, WWH09]. We will take up these contributions later again. Also, there can be different polarities of opinions in different parts of one article.

For example, if we take a look at figure 1.2, it shows a news article about protests against cuts in solar funding in Germany. This article has relevance for a governing party or party of the opposition, for example. In the first paragraph, the government is criticized for plans to cut the funding in the solar industry. Also, the first three sentences of the third paragraph express criticism of the government's plans, in particular the Chancellor of the Federal government, while the last sentence of the second paragraph talks about the solidarity of the opposing parties' leaders and so is relevant for them. Simultaneously, there are parts, which are not relevant for any party (the first sentence in the second paragraph is one example). As a result, Opinion Mining in newspaper articles means firstly that text parts have to be identified, which contain an opinion. This is not such a big deal in customer reviews, where more or less the whole review is one opinion. It looks a little bit different in film reviews, book reviews or similar items, because sometimes an objective part of text describes the plot. But this only puts more noise to the Sentiment Analysis tasks and can be filtered out [TBS09, SPV11]. In the news, it creates a completely new task. These texts parts with opinions are called statements in a Media Response Analysis, so we call the task

Over 11,000 people protest against solar exist in Berlin

Organizers appeal to the German Bundestag to give up radical cuts in solar funding

Berlin (ots) – On Monday, around 11,000 people have protested against "solar exist" in Berlin in front of Brandenburg Gate. This is an impressive picture to the Federal Government to give up radical cuts in expansion of solar energy and solar funding.

The German Association of the Solar Industry (BSW-Solar) together with German Trade Union Confederation (DGB), the Industrial Metal Union (IG Metall), the Mining, Chemicals and Energy Trade Union (IG BCE), and the German Environmental Aid (DUH) had invited for the mass rally at the Brandenburg Gate. On the demonstration, top-level politicians such as Sigmar Gabriel, Jürgen Trittin, and Gregor Gysi express solidarity with the employees of the solar industry.

The organizers appeal to the members of the German Bundestag and the Federal Chancellor to forego radical cuts in the anyway declining solar funding as far as possible. The solar industry fears a market collapse of up to 75 percent in cases where the legislative initiative will not be stopped or improved considerably. Otherwise the operation of new solar plants will be no longer viable preponderantly, a wave of insolvencies unavoidable, and 100,000 workplaces would be in danger. The energy turnaround cannot be achieved without a powerfully further expansion of use of solar electricity.

"Instead of accelerating, the government brakes in the energy turnaround. It is absurd that the development of photovoltaic should be massively limited precisely at the moment, when the costs for new solar plants decrease significantly and the funding of solar energy has hardly any considerable effects on electricity rates", says Rainer Baake, managing director of the German Environmental Aid. Prof. Dr. Eicke Weber, director of the Fraunhofer Institute for solar energy systems in Freiburg, share this view like many other scientists: "This rapid knee-jerk reaction is without any scientific foundation. On the verge of a breakthrough, we run the risk of gambling away careless the fruits of a long lasting technological leadership. Germany must come back to a predictable economy and energy policy."

[...]

Figure 1.2: A translated example [SCH12] of a news article.

statements extraction. Earlier approaches [BSG+09, BSK+10] tackle this problem by the extraction of quotations. In the last shown paragraph, the article of figure 1.2 contains two relevant statements for parties which contain a quotation. Of course, the quotations contain an opinion and so quotation-based approaches for Opinion Mining in newspaper articles lead to results with a high precision. But the recall of such approach is typically rather low. In our article of figure 1.2, this approach will find no more than two (or three, the first sentence contains a small proportion of quoted text) of five relevant statements. And this is even a generous example for a quotation-based solution. We will show this later in this thesis.

Figure 1.2 illustrates also two other aspects: On the one hand, not all parts have the same polarity of sentiment. In contrast to most of the other statements, the sentence about the party leaders' solidarity contains a more positive sentiment. In the area of an MRA, we speak of the tonality of a statement and mean that it can be a positive, a neutral, or a negative statement. Also, further gradations are conceivable, but they are used less in practice (we will show in this thesis, how difficult the distinction of neutral and subjective statements is even for humans). Unlike in reviews, documents containing different polarities or orientations of opinions represent the normal case, not the exceptional case.

On the other hand, a second effect is strongly connected to this matter of fact. The tonalities of statements belong to a perspective. We will speak in this thesis about viewpoints. Under the viewpoint of a governing party, the statements express a negative tonality mainly. Under the viewpoint of the opposition, the situation is different, because at least one sentence mentions the opposition in a positive way. Thus, the statements get a concrete viewpoint. This means that the statement has one certain tonality for one certain viewpoint. Nevertheless, this also includes that one statement can have two different tonalities for two different viewpoints or the same tonality for two different viewpoints. Viewpoints provide a lot of new aspects and are hardly present in reviews. Only comparative sentences such as "The Sony Walkman is better than Samsung's players." can be seen as a related issue, which attract some interest in scientific research [GL08].

1.2 Contributions

In the following section, we briefly describe the main contributions of our research overviewed by this thesis. Besides the main topic of Opinion Mining, it mentions some earlier work on Information Extraction and especially Information Retrieval, which also gave us pointers regarding the requirements of our research in Opinion Mining, but these contributions are mentioned very briefly and for reasons of completeness, so that we avoid losing our main focus.

We published a publicly available evaluation corpus [SCH12] of a Media Response Analysis for Opinion Mining in newspaper articles for the community. On this corpus, tasks such as opinion extraction, the sentiment classification (we will later refer to the classification of tonality more precisely), and viewpoint determination can be trained and evaluated. This corpus is called the pressrelations dataset (one translated example was already shown in this chapter in figure 1.2).

To mine the opinions for an MRA, we developed a powerful technique in order to identify and extract relevant statements [SC13a]. This step can be adjusted for different analysis objects (for a list of parties, companies, organisations, people, or a combination of these entities). This method is based on two machine learning techniques to classify sentences as relevant or not and to filter misclassified examples, before the resulting set of sentences are combined to statements.

Furthermore, a profound analysis [SCW12, SC13b] of characteristics for Opinion Mining in newspaper articles was performed in order to develop an algorithm for classification of the tonality. Here, we have investigated the characteristics of news texts [SCW12] and linguistic factors [SC13b] for Opinion Mining tasks, also the specifics of neutral examples and the problems of integrating them during a classification process.

On this way, we created algorithms and methods for solving partial aspects of Opinion Mining. We compared different methods [SCW12] for the creation of sentiment dictionaries and to determine the polarity of sentiment (distinction between positive and negative statements), which we improved by new weighting methods and linguistic features for news [SC13b]. With all these steps, we gain relevant knowledge for our final solution.

As a result, we present an algorithm [SC13c] extracting a graph of a training collection in order to identify tonality indicating word connections. These connections are used by an entropy-based weighting to create tonality features for machine learning. By these features, machine learning is able to perform a tonality classification of statements quickly (the approach requires less training) and precisely. We evaluate and compare this technique against four state-of-the-arts methods in Opinion Mining [WWH09, DLY08, SPV11, TBT+11].

To determine a concrete viewpoint of a statement, we propose an ontology-based approach [SC12] to handle different viewpoints of statements and viewpoint features, which give machine learning the ability to learn the influence of viewpoints for the tonality and gain an accuracy improvement for the classification. Within this evaluation, we also provide some characteristics of the role of viewpoints of an MRA and the limits for an automated approach to switch between different viewpoints.

Besides, we introduce Information Extraction techniques [SAC09] to improve image annotations [VSC08] for Multimedia Information Retrieval, as well as we develop a machine learning-based method [SC11] for very specific subtasks of Text Mining: Style Analysis of Writing, Authorship Attribution, and Web Page Genre Identification. Since this thesis is mainly focused on tasks related to Opinion Mining, these contributions are only discussed shortly in section 2.3.1 and 2.3.4.

1.3 Project ATOM and Cooperation with the Industry

Most of the results of research presented in this thesis are obtained, while the author worked in the research project ATOM. The acronym ATOM stands for Automated Topic Tracking and Opinion Mining for a Media Response Analysis. It is a cooperation project between the Heinrich-Heine University and the pressrelations GmbH.

The pressrelations GmbH operates as a media monitoring company which also analyse media for their customers and report results in a sustainable Media Response Analysis. As a consequence, the company employs over fifty media analysts who collect the data for an MRA. These media analysts read all news articles which are potentially relevant for the company's customers (clients of an MRA), mark sentences which are

relevant for a customer or a customer's competitor, and rate the tonality of a statement.

The aim of the ATOM project is the application of Opinion Mining and Topic Tracking for German news articles to support the media analysts creating a Media Response Analysis. The author worked in this project as the scientific staff member of the Heinrich-Heine University and was in authority mainly for the Opinion Mining components. As well, the author has greatly contributed to plan this project and wrote the research proposal.

The ATOM project is funded by the German Federal Ministry of Economics and Technology under the ZIM-program (Grant No. KF2846501ED1). The ZIM program is a funding program which supports innovation, especially the creation of new products or services as well as the improvement of production processes. The duration of the project is two and a half year and started in July 2011.

1.4 Structure of this Thesis

Opinion Mining forms an important part in the research field of Text Mining. Because of this, we discuss Text Mining in general in the next chapter. Thereby, we introduce especially the preprocessing of text with NLP and Information Extraction, which is an own subtask in Text Mining and plays an important role in this thesis. In chapter 3, we describe related research on Opinion Mining and show a broad range of applications. After that, we start introducing our own contributions to this field and, as the first step in this direction, we explain the creation of our dataset, which is a publicly available Media Response Analysis for the research community. In the chapters 5 to 8, we present our contributions for tackling the problem of Opinion Mining in newspaper articles. It covers the whole process of the human way analysing textual media. It begins with the extraction of relevant statements in chapter 5. Then, we investigate many characteristics of the tonality and Opinion Mining in newspaper articles in chapter 6. We describe the creation of sentiment dictionaries and perform an analysis of typical features for news, especially for the neutral statements. In chapter 7, we explain and evaluate our solution for the tonality classification and the assignment of viewpoints is shown in chapter 8. We draw conclusions, especially for the application in a practical environment, and show starting points for future work in chapter 9.

2

Text Mining

On our top-down journey, on which we approach our goal of Opinion Mining for a Media Response Analysis, we are now getting into the subject of Text Data Mining or Text Mining in short. Text Mining is a broad research area [AZ12], so this introduction contains especially these aspects of Text Mining, which cover the most applications of Text Mining and simultaneously become relevant in the following sections.

In this chapter, we start with the basics of Text Mining by introducing both of its 'parents': Knowledge Discovery and Information Retrieval. Elements, scientific issues, and techniques of these top level areas can be found again in Text Mining. More precisely, Text Mining combine techniques of Data Mining with Information Retrieval techniques in many ways.

After the introduction of Knowledge Discovery and Information Retrieval, we talk about concrete Text Mining applications such as Information Extraction, Text Summarization, Text Clustering, Text Classification, or Topic Tracking. We illustrate each issue with exemplary research work.

In the subsequent part of this chapter we provide an overview about Natural Language Processing (NLP). We explain NLP with our Information Extraction module, which we have implemented for our research, and we illustrate the linguistically theoretical aspects of NLP with examples.

Parts of the introduction of Knowledge Discovery and NLP are published in the book chapter *Opinion Mining für verschiedene Webinhalte* (Opinion Mining for different web contents) [Sch13] within the specialist book *Methoden der Webwissenschaft* (Methods of Web Science) [SV13].

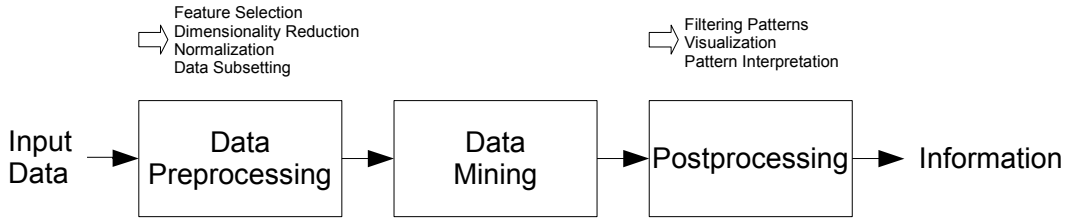

Figure 2.1: The knowledge discovery process by Tan et al. [TSK06].

2.1 Basics of Knowledge Discovery

Knowledge Discovery is often introduced as a process containing several steps from the input data to knowledge as new information [TSK06]. The Knowledge Discovery's pipeline (cf. figure 2.1) consists of preprocessing steps such as feature selection, dimensionality reduction, normalization, and data sub-setting, a post-processing containing filtering patterns, visualization, and pattern interpretation, and especially the data mining step between the input and the output [TSK06]. This step is so prominent and important, that the whole process of Knowledge Discovery is sometimes referred as Data Mining.

Data Mining tries to identify patterns in (usually large) data. For example, Data Mining can analyse the large number of transaction data of a financial institute in order to obtain knowledge about creditworthiness of persons. In this way, Data Mining can be used to identify fraudulent transactions, for example: Data Mining algorithms try to identify patterns, which are characteristic for fraudulent transactions, and these patterns are applied to identify these transactions in future. In the domain of Knowledge Discovery, many machine learning algorithms are proposed for this extraction of patterns. Data Mining has different areas such as Classification, Cluster Analysis, Outlier Detection, and recognition of Association Rules. These techniques are taken up by many approaches in Text Mining [ST00, TBS09, SC11, SB12] and in Opinion Mining [PLV02, HPD+08, WWH09, SPV11, SCW12, SC13a, SC13b], too.

In **Classification**, an algorithm learns from a training collection a classification model. The training collection contains examples with a set of attributes (features) and a class label. The label is an annotation which categorize the class of this training example. Positive, neutral, and negative could be class labels for texts in Opinion Mining, for example. The classification model is able to classify new (unseen) objects as one of the classes in the training data. Classification techniques such Naive Bayes [MST94, Mit97] or Support Vector Machines [Vap82, Vap95] are also very frequently used for Text Mining [NKM01, DVDM01, LM02, LBC09] and even in Opinion Mining [PLV02, WWH09, SPV11], as well as in this thesis. To explain both classifications briefly: a Naive Bayes classifier assumes that all attributes of data are independent

and classify each object to the most likely class based on the attributes values, whereas an SVM looks for a hyperplane, which creates the greatest gap between two groups of data points, and thus is a binary classifier.

Clustering or **Cluster Analysis** tries to group similar objects. The similarity of the objects is normally based on the attributes. Similar objects form a cluster. Clusters provide interesting knowledge. They represent a special kind of customers in data of an online store or documents, which share the same topic, for example. Clustering techniques are called unsupervised in contrast to classification techniques, which are referred as supervised learning, because clustering does not require labeled training data in contrast to classification. The clusters are found through the distribution of the data itself. The similarity of the objects' attribute values are examined, for this a class label is superfluous.

Whereas **Outlier Detection** identifies objects, which are not similar to other examples of the collections, but fall out of their frames. The anomalies within the data represent outstanding objects in the real world: a fraud in bank transaction, an extraordinary good seller, or a very well reviewed film for example. Clustering and Outlier Detection do not play a mayor role in Opinion Mining, because the classes such as positive, neutral, and negative are defined before analysis and very often annotated data exists, but we test Clustering for Opinion Mining later in one of our contributions.

Association Rule Mining is often treated in context of market basket analysis. In this field, analysts want to find out which products are often bought together. A market basket analysis supports the marketing in different aspects. In our context, approaches such as [HPD+08, KRKK09] tackle Opinion Mining with Association Rules. Their basic idea is that the appearance of a word indicates a certain tonality. We discuss and evaluate this idea later on.

2.2 Connections to Information Retrieval

Information Retrieval (IR) concerns all different aspects of searching for information [MRS08], which users request. An user formulates typically a query in search of information. At the beginning of Information Retrieval, the application ranges are mainly scientific publications and library records [MRS08], but with the breakthrough and the growth of the World Wide Web Information Retrieval is needed for other types of content. For example, the search requests can cover multimedia objects such as images, audio or video files, and so on. The sub-field Multimedia Information Retrieval handles multimedia objects. Some aspects of Information Retrieval can be retrieved in Text Mining and Opinion Mining.

One aspect of Information Retrieval, as long as it concerns at least partially tex-

tual content, is also very important for Opinion Mining and Sentiment Analysis: the **weighting of terms**. One very prominent weighting is the TF-IDF (term frequency - inverse document frequency) [Jon72], which is often applied within the subsequently developed vector space model [SWY75]. With both approaches, the similarity of a query and a document can be calculated (distance measures such as the euclidean distance in the vector space model) and also the importance of one term in a document (TF-IDF). We use this idea for our first evaluation in Opinion Mining and later on. With weighting of terms, approaches can express how positive or negative is a certain word for example. This is often estimated through a training. A large collection of reviews can be used for training, for example. As a result, sentiment dictionaries are created, which contain words and their sentiment score.

Also, both disciplines use **dictionaries** to expand textual queries or input with synonyms, for example. In Text Mining and Information Retrieval, approaches sometimes **eliminate stop words** and/or **words are stemmed** or **lemmatized** to their stem form or lemma.

The area of **Text Classification** (cf. section 2.3.4) also forms a part of Text Mining. With Text Classification, it is possible to request only certain types of text documents [DVDM01, LM02], for example. So, queries, which refer to only certain genre of web page or text type (news, book, blog entry, and so on), are possible [DVDM01, LM02]. Furthermore, the task of Opinion Retrieval also is located in the intersection of Information Retrieval and Text Mining. If documents, text parts, or other pieces of information are tagged with a tonality by an Opinion Mining approach, queries such as 'search for negative documents about president Obama' are possible analogically.

As in many Text Mining subtasks, the measures **precision and recall** provide information about the success of a method in Information Retrieval. Generally speaking, precision is the number of correct alarms divided by the number of all alarms, which is the sum of correct alarms and false alarms (cf. equation 2.1).

$$\text{precision} = \frac{|\{\text{correct alarms}\}|}{|\{\text{correct alarms}\}| + |\{\text{false alarms}\}|} \tag{2.1}$$

$$\text{recall} = \frac{|\{\text{correct alarms}\}|}{|\{\text{correct alarms}\}| + |\{\text{false dismissals}\}|} \tag{2.2}$$

For example, a query retrieves five objects in an IR system and two of these objects are relevant and thus correct alarms. So, three objects are false alarms and the precision would be 40%. The recall is defined by the number of correct alarms divided by the number of correct alarms plus the number of false dismissals (cf. equation 2.2). In our example, we imagine that two other objects are relevant, which are rejected falsely. Thereby, the recall would be 50%.

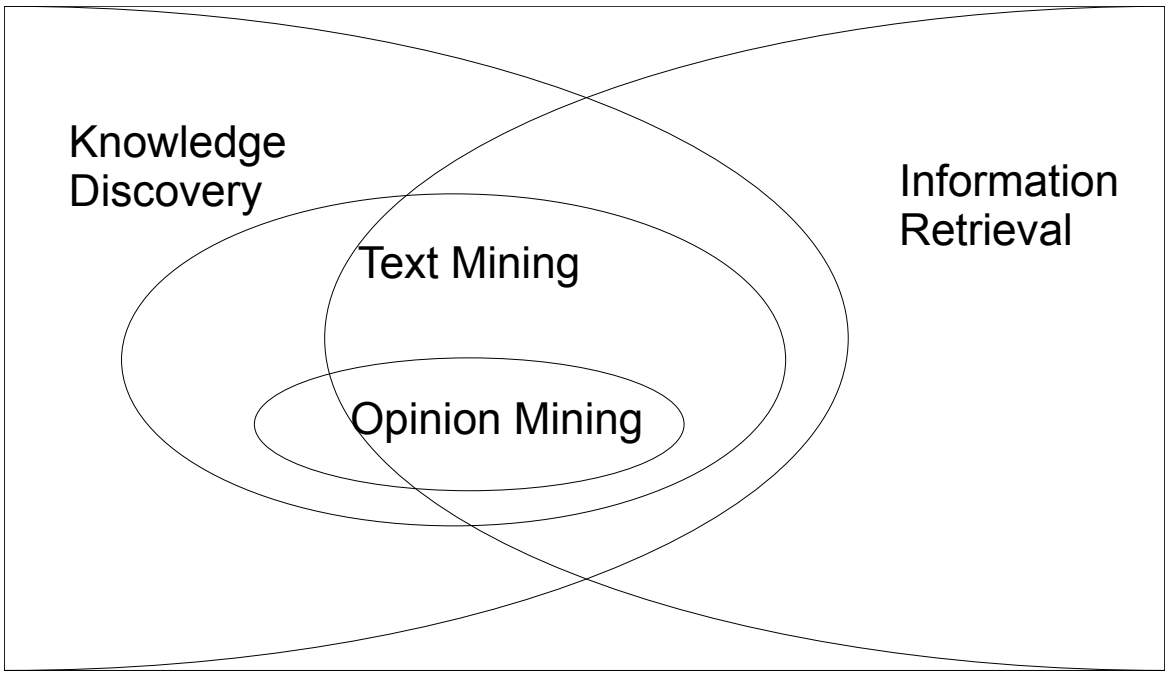

Figure 2.2: Schematic correlation between the areas Knowledge Discovery, Information Retrieval, Text Mining, and Opinion Mining.

2.3 Text Mining Areas

Text Mining or sometimes called Text Data Mining pursues miscellaneous goals. Some techniques try to summarize texts, others extract detailed information about persons, or topics are recognized in large document collections. We summarize the most important Text Mining areas besides Opinion Mining, which are also somehow related to this thesis: Information Extraction, Text Summarization, Text Clustering, Text Classification, and Topic Detection and Tracking.

2.3.1 Information Extraction

Information Extraction tries to collect facts from texts automatically. This means that structured information are obtained from document collections [TDB08]. Typically, this goes beyond results of Natural Language Processing (cf. section 2.4.1), which identifies words as nouns or adjectives and maybe the stem or the lemma of a word. Often Information Extraction (IE) requires rather Natural Language Processing as a preprocessing step. Rudimentary IE identifies entities such persons, organisations, locations, and dates in texts [CMBT02].

More elaborated IE contributions extract further pieces of information, which are triplets, for example [RDF+07]. Triplets represent an unit of a subject, a predicate, and an object. These triplets can be obtained from parse trees, which are generated by syntactic parsing (cf. section 2.4.1, last paragraph). With triplets, question answering

systems can be improved. The triplets can be combined to a semantic graph and in this way a natural language interface is possible for searching [DRF+09]. In addition, these semantic graphs can be utilized for a document visualization [RFM+09]. We will reuse the triplet model for Opinion Mining in section 6.4.3.

To store, organize and process the structured information, many contributions [KED+07, LLX07, LSL+09, LGW09] extract ontologies from textual content. The ontologies organize knowlegde in a hierarchical manner. A hammer is a tool, which is a thing and so on, for example. Entities can be categorized in this way [GKV08]. In addition, an ontology can be considered as knowledge structured in a graph. These knowledge graphs can be used to summarise entities [SPSS10] or allow a knowledge search [ERS+09].

We have shown in early research, that Information Extraction techniques can improve a web mining approach [SAC09] in order to generate automated annotations for images. These annotation can be used efficiently for an IR system for images [VSC08], which we called the GLENARVAN retrieval system [VSC08] and which was developed during the same research project as our web mining technique [SAC09]. The annotations describe the depicted persons, objects, places, and actions on the picture, as well as the photo is assigned to a certain context, in which the photo is taken (for example the club or the national team of a football player).

IE is an important aspect of the contributions in this thesis, because some of our algorithms work with results of our IE module (cf. section 2.4.2). Comparison methods imply their own IE methods, too. Also, some tasks represent an IE task itself such as the extraction of statements.

2.3.2 Text Summarization

Approaches in Text Summarization, as the name implies, try to compose summaries of given texts automatically. Two types of automated Text Summarizations exist: The more popular type selects sentences from the origin text, which are the most important ones, and presents this selection as a summary. The other type composes own sentences or represents the summary in different ways: For example, the summary is represented by several keywords, but these approaches belong also partially to topic detection and tracking (cf. section 2.3.5).

For the first type, which selects important sentences, the previous extraction of important keywords is famous: GenEx [Tur00] and TextRank [MT04] were state-of-the-art techniques with this strategy a long time. And later DegExt [LLA+11] achieved even better results with the same strategy by representing the text as graphs and selecting the nodes with the highest connectivity. We will explain this method, GenEx, and Text Rank in more detail in section 5.2 and 5.3, when we want to find out, if our

task of extracting statements can be solved by Text Summarization. With another strategy, Khosravi et al. [KKE+08] analyze features such as the length of sentences, the position of sentences, the similarity to the title and keywords in order to decide which sentences should be selected for a summary. Also, multi-document summaries can be created by selecting the right sentences [CHT11].

Concerning the second type, one very early approach [AHG99] extracts coreference chains (cf. section 2.4.2) to summarize texts. Here, the coreference chain models a topic, which summarize the text. Tanaka et al. [TKK+09] rewrite lead sentences to summarize news in Japanese.

Moreover, given features and scoring methods can be improved by reinforcement learning [RA12]. Reinforcement learning means here that the approach tries to find an optimal policy [RA12] to construct the summary based on a given score function and given feature representation. So, the summary should maximize the function typically by balancing the tradeoff between redundancy and relevance [RA12].

2.3.3 Text Clustering

Text Clustering requires no labeled data in general, but it tries to figure out groups of similar documents, whatever similar documents mean. They could share the same topic or belong to the same genre.

Slonim and Tishby [ST00] use the information bottleneck method to cluster collections of documents via word clusters. First, they cluster the words which contain the most information about the documents and then they cluster the documents based on the word clusters. In general, the information bottleneck method proposed by Slonim and Tishby [ST99] compresses one variable x in this way, that most information about variable y through x survives.

In addition, Text Clustering can be used to create multi-document summaries. Silveira and Branco [SB12] propose a double clustering strategy: First, they cluster sentences based on overlapping words and subsequences, select one sentence as cluster representative and then these sentences are clustered according to topics, which are represented by keywords.

Hu et al. [HFC+08] cluster news items and medical documents. For the clustering, they enhance similarity measurements of texts by a thesaurus, which includes synonyms, hypernyms, and content-based relations and is created through analysing Wikipedia. Their evaluation shows an improvement of their method in both domains.

Text Clustering is a relevant technique for this thesis, because we will cluster sentences for the extraction of statements. This sub-process works as a filter and increases the performance of our extraction, as we will explain later.

2.3.4 Text Classification

Classification problems relating to texts appear in multiplicity of tasks. The applications such as news filtering, document retrieval, and spam email detection are numerous. Likewise, a high number of classification techniques such as decision trees, pattern-based classifiers, SVMs, neural networks, or Bayesian classifiers [AZ12] are proposed to solve the tasks. Also, Opinion Mining belongs partially to this area, especially when we talk of the sentiment classification, or more precisely the tonality classification. This is a typical task of Text Classification. One example is the approach of Sarvabhotla et al. [SPV11] which classifies film reviews as positive or negative.

Nigam et al. [NLM99] calculate a uniform distribution with the maximum entropy for Text Classification. The method classifies differently labeled texts: Web pages of computer science departments are categorized as student, faculty, course, and project, for example. Mukherjee and Liu classify the gender of the author and apply the method to blog texts [ML10]. Other approaches perform a more detailed analysis by classifying different parts of a text. Taboada et al. [TBS09] identify different types of genre in paragraphs of film reviews.

For very specific subtasks, we created an approach [SC11] which is based on Self-Organizing Maps. This approach is able to predict the authorship attribution of a document and the genre of a web page based on style features. As well, it analyses the style of academic writing. The intention is that authors of papers can obtain feedback from the system in order to improve their writing style.

2.3.5 Topic Detection and Tracking

One important question of topic detection and tracking is the choice of a representation for a topic model. Some approaches are limited to keywords [WZHS07, LK08, TYZ09, ZWW10] or entities such as persons [KMS+10]. Another approach follows citations of reported speech [LBK09], because these quotations can be more easily followed. So far, the combinations of these topics' representations can be found rarely [LLLW05], because this implies a preprocessing of Information Extraction. IE requires some computation time and especially Topic Tracking can depend on fast processing. Applications in this area visualize frequencies of entities [KMS+10] (mainly persons and organisations) and discover sources of news. Topic Tracking is also part of the ATOM project and we have also performed experiments to this issue [Sch11]. Our approach models topics as a collection of entities (persons, organisations, places like cities or countries). The results indicate that Topic Tracking should be handled as an independent task. Thus, we still concentrate on Opinion Mining and do not discuss Topic Tracking in detail.

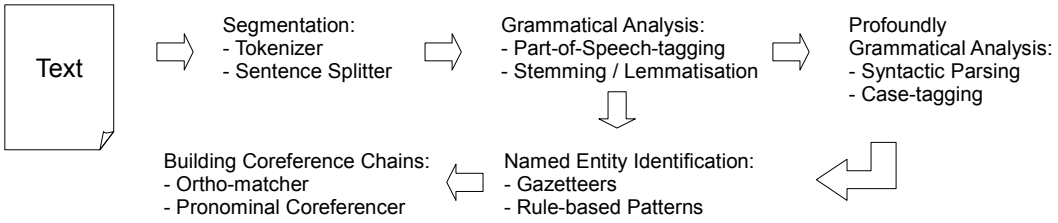

Figure 2.3: Overview about the hierarchical execution of NLP.

2.4 Textual Preprocessing

2.4.1 Natural Language Processing (NLP)

The application of different techniques of Natural Language Processing (NLP) can support Text Mining purposes in many ways. As a consequence, Text Mining starts very often with NLP as a preprocessing step. In this section, we clarify the most important NLP preprocessing steps for our purposes. NLP can be understood as a basic tool which makes text in natural language more comprehensible for a computer system. On the one hand NLP ensures that the text becomes more structured through segmentation and on the other hand additional information is extracted. The border between NLP, NER (Named Entity Recognition, explained later), and coreference resolution (also explained later) is not clearly defined in literature. Sometimes NER and coreference resolution belong to Information Extraction [CMBT02] and sometimes it belongs to NLP, but the frontiers are fluent like in this chapter. In the following, we sketch a typical NLP process from bottom to top, because NLP works typically in a hierarchical manner (cf. figure 2.3).

First, a stronger structure of the text is obtained by dividing the text into sentences and words by using so-called sentence splitter and tokenizer. After that, single words are tagged by their Part-Of-Speech. This means that every word is assigned to a word category such as noun, verb, adjective, adverb, and so on. The word gets a tag, which is an abbreviation for the category (NN for normal noun, ADJA for an attributive adjective, or ADV for an adverb [STT95]). These tags are specified in tagsets such as the Stuttgart Tübingen Tagset (STTS) for German [STT95]. Very well-known examples for Part-Of-Speech-tagger (POS-tagger) are the TreeTagger [Sch94, Sch95] or the Stanford Tagger [TKMS03]. As not otherwise stated, we use the TreeTagger for our experiments, because its results have a high accuracy and it can tag many different languages such as English, German, French, and many more [Sch94, Sch95]. Powerful POS-taggers such as the TreeTagger provide also additional information such as the actual conjugation of a verb and identify nouns which are a proper name. Furthermore, the words are often stemmed or lead back to lemmas (the TreeTagger is also able to

do that). This is especially important for languages such as German which have more inflected forms than English, for example. Stemming or lemmatisation, respectively, can simplify the process of many approaches. Not every form of a word must have an own entry in dictionaries for instance.

In addition, the opportunity exists to parse text syntactically. The Stanford Parser [RM08] can perform syntactic parsing. Syntactic parsing means that constituent structures (phrase structures) or dependency structures, respectively, of the words are identified within sentences [Niv10]. As a result, phrases or dependencies, respectively, are shown in a tree representation [Niv10]. We discuss syntactic parsing in more detail in section 7.1.1, when we need this technique for another approach. For our own approaches, we avoid syntactic parsing, because we believe that it provides not much benefit for our tasks. We verify this assumption in section 7.3 and section 8.4.2, when syntactic parsing is applied by two state-of-the-art approaches for the tonality classification and the viewpoint determination, but it does not lead to high accuracies. Moreover, it is very time consuming. According to own measurements, syntactic parsing can double the time for preprocessing approximately.

Now, we come to Named Entity Recognition and coreference resolution, which we like to explain with our Information Extraction module. In contrast to the previously explained segmentation, POS-tagging, and so on, we have to implement own solutions or expanded existing solutions for these issues, because existing tools/components are incomplete or show weak results.

2.4.2 Our Information Extraction Module

After the POS-tagging, a so-called Named Entity Recognition (NER) identifies persons, organisations, places, products, and so on. This is especially important for the extraction of products and brands in customer reviews and even more important for the identification of persons and organisations in news items. These entities can be grouped together in coreference chains, if the same entity appears several times in one text. For this coreference resolution, an ortho-matching module is needed among other things, because the textual representation of an entity can change (Deutsche Bahn AG or DB belongs to Deutsche Bahn, e.g.). The most ortho-matchers use heuristic rules for that. By using a pronominal coreferencer, persons and organization can be resolved, if they are only mentioned by he, she, it, him, his, her, and so on. For example, GATE (General Architecture for Text Engineering)[1] provides processing resources for NER in different languages. So, we use GATE as a basic framework for our approach. We also apply the GATE sentence splitter and tokenizer for the segmentation of texts in sentences and words.

[1]GATE: http://gate.ac.uk/

For our NER sub-module, we integrate several lists in the gazetteer. A gazetteer is a GATE component, which holds lists such as lists of forenames to identify named entities or parts of a named entity. Two general lists contain the most commonly occurring forenames (one for male and one for female names) and one list contains the most frequently occurring surnames in Germany. Each list for the forenames contains the top one thousand names of a telephone directory from 5th January 2005, while the entries of the surnames list have to appear at least one thousand times[2]. 3,422 surnames fulfil this condition. But we remove 22 surnames which are also very frequently used nouns (such as 'Abend' (evening)). In addition, we create an interface to add a list of important entities to improve our NER. This secures that these entities are found in any case. For this purpose, we have designed new JAPE Rules[3] which handles the string of the listed entity with the highest priority.

For our coreference resolution, we have improved the ortho-matching module by editing some rules because it sometimes creates problems with the identification of abbreviations in the names of organisations. And finally, we added a pronominal coreferencer for German in our project. So the approach can follow the mention of the person/organisation/product even when the person/organisation/product is only mentioned with he, she, it, and so on. This task requires a gender information which is created in the NER process. As a consequence, the list of important entities is divided into three parts: female person, male person, and neuter (organisations, products). For the other entities, the gender information is obtained through the lists as stated above.

We want to illustrate the extraction of coreference chains with a news article depicted in figure 2.4. Our coreference resolution creates the coreference chain for the entity 'Angela Merkel', because she is one of the persons mentioned several times in the article. But she is only mentioned two times by her full name in the article as 'Angela Merkel' and 'CDU leader Angela Merkel'. Our NER process should have recognized that 'CDU leader' is somekind of profession, and in the best case that 'Angela' is a given name and 'Merkel' is a surname. However, our process can identify that 'Angela Merkel' and 'CDU leader Angela Merkel' refers to the same entity by using pattern-based rules of the ortho-matcher. This concrete rule matches a forename-surname combination with the same forename-surname combination plus prefixed role (here 'CDU leader').

Through further ortho-matching, our process would add the four mentions 'Merkel' in our chain, because another rule of our ortho-matching module matches person entities, if they share the same surname. There are also exceptions for persons with the same forename or surname, but here these rules are just presented as examples, because

[2]lists by courtesy of Wiktionary: http://de.wiktionary.org/wiki/Wiktionary:Deutsch

[3]Developing Language Processing Components with GATE Version 6 (a User Guide): http://gate.ac.uk/

Dealing with anti-euro party AfD: Merkel bawls out her critics

From Philipp Wittrock

Angela Merkel proves nerves. She would punish the Alternative for Germany (AfD) with disregard preferably, but three CDU fraction leaders demand an offensive debate on the new anti-Euro party. The Chancellor reacts furiously.

Berlin – CDU leader Angela Merkel is annoyed about the internal criticism, which went public, of the party leadership's dealing with the new party: Alternative for Germany (AfD). In the sessions of the CDU management bodies on Monday, according to details provided by participants Merkel resents a remonstrative letter, in which the three leaders of state parliamentary groups Christean Wagner (Hesse), Mike Mohring (Thuringia), and Steffen Flath (Saxony) dun for a more offensively substantive dialogue with the anti-Euro party. The SPIEGEL reported on this paper at the weekend.

According to participants, Wagner, Mohring, and Flath mainly garner the wrath of the CDU chairwoman, because she finds out about the criticism from the SPIEGEL. Only last weekend, she sat together with the authors of the paper on a conference of the fraction leaders of CDU and CSU in Dresden, Merkel complained according to this. But nobody wished to speak critically about the handling with the AfD there. "We are already both over 18", the Chancellor is supposed to have said directly to the address of Mohring at the management board. The assistant party leader Armin Laschet and Julia Klöckner are alleged to support Merkel in her criticism, as well as CDU Euro-politician Elmar Brok.

[...]

Figure 2.4: This snippet is a translated SPIEGEL ONLINE article from 13th May 2013.

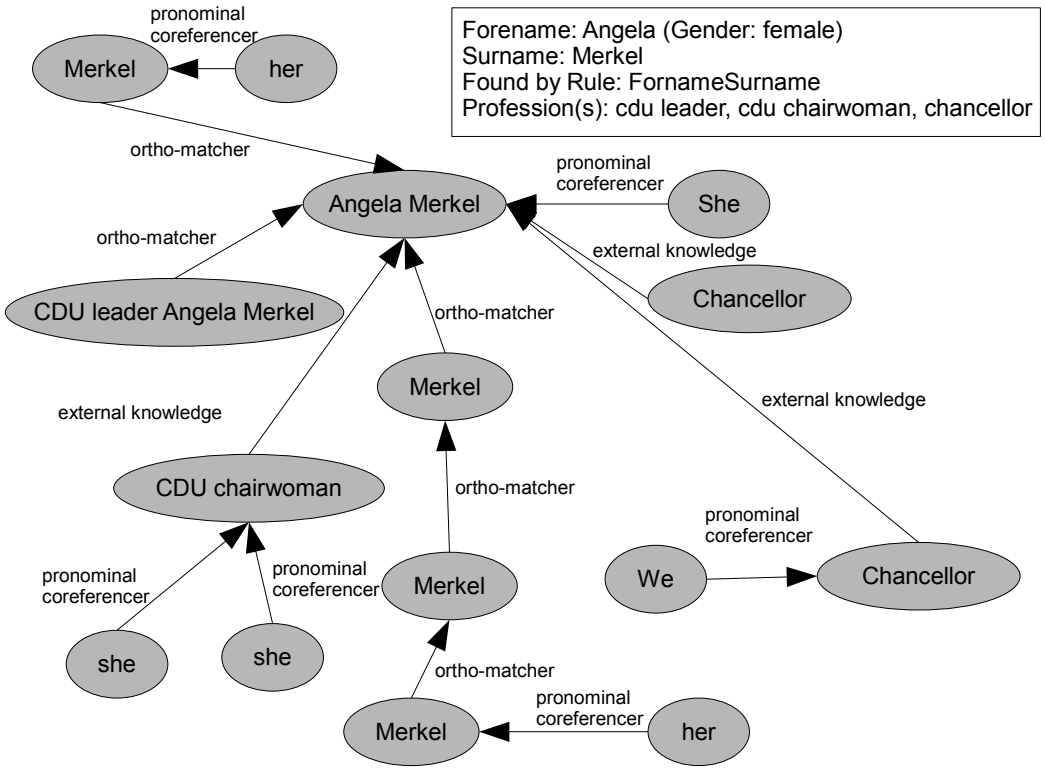

Figure 2.5: The resulting coreference chain of the entity 'Angela Merkel' with roles 'CDU leader', 'CDU chairwoman', and 'Chancellor' (external knowledge).

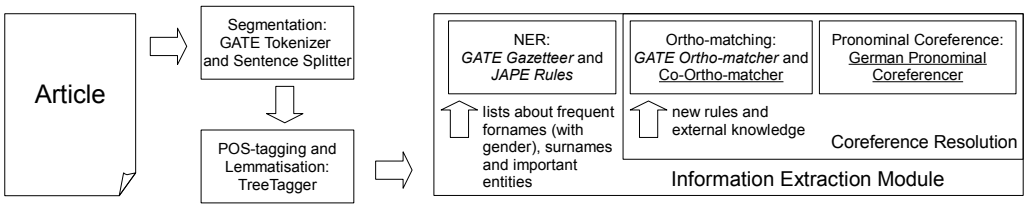

Figure 2.6: Our NLP and IE pipeline: Improved components are written in italics, while own components are underlined.

the explanation or even the mention of all rules would exceed the scope of our work.

For adding the pronouns into the chain, a pronominal coreferencer is needed. Our German pronominal coreferencer connects the three pronouns 'she' to the entity 'Angela Merkel', because all three are related to an element of the 'Angela Merkel' chain. Our pronominal coreferencer looks back three sentences and identifies the correct entity by comparing the gender. So, the third 'she' is more difficult to identify. In the same way, the pronominal coreferencer finds the two possessive pronouns 'her' and the pronoun 'we'. The 'we' is also a special case, because here the number is changed from singular to plural and it appears in a quotation. Quotations require additional rules, because a 'he' or 'she' in a quotation cannot be matched with the speaker, but 'I' or 'we' should be matched with the speaker.

The matching of references which only mention persons or organisations without any parts of the name or pronouns, but with roles such as a profession or position ('Chancellor', 'CDU chairwoman'), is more difficult and cannot be solved through ortho-matching, if the role is not mentioned during the chain with an identified element. This is not the case in our example text, because it does not contain a hypothetical element such as 'Chancellor Merkel' or 'chairwoman Angela Merkel', for instance. So, these matches cannot be derived directly from the text and require external knowledge. As a result, we introduce a so-called co-ortho-matcher in our NLP and IE pipeline (cf. figure 2.6). The co-ortho-matcher gets additional information about entities and their roles, and is able to match these kinds of references through rules in this way.

This is especially important for prominent entities. Many readers know that Angela Merkel is the Chancellor of the Federal Republic of Germany. Uninformed readers can derive this information by reading more texts and learn these facts from other co-occurrences. At the same way, an automated approach can learn and collect this information. However, we have noticed during our research project that this is more necessary for Information Extraction tasks than for Opinion Mining and represents a big effort in terms of computation time and in particular manual maintenance. Thus, we do not develop our own ideas in this direction.

Figure 2.5 shows the resulting coreference chain of the entity 'Angela Merkel'. In

our research, we try always to retrieve the best name of a person and consecutively label the chain after this name. The chain of this article would get the name 'Angela Merkel', because it represents the regular name with forename and surname of the person. The roles such as the professions 'Chancellor' and 'chairwoman' would be stored as additional information about the chain. The task of getting a good name for a person will be taken up in our viewpoint determination algorithm and partially in our statements extraction.

3

Opinion Mining

In this chapter, we introduce Opinion Mining: We explain several aspects of it and the application for different domains. Opinion Mining (also known as Sentiment Analysis) tries to identify and rates opinions in text. We will deal with Opinion Mining in three different domains: reviews, entries out from social media, and articles in news. Furthermore, this chapter elaborates on different subtasks: Subjectivity Analysis, the extraction of related information on opinions (such as viewpoints, opinion holders), the identification of opinion-bearing text, and the influence of modifiers. Then, we will focus on available resources (datasets and dictionaries) for Opinion Mining.

Opinion Mining and Sentiment Analysis imply the identification and classification of tonality-bearing texts, as well as the extraction of belonging information about or within these texts. In the simplest case, the classification of the tonality-bearing text is the distinction between positive or negative text, but we also present approaches which also can handle neutral examples or even a multi-point scale. Opinion Mining is a branch of Text Mining, thus the proposed techniques share many commonalities with techniques from the areas of Knowledge Discovery and Data Mining, Information Retrieval, and even Information Extraction, as we already saw in last chapter. In addition, many processes in Opinion Mining are based on Natural Language Processing (NLP). This chapter investigates several approaches in different domains, various tasks, and aspects of Opinion Mining and important resources.

And finally, we explain our requirements on Opinion Mining in newspaper articles for Media Response Analysis in order to show the differences between the related work and our research. In this way, we deepen the motivation for our research even further.

Word	positive score	negative score	Word	positive score	negative score
good	0.625	0	great	0.25	0.125
bad	0	0.625	awfully	0	0.75
crisis	0	0	solve	0	0
winner	0.25	0	slowly	0	0
spartan	0.5	0.25	criticize	0	0.25

Table 3.1: Examples from the SentiWordNet 3.0 [BES10]. Some of the examples have several entries, so we try to show the most context-independent one.

3.1 Opinion Mining in Customer Reviews

Research to date has tended to focus on mining and analysing opinions in customer reviews rather than in newspaper articles [PL08]. Approaches for Opinion Mining deal with reviews from the beginning and this domain is still a very important (maybe the most important) area of operations. The explanation is simple: There exists many more test data and training data for this area than for newspapers or contributions in social media. Annotated datasets need to be created for Opinion Mining in newspaper articles or social media, whereas online stores such as amazon.com provide numerous reviews on their websites, which can be accessed publicly and the allocated stars or points form a sentiment annotation.

Opinion Mining in customer reviews can be used for summarizing opinions [HL04, PLV02], for the detection of incorrect or faked reviews [JL08], or for a detailed analysis of the sentiment about different features of one product [BES09], because different product features can be differently relevant for various customers. While summarizing opinions is not so interesting in our opinion (cf. chapter 1.1), opinion spam detection and analysis of product features are more interesting and thus they maybe receive more attention in recent research. Reviews with untruthful opinions (fake reviews, bogus reviews), reviews on brands only, and reviews, which are advertisements or contain only questions, answers, or irrelevant text, belong to opinion spam [JL08].

One early study [PLV02] from Pang et al. applies three machine learning methods (Naive Bayes, Maximum Entropy Classification, SVMs) to rate film reviews as positive or negative. A more comprehensive study by Pang and Lee expands the problem to a multi-point scale [PL04]. This study evaluates also human performance on this task.

Many approaches have focused on the creation of sentiment dictionaries. A sentiment dictionary does usually contain words with a score, which shows how positive or negative the word is in general. Table 3.1 show some examples of the SentiWordNet [BES10]. Kaji and Kitsuregawa use chi-square and pointwise mutual information based

values (PMI) [Tur02] to select polar words and phrases from a massive collection of HTML documents [KK07].

The following approaches construct dictionaries from multi-domain sources. Du et al. [DTCY10] use the information bottleneck method to construct sentiment lexicon based on one domain for a new domain. Here, out-of-domain dictionaries are constructed by in-domain knowledge [DTCY10]. Besides, domain independent sources such as a general sentiment lexicon and a dictionary for synonyms and antonyms can be combined with domain-specific reviews for a context-dependent sentiment lexicon [LCDZ11]. Bollegala et al. construct a thesaurus for Opinion Mining [BWC11], which can be applied for cross-domain sentiment classification. By using labeled and unlabeled data, the thesaurus finds terms which express the same semantic orientation, although they are from different domains.

One major aspect of Opinion Mining in product reviews is the collection of sentiment bearing words [HPD+08, KK07] or the construction of complex patterns which represent not only the sentiment, but also extract the relationship between the sentiment and the features of the products [KIM+04]. In a similar manner, Association Rule Mining can be applied to customer reviews [KRKK09]. They collect opinions about products by four types of rules (product → opinion, product → feature, feature → opinion, product → [feature → opinion]).

Of course, the extraction of product features can be understood as an own subtask in this domain. With the extraction of all relevant aspects, a more detailed rating is possible [BES09]. The discovery of products can be performed by using patterns [DLZ09] and the correct assignment between products/features is done by an algorithm [DLZ09] which also concerns comparative and superlative sentences and estimates the orientation of an opinion. As well, the information bottleneck is adapted [DT09] for this problem. Or a two-way learning is possible: The proposed algorithm [QLBC11] learns new opinion words through known targets and new targets by known opinion words. Here, targets can be product features or topics.

Another technique called **Opinion Observer** [DLY08] is a very detailed algorithm, which aggregates opinions for product features. Ding et al. [DLY08] start with a basic lexicon. This dictionary includes opinion words (positive, neutral, and negative words) and idioms, too. Ding et al. [DLY08] propose many different rules to identify the sentiment orientation of words. These rules are described in algorithms, which can handle negations, inter-sentence conjunctions, but-clauses, the modifier “too”, and other contextual influences. Their Opinion Orientation algorithm [DLY08] implies the extraction of relations between opinion words and corresponding product features. Thereby, Opinion Observer generates a detailed analysis of product reviews. We will provide a comprehensive description of Opinion Observer in section 7.1.2, when Opinion Observer is applied for the tonality classification.

Sarvabhotla et al. [SPV11] extract the subjective excerpt of a text by a technique, which they call consequently Review Summary (**RSUMM**). RSUMM builds two word-vectors: The most important and most characteristic words for a domain are listed in the average document frequency vector. Subsequently, the most subjective terms are selected for the average subjective measure vector. An SVM (SVM-Light [Joa99]) finally classifies the reviews based on these well selected word features. RSUMM requires hardly any natural language preprocessing. It needs only a sentence splitter and a tokenizer. RSUMM will be explained in detail in section 5.4 and 7.1.4. In section 5.4, we will extract statements with RSUMM, and we classify the tonality with RSUMM in section 7.1.4.

Taboada et al. [TBT+11] calculate the semantic orientation of opinion-bearing words (**SO-CAL**). SO-CAL works with a fine-grained lexicon of adjectives, adverbs, verbs, and nouns. These words have a score from -5 to +5. "Inspiration" has a score of +2 and "sham" of -3, for instance. Based on this sentiment dictionary, SO-CAL analyses intensifiers, negations, and irrealis [TBT+11] and thereby modifies the score of the words through rules and formulas. Moreover, text-level and other features have an impact on the final score. All these methods are explained in detail in section 7.1.3, when SO-CAL is also applied for the tonality classification. SO-CAL is listed in the section customer reviews, but it could also be mentioned in the section social media or newspapers, because Taboada et al. promise a solution which works across domains and in their evaluation they show good results in reviews, news articles, and contributions in social media [TBT+11].

Many techniques rely on a sentiment dictionary such as SentiWordNet [BES10]. In this work, we apply the SentiWS dictionary [RQH10], which we explain in the section about resources (3.5).

3.2 Opinion Mining in Social Media

Approaches in this area analyse tweets from twitter, entries from blogs, facebook, or comments to youtube videos. In our opinion, the analysis of opinion-bearing text in social media is just as interesting as Opinion Mining in newspaper articles, because detailed overview systems for the sentiment do not exist for social media, too. Although partially sentiment scores can be calculated directly by some features like on youtube (likes and dislikes) without analysing the textual content of posts, and thus, social media open up some more opportunities than in newspaper articles, overall sentiments can be estimated in customer reviews much better. Again, probably the huge amount of available training data for reviews is one reason, why much less research had been done even in social media. However, results of the analysis of data from social media

could be inserted into a Media Response Analysis. Nevertheless, it has been shown that Opinion Mining in social media is very different in contrast to Opinion Mining in newspaper articles and require different techniques to handle characteristics such as slang, for example. Furthermore, sarcasm and irony are a strong limitation in Sentiment Analysis in social media [TBP12], while we found out that this is not a big problem in newspaper articles.

On the other hand, approaches also profit from other characteristics such emoticons (smilies) [HBF+13]. Pak and Paroubek [PP10] classify entries of Twitter with emoticons and study distinctive characteristics (such as frequent POS-tags). Their findings are particularly interesting: Personal pronouns appear more often in a subjective tweet and verbs in past tense are a small hint for a negative tonality.

Harb et al. [HPD+08] analyse blog-entries to extract negative and positive opinions. They use Association Rules in order to extract relevant adjectives for this purpose.

One state-of-the-art approach is SentiStrength [TBP12]. SentiStrength (version 2) [TBP12] calculates a positive and negative value for texts at the same time. This technique includes a spelling correction algorithm, special word lists and adjustments (such as repeated letters or repeated punctuation). They achieve results as good as human annotators, if the texts do not contain sarcasm or irony.

The study of O'Connor et al. [OBRS10] investigates the question, if Sentiment Analysis in twitter may substitute or at least supplement opinion polls. They use traditional polls such as Obama vs. McCain or Obama's job approval as gold standards and apply the lexicon from Opinion Finder [WWH05] for their Sentiment Analysis step. Finally, the authors answer the question cautiously with yes, because their findings replicates the polls. A similar study [BMZ11] uses twitter to analyse the public mood. The results show their correlation with the stock market (Dow Jones) and that it is even possible to predict the stock market's behaviour by analysing the public mood via twitter.

Even in this domain, the connection between Opinion Mining and (Multimedia) Information Retrieval can be shown: The analysis of sentiments in social media can also lead to recognition of images' sentiments [SMDH10].

3.3 Opinion Mining in Newspaper Articles

The research to date has tended to focus on Opinion Mining in customer reviews and to a limited extent in social media. Far too little attention has been paid to Opinion Mining in newspaper articles. And, in our opinion, solving the problem for newspaper is not achieved with methods or outcomes of research concerning reviews, because these methods are rarely evaluated for newspaper articles and if so they restrict the

evaluation's scope (only positive and negative examples [TBT+11], e.g.). Nevertheless, approaches for the news domain can be found in scientific literature.

One of the first approaches [SGS02] for Opinion Mining in newspaper applies machine learning to classify financial news, which are annotated by domain experts. The whole news articles are rated as *good*, *good uncertain*, *neutral*, *bad uncertain*, or *bad* based on the description of the financial situation of a company. Another early approach [KS06b] correlates news stories about stocks to the performance of the stock price. They classify short news stories as good news or bad news and use the performance of stock price during two days in two different time windows for annotating a training and test set. Also, Devitt and Ahmad [DA07] analyse financial news and their impact on the financial market. They use SentiWordNet and WordNet 2.0 for the analysis of news texts about a company takeover. But they consider only positive and negative news.

More recent approaches [BSG+09, BSK+10, PLS11] focus on reported speech objects. There are several reasons for this: As we have previously shown, different parts of newspaper articles can contain different opinions. So, Sentiment Analysis does not make sense for complete articles. Thus, an approach has to identify firstly text parts which contain an opinion. For an MRA, statements have to be extracted. A second reason is the fact that news articles are less subjective [BSG+09] than customer reviews for example and quotations are more subjective than other parts in news [BSG+09]. In addition, it is easy to identify reported speech, even in different languages [PSB07]. During this process, also additional information such as the opinion holder (the speaker) can be extracted. But these approaches are very limited, because many valuable opinions get lost, as we mentioned briefly in section 1.1 and we will verify this in section 6.2.

Wilson et al. [WWH09] introduced an approach for the classification of words. This method (denoted as **Wilson**) analyses the contextual polarity. The approach generates word features and sentence features by using a dictionary and POS-tags. This technique also performs deep natural language analyses with dependency parse trees. As a result, the method obtains (general and polarity) modification features and structure features. Finally, the approach calculates 32 features for neutral-polar classification and 10 features for the polarity classification. These features can be applied by machine learning. Wilson et al. [WWH09] evaluate different kinds of machine learning techniques such as Ripper [Coh96] or BoosTexter [SS00]. We explain this method in detail in section 7.1.1, because Wilson is one of our state-of-the-art comparison methods for the tonality classification.

3.4 Different Tasks and Aspects of Opinion Mining

Many approaches tackle different subtasks of Opinion Mining alongside concrete techniques for certain domains. And most of these approaches cannot be correctly assigned to a concrete domain. Findings in Subjectivity Analysis are interesting for several kinds of Opinion Mining tasks, for example.

3.4.1 Subjectivity Analysis

Subjectivity Analysis is the task of differentiating between subjective and objective words/sentences/texts. To create subjective extracts, Pang and Lee create a graph from the sentences of reviews and compute a minimum cut to obtain only subjective sentences [PL04]. They validate their approach on film reviews. The objective sentences belong to plot summaries, which are given in reviews, whereas the subjective sentences are these parts of the review, which express the opinion about the film.

Two other approaches determine word senses as subjective or objective: For the first one [GWMA09], overlaps of relationships in WordNet, word vectors, a sense, and a domain score are estimated as features. They use machine learning (SVM Light [Joa99]) and senses extracted from a subjective lexicon [WR05] and WordNet. The second study [SM09] uses unigrams and POS-tags as features. They use the LIBSVM (a more recent description of this library can be found in [CL11]) as baseline and propose semi-supervised mincuts as a better classification techniques. They construct a graph, in which its edges represent the similarity between the vertices. Vertices represent the word senses. In this way, similar items should be grouped together by the minimum cuts.

From a more theoretical perspective, Koppel and Schler show [KS06a] that neutral examples are not something in between positive and negative texts. Many approaches made this assumption. Neutral examples should be treated as a separate category. If this is done, then it can increase the distinction of positive and negative examples, too.

3.4.2 Negations, Irony, Conjunction, and Co: Modifiers of the Polarity

Opinion Mining always deals with text and therefore with language. Although this thesis does not originate from a linguistic or computational linguistic background, analyses of texts can benefit from the consideration of linguistic influences.

Contextual valence shifters such as negations, intensifiers, and irony are treated by Polanyi and Zaenen in a theoretical paper [PZ06]. This paper influences a lot of other contributions such as the state-of-the-art methods SO-CAL [TBT+11] and Wilson [WWH09].

Irony, which is treated by Utsumi in a theoretical paper [Uts96], assumes relevant importance in the work of Carvalho et al. [CSSdO09] which explores characteristics such as emoticons, onomatopoeic phrases, punctuation/quotation marks, and interjection in positive comments. Irony is a great challenge for Opinion Mining in social media [TBP12], while it seems that journalists do not often formulate their articles in an ironic way in news.

Certainly the best investigated modifiers are negations. Whole papers deal with negations for Sentiment Analysis: A survey of Wiegand et al. [WBR+10] presents various approaches for the detection, the scope, and representation of negations. One contribution [JYM09] proposes techniques for the effect and scope of a negation. These ideas will be picked up later in this thesis.

Also, conjunctions can provide information about the sentiment, because conjunctions can express a support or a contrast [SC13b]. One of the first approaches in Opinion Mining bases on this idea [HM97]. New opinion words with the same polarity are learned by the conjunction 'and', new words with the opposite polarity are learned by the conjunction 'but'. Zhou et al. [ZLG+11] use conjunctions in order to create relations (contrasts, conditions, continuations, causes, and purposes) which can be used as features for opinion-bearing texts.

3.4.3 Analysis of Different Points of View

Another subtask is the analysis of different perspectives for Opinion Mining. This task is especially important for our solution, whereas different perspectives are not so substantive for Opinion Mining in customer reviews, because only one person expresses generally his/her own opinion in a review.

For the news domain, Park et al. [PLS11] extract two groups of people who share the same opinion about a topic and have a contrary opinion to the other group. A technique called OPUS (observable proxies for underlying semantics) [GR09] identifies the perspective in opinionated texts by adding syntactic information to words. We will refer to OPUS in section 8.2.1 and analyse, if OPUS is able to perform a viewpoint determination for an MRA.

A graph-based approach [TPL06] investigates records of debates to determine the speaker's agreement (support or opposition) about proposals, while a lexicon-based approach [SW10] recognize attitudes in online debates (such as healthcare, gun rights, or abortion). DASA [QHZ+10] (Dissatisfaction-oriented Advertising based on Sentiment Analysis) extracts topic words and related opinion-bearing words through syntactic parsing. The system recommends products of competitors, if a product is connected to negative sentiment. DASA will be described exhaustively in section 8.2.2, in which DASA extracts viewpoints in an MRA.

3.4.4 Extraction of Opinion-bearing Text, Opinion Holders, and Opinion Retrieval

The extraction of opinion-bearing text differs from Subjectivity Analysis. While in Subjectivity Analysis the text will be rated as subjective (positive,negative) or neutral, text parts which can be rated as positive, negative, or neutral can be separated from text parts which contain no opinion (not relevant). However, this task depends on the actual use case, but it is essential for an MRA, for example. An early contribution [KH06] identifies opinions by means of dictionaries containing opinion-bearing words. In combination with the identified opinions, topics and opinion holders are estimated.

Opinion holders represent persons, who express an opinion. Wiegand and Klakow [WK10] apply different kinds of kernels to extract opinion holders in the MPQA corpus. Whereas Jakob and Gurevych [JG10] use conditional random fields to extract opinion targets.

In order to extract different text types of reviews, Taboada et al. apply SVMs, Naive Bayes, and Linear Regression Classifiers to qualify text parts of film reviews as comment, description, or formal [TBS09]. Also, the earlier mentioned RSUMM proposed by Sarvabhotla et al. [SPV11] extracts the opinion-bearing text. They calculate two scores, which show how important one sentence is. Film reviews are their use case, too. But their methods are very domain-independent and language-independent, thus we apply RSUMM for the statements extraction in section 5.4.

For retrieval purposes, Huang and Croft introduce a Language Model for Opinion Retrieval in which words are divided into topic words and opinion words [HC09]. They also propose a method based on relevance feedback to extract opinion words; text corpora and queries are combined for this technique. They evaluate their model on the Blog06 corpus, for example (we explain this corpus later in this chapter).

3.4.5 Emotion Analysis

So far, Sentiment Analysis and Opinion Mining differentiate between positive, neutral, and negative texts (or text parts). But some approaches understand sentiment in a much broader sense. These approaches try to identify emotions in texts such as Joy, Fear, or Anger.

Feng et al. [FWY+11] clusters sentiments based on a Probabilistic Latent Semantic Analysis (PLSA). They analyse blogs in Chinese. With this technique, they show that they can create clusters with emotional words. One cluster contains the emotional words for regret and another cluster represents the emotional state of disappointment.

In news headlines, Strapparava and Mihalcea [SM08] identify six emotions: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise*. They propose knowledge-based and corpus-

based methods to tackle this problem. Their knowledge-based solutions require additional resources such as dictionaries and/or apply Latent Semantic Analysis for a semantic similarity. The corpus-based solution can be described shortly as training a Naives Bayes classifier on a corpus of blog entries. The findings of their experiments suggest that their knowledge-based solutions perform better for this task.

3.4.6 Topic Models for Sentiment Analysis

Sentiments often are concerned with topics. Even our research project deals with topic detection and tracking. Thus, special topic models have been introduced to model their relations.

In customer reviews, Titov and McDonald [TM08] use topic models to extract features/aspects of products. For reviews about Italian restaurants, they maybe identify topics such as *wine*, *pizza*, *pasta*, *location*, *service*, *value*, or *atmosphere*, for example. As mentioned above, in this way a very detailed Sentiment Analysis on different aspects is possible and users can compare opinions about their favourite aspect(s).

Fang et al. [FSSY12] handle the problem of contrastive opinion modeling. Through their topic models, they can extract opinion words. Since this happens under different viewpoints, perspectives are obtained.

The combination of topic models and sentiment can be used to mine meanings. In political contexts, an approach [SZ12] guesses the party of ministers based on party programmes and coalition agreements.

3.5 Resources for Opinion Mining

Besides the general resources for NLP and Text Mining tasks, some datasets and dictionaries are generated especially for Opinion Mining and Sentiment Analysis.

3.5.1 Datasets and Corpora

We begin with datasets in this research area. Datasets are published for different purposes. Hu and Liu work in their paper [HL04] about Opinion Mining in customer reviews with a dataset of 322 reviews. It contains five categories of products. This dataset was increased for a later publication [DLY08] to 445 reviews of 8 different products (the publication speaks about 8 different products, while the website offers a dataset with 9 different products). In the same year, Jindal and Liu published a dataset with more than 5.8 million reviews for opinion spam detection [JL08].

For social media, a corpus of 500 short messages [Mom12] is available for Opinion Mining. It contains texts from Facebook, Amazon, Youtube, blogs, forums and fan

pages, which talk about celebrities, especially singers and musicians. The texts are annotated by three persons with two scores: A positive score from 0 to +3 and a negative score from 0 to -3. The Blog06 [MO06] corpus is a test collection of blog posts. Besides the classes positive and negative opinions, it contains a class with the mixture of both polarities, but it does not contain neutral opinions.

For Opinion Mining in newspaper articles, the MPQA corpus [WWC05] is a very famous dataset. It contains word-based annotations for news texts in English. We discuss this dataset in detail in section 4.2. The NTCIR-6 [SEK+07] contains annotated sentences from Japanese, Chinese, and Korean newspapers in three languages: Chinese, Japanese, and English. The English part consists of 439 documents with 8,528 sentences. Their inter-annotator agreement is very low (only 29.47 averagely in case of opinionated text). The dataset is more concentrated on assigning opinion holders to the sentences, while viewpoints are missing in contrast to a corpus of a Media Response Analysis. Furthermore, 702 French sentences in a collection of Bestgen et al. [BFK04] contain named entities (persons or organisations) and are tagged as pleasant or unpleasant on a scale from -3 to +3 (seven graduations).

3.5.2 Sentiment Dictionaries

Many approaches such as [DLY08, WWH09, BSG+09, TBT+11, TBP12] rely on sentiment dictionaries. Probably the best-known dictionary is SentiWordNet [BES10] for English. The SentiWordNet is constructed by means of WordNet [Mil95], which is a general dictionary for English. In WordNet, terms or, more precisely, synsets are connected to other synsets by relations such as “see also” or “direct antonym”. Baccianella et al. [BES10] begin with a small number of seed terms (7 unambiguously positive and 7 unambiguously negative terms [TL03]). They use the WordNet relations to increase the number of labeled synsets. Having these synsets as training examples, they classify new synsets. By a random walk between these synsets on special relations in WordNet, they obtain an orientation score (polarity score).

The General Inquirer [SDSO66] is also very famous and often used for different subtasks. It contains labels for many words, which show their semantic orientation (positive, negative), as well as lists of words which express agreement or disagreement.

But also dictionaries in other languages are publicly available. The NTU sentiment dictionary [KLC06] provides positive and negative terms for Chinese. In addition, HowNet [DDH10] contains cross-lingual sentiment entries for words in Chinese and their equivalents in English. The SentiWS [RQH10] contains over 3,500 words in lemma for German. The sources of SentiWS are the General Inquirer, which is translated into German, a large collection of product reviews, and a special German dictionary. They compute and refine a sentiment score for their words based on the three sources by

applying the pointwise mutual information method [CH89] for the score. Hereby, they obtain 1,686 positive and 1,818 negative German words in lemma. We will apply the SentiWS for various experiments during this thesis.

3.6 Opinion Mining for a Media Response Analysis

Now that we have an exhaustive overview about Opinion Mining, we turn to Opinion Mining for a Media Response Analysis. Thereto, we want to motivate our research even more than in the first chapter. We give more information about the progress of a Media Response Analysis and contrive a top-level draft for the solution. The different parts and aspects of the motivation and the progress of an MRA have been published in our contributions [SCW12, SCH12, SC12, SC13a, SC13b, SC13c].

3.6.1 Motivation for Opinion Mining in this Area

With the growth of online portals of newspapers and news services, Opinion Mining in news has become a central issue for media monitoring. Opinion Mining in newspaper articles [MG05, WN07] presents a major challenge and a high benefit at the same time. Public relations departments of companies, organisations such as political parties, associations, institutes, or foundations and even distinguished public figures need an analysis of their public image, because they have to care for their public relation. Advertising, PR campaigns, or election campaigns require such analysis, so that they are able to analyse the success in PR activities or the media's image about themselves or their products. To obtain this information, they have to perform a Media Response Analysis (MRA) [MG05, WN07].

How is the media image about company XY? Is the sentiment changing after the last advertising campaign? How does the media talk about the new product of company XY? Is the tonality in the news changing after the speech of the chairman/chairwoman of the party? A Media Response Analysis (MRA) answers these questions [MG05, WN07]. Therefore, it represents an own business segment for media monitoring companies, but it means a big human effort. So, Opinion Mining is very interesting in the context of news articles because it would save much time and human effort/running costs.

One reason why this is urgently necessary is that the number of relevant articles is growing continuously [MB10, HF12]. Every day, many news texts are published and distributed over the internet (uploaded newspaper articles, news from online portals). As a consequence, these news articles are accessible in digital format. They contain, besides useful information, potentially valuable opinions. At the same time, analyses in more detail are requested and the clients want to obtain their results in real time,

because they want to react to dramatic changes in the polarity of opinions quickly, especially when the prevailing mood becomes a negative one. In a worst case scenario, the press spokesman/spokeswoman learns about a critical report on his/her company from waiting camera team which rings at the door.

So, the tasks are getting more and more difficult. As a consequence, media monitoring services require more machine-aided methods. Thus, a core issue for media monitoring is going to be Opinion Mining.

In an MRA, the most important key performance indicators are the media reach and the sentiment (the more concrete term is tonality in this context). The media reach shows how much the article is distributed in the media (how many sources like online portals, for example, publish this article). The calculation of the media reach is rather simple and automatically realised. An automated calculation of the tonality is much more difficult. This is one of the major challenges of our research.

3.6.2 Procedure of a Media Response Analysis

This section describes the procedure of an MRA briefly. A lot of things can be mentioned about MRAs, but we want to concentrate on issues which are relevant to our research in this area.

In order to create a report about the media attention, an initiator of an MRA has first to define key words of interest like the names of companies (the client's own company or organisation, subsidiary companies or organisations, competitors, etc.), products, or important persons (chairmen/chairwomen, press agents, advertising media, opinion leaders for this domain, etc.). Thereby, all news items containing these terms are automatically collected by a crawler system.

But it still requires a big human effort for media analysts to read the articles. One media analyst has to analyse approximately between 200 and 800 articles each week. Then the analysts have to select relevant statements from these articles. A statement is a consecutive sequence of sentences which are relevant for one or more analysis objects. In the next step, the analysts have to set the tonality of the statement. In most tasks, they code the statement for a certain group. This represents a viewpoint for the tonality.

A positive statement for one viewpoint might not have a tonality (is irrelevant) or might be negative or neutral for another viewpoint. Thus, a statement can have two or more different viewpoints with (not necessarily) different tonalities. But for one certain viewpoint, the tonality does not change within one statement. The viewpoints represent generally a group of analysis objects, although some analysis objects do not belong to any perspective (opinion leaders, e.g.).

We want to illustrate viewpoints by means of an example, which was published

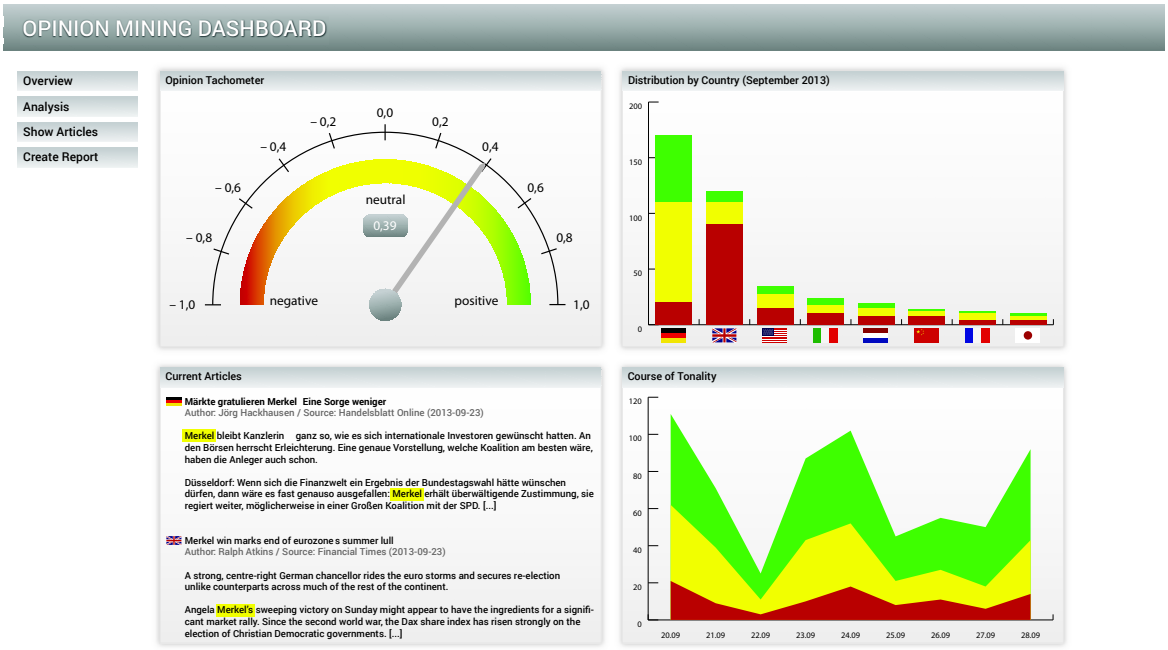

Figure 3.1: A typical MRA overview system (adapted from [PK13]).

in our paper about our algorithm for viewpoints [SC12]. The following statement is positive for US president Barack Obama and the Democratic Party, e.g.:

- President Obama made the tough, politically unpopular decision to rescue an industry in crisis, a decision that saved more than 1.4 million American jobs. (Code: positive, Democrats)

This statement is not positive for the viewpoint of the Republican Party (the competitors of the US Democrats), but it is also not negative or neutral, as one might have expected. The statement is irrelevant for the viewpoint of the Republicans, because there is not any mention of the Republican Party or a member of them. Of course, humans can interpret a statement and a positive statement for one viewpoint can be a negative statement for a competitive viewpoint. Especially in politics, many statements can be interpreted in that way. But this is not applicable for all domains. The statement "VW generated recorded sales. The people buy more cars once again." can imply also a positive statement for competitors. However, we will investigate this question in detail in chapter 8, when we will discuss the automated assignments of viewpoints.

To put this into practice today, a big human effort is needed. At the same time, the tasks are getting more and more difficult. The number of potentially relevant articles is increasing and the clients want to obtain their results in real time, because they have to be able to react quickly to dramatic changes in tonality. As a consequence, more machine-aided methods are needed for the field of Opinion Mining in newspaper

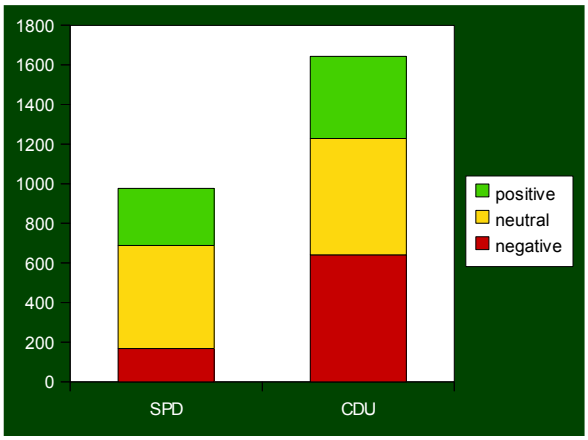

Figure 3.2: Analysis results are presented as bars.

articles.

The results of an MRA can be presented like in figure 3.1 or in figure 3.2. Figure 3.1 shows an overview system for the results of an MRA. The central tachometer shows the average tonality for example. Also, time series for the tonality and results of special queries are depicted. With all the meta information of the Web crawler system, detailed queries are possible, which concern a certain period of time, a certain country, certain companies (viewpoints), or selected media such as designated newspapers, for example. Also, another typical presentation of the analysis result is the distribution of the tonality for the viewpoints. Figure 3.2 shows the distribution for two viewpoints. PR people can analyse their work's success with these results and every PR department needs another style of presentation.

The results can be enriched with other business information such as the media equivalent advertising value in order to show the importance of the news article. The value is normally given in monetary unit, and an article in a national newspaper with a high print run is worth more than an article in a technical journal with relatively low circulation. So, the extracted and rated statements can be weighted according to the equivalent advertising value. These values are stored in a database and maintained from time to time.

But for the basic data, only a statements extraction, a tonality classification, and a viewpoint determination is required. And in this thesis, we propose a solution for all these three problems by following a top level solution strategy explained in the next section.

3.6.3 Solution Policy

We propose a divide and conquer policy for our Opinion Mining solution. First, statements should be extracted from news articles. Then the tonality should be identified

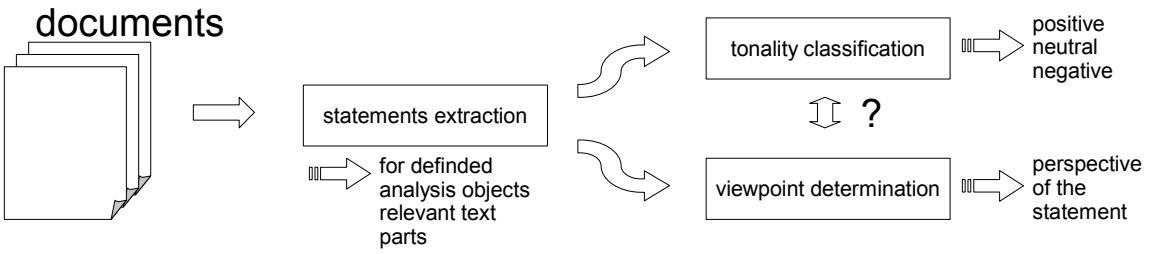

Figure 3.3: Top-level overview: Divide and conquer principle for our solution. One open question is the relationship between the tonality classification and the viewpoint determination.

in a second step, and also the viewpoint should be determined afterwards. It was not decided at the beginning, whether the determination should be influenced by the result of the tonality, but this open question should be answered during this research project.

We decided to apply a divide and conquer principle, because it can be understood easier with its partial results. Our research application intends a hybrid solution: The human analysts should be supported by a machine-based solution. So, an analyst should comprehend every step of the machine as well as possible. Also, a solution which can generate the partial results is interesting, because it is more flexible. In this way, it is also possible that a media analyst selects a statement and the system propose a tonality for this statement.

Also, just the extraction of relevant statements represents a great alleviation of work for the analysts. The application of the developed methods was not clearly defined at the beginning of the project, but it should be found out during the project. Besides, we also planned a completely automated solution, if the results are good enough. This analysis can be offered at a lower price, because it requires very few human effort, but it is expected that this solution produces worse results.

In the next chapter, we show the basic data of an MRA by presenting the datasets, especially the pressrelations dataset, which we will use for the evaluation. In this way, we demonstrate a concrete example of the MRA procedure. Then, we deal with every sub-task of our divide and conquer policy in the following chapters 5 (statements extraction), 6 and 7 (polarity and tonality classification), and 8 (viewpoint determination).

4

Evaluation Framework: The Publicly Available Corpus

In this chapter, we present our evaluation framework. Corpora and other resources were published for many different tasks in Opinion Mining. However, as far as we knew, there was no corpus publicly available for Opinion Mining in a Media Response Analysis at the beginning of the ATOM project. An available corpus for Opinion Mining in news was the MPQA corpus [WWC05], but this corpus is not designed as an MRA. It is annotated in a different way, which we explain in this chapter. So, we published an own corpus, the *pressrelations dataset* [SCH12], which has been created by two professional media analysts and contains 617 articles with 1,521 statements. We explain the human-generated-data of a Media Response Analysis by means of this dataset as a typical example. In addition, we also use another corpus for our experiments, the *Finance dataset* which is a real world dataset of a Media Response Analysis of a financial service provider and its four competitors. This dataset is also introduced shortly in this chapter.

4.1 Introduction

An MRA can be performed in different ways. In the simplest case, an article can be positive, neutral, or negative as a whole. In the more interesting case for a media analysis, it contains statements which are positive, neutral, or negative. In addition, the statements belong to a viewpoint, which shows that this statement has a certain tonality for that viewpoint. The following example is positive for the viewpoint SPD, which is the political party of the German Social Democrats:

(4.1) Therefore the SPD supports a quota of women's representation of the governing boards in companies. The SPD has just presented a draft bill in the Bundestag. Now the other parliamentary groups and the government have to show their colour. (**Code:** positive, SPD)

This and all following examples in this section are taken from the *pressrelations dataset* [SCH12]. An example statement for the governing party of the German Christian Democrats (CDU), which has a negative tonality, is this:

(4.2) The childcare supplement is a family-policy and women-policy disaster; even the women's union and the employers' association say that. It is another evidence of incapacity of the black-yellow government, that nevertheless Mrs. Schröder is going to rush through the bill. (**Code:** negative, CDU)

To illustrate how difficult the task is even for humans, here is a neutral example from the same topic:

(4.3) And already on Tuesday, a commission of experts for research and development (EFI), which was convened by Chancellor Angela Merkel, has argued in their opinion not to introduce the childcare supplement. (**Code:** neutral, CDU)

This example may be interpreted as a negative statement for the CDU, because the childcare supplement is a piece of proposed legislation, which belongs to the project of the CDU. But this statement is not direct criticism of the CDU, it is more an information and CDU could not introduce the childcare supplement based on this information. We will show later, how difficult this task is in our inter-annotators' agreement study (cf. section 4.4).

As previously mentioned, there is a lack of resource to design new automated approaches for Opinion Mining, especially in the newspaper context and for other languages than English (e.g. the MPQA corpus consists of English news articles). Moreover, the existing resources do not fulfil the requirements of Opinion Mining tasks for an MRA, because they do not include the concept of relevant statements nor a sentiment value for a single statement. Also, the tonality does not belong to a certain viewpoint.

Therefore, we introduce the *pressrelations dataset* which is a new corpus for Opinion Mining in newspaper articles. The corpus is created by professional specialists. The corpus can be used for several tasks in Opinion Mining: Sentiment classification [DLY08, SPV11, WWH09], Subjectivity Analysis [PL04, WWH09], opinion extraction [SPV11], the determination of viewpoints [GR09], the identification of argumentation stands [PLS11, SW10], and the creation of sentiment dictionaries [DLY08, RQH10].

The remainder of this chapter describes the following: In section 4.2 we characterise related work which addresses primarily corpora and language resources for Opinion Mining. In the third section, we explain our dataset and in particular the annotation scheme and differences to other resources. Then, we perform an inter annotators agreement study for the tonality. This illustrates how difficult the task is. We do not expect that our solution achieves a higher level of congruence between the machine and an analyst as two media analysts would achieve. Furthermore, we introduce our second corpus, the Finance dataset, and explain its characteristics, before we summarize shortly in the last section.

4.2 Related Corpora and Resources

Corpora and resources have been designed for many different tasks, because Opinion Mining and Sentiment Analysis [PL08] is a far-reaching subject.

In the context of film and customer reviews, the dataset of Hu and Liu [HL04] of 322 reviews, which have been increased to 445 documents [DLY08], is a benchmark dataset [DLY08] for sentiment analysis in product reviews. The products are two digital cameras, two cellular phones, a MP3 player, a DVD player, a router, and one anti-virus software. Another benchmark dataset of Pang and Lee [PL04] contains subjective and objective sentences which are extracted from film reviews and plots. They collect 5,000 subjective sentence and sentence fragments (from www.rottentomatoes.com) as subjective examples and 5,000 sentences from plot summaries (www.imdb.com) as objective examples. The sentences are at least 10 words long.

For Opinion Mining tasks in news articles, the MPQA Corpus [WWC05] contains a word- and phrase-based annotated corpus which consists of 535 English news documents (11,112 sentences and 19,962 subjective expressions). The tasks evaluated on this dataset cover contextual polarity [WWH09]. Here, we show a short excerpt from an MPQA document [WWC05] with added sentiment annotations in brackets:

(4.4) United Nations, New York, Nov 11, IRNA – President Mohammad Khatami and his Venezuelan counterpart Hugo Chavez here Saturday *cited* (neutral, medium) cooperation among OPEC member countries *as the key* (neutral, medium) *to forestalling further oil price declines* (negative, medium). The Iranian president also *hailed* (positive, medium) Chavez's recent efforts in travelling to the OPEC and non-OPEC member countries, in a bid to garner *support* (positive, medium) for propping up the sagging crude prices. [...]

In contrast to our corpus, here single words and phrases (cursive parts) are annotated with sentiments and the strength of a sentiment is given (in the example above,

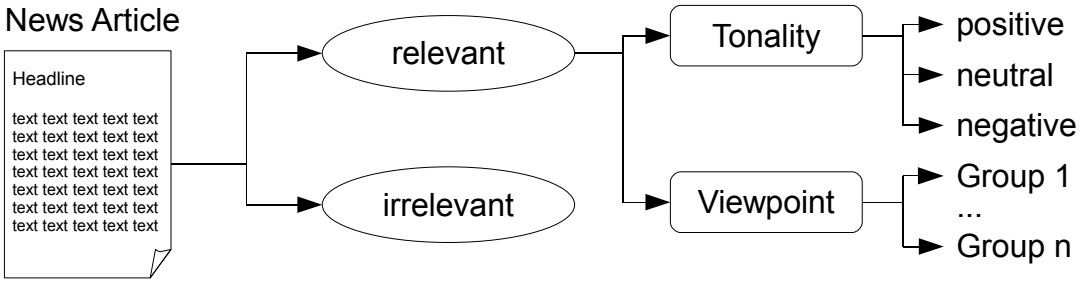

Figure 4.1: Hierarchical relationships between the categories of relevance, tonality, and viewpoint.

all intensities are *medium*, but also *low* and *high* are possible). It is not designed as an MRA, because it does not contain relevant areas or viewpoints.

To use Opinion Mining for forecasts, a corpus of the Canadian federal election prediction is used [KH07] to extract predictive opinions. It consists of 9,538 messages (on average 98.8 words) which are posted by many different users on a page of an election prediction project.

In German, resources are limited. SentiWS [RQH10] is the first publicly available dictionary for Sentiment Analysis. It includes 1,650 positive and 1,818 negative words in lemma, which cover 15,649 positive and 15,632 negative word forms in the German Language. Momtazi [Mom12] introduces the first German corpus for Opinion Mining in social media. It contains 500 short texts about celebrities.

The main difference is that all these corpora do not include marked areas which represent relevant statements and viewpoints for the tonality of statements. But these components are necessary to develop, to train, and to evaluate approaches which tackle the extraction of relevant statements, the calculation of their tonality, and the determination of their viewpoint for an MRA.

4.3 The Corpus

4.3.1 The Task of the MRA

Two media analysts (professional experts in field of media monitoring and analysis, hereinafter called the annotators) annotate news articles about the two biggest political parties in Germany: The CDU is the governing party under its chairwoman Chancellor Merkel and the SPD is the strongest opposition party. Web crawler systems have collected press releases because those have a huge media reach and must not be licensed or have archiving costs (what is a big problem for the provision of a publicly available MRA corpus). The annotators collected 617 relevant articles (consisting of 15,089 sentences) and annotated them as follows.

4.3.2 The Annotation Scheme

There are four different categories of text in an MRA: Positive statements, negative statements, neutral statements, and not relevant text areas. The text passages of statements (positive, negative, neutral) are also called relevant. Figure 4.1 shows the hierarchical relationships between these categories.

In the annotation process, the media analysts extract relevant statements. The statements have these attributes:

- **News no.**: This is the identification number of the news article which contains this statement.
- **Statement text**: The complete text of the statement is usually between one and four sentences long.
- **Tonality value**: The tonality can have one of three values (positive: one (1), neutral: zero (0), and negative: minus one (-1)).
- **Codes**: The codes specify the viewpoint. The statement is relevant and has the given tonality for the indicated company or organisation.

The corresponding news articles have the attributes **news no.**, **headline** and **text**. The crawlers have collected these articles because they contain the string 'CDU' or 'SPD'. Then the annotators have rejected articles which are no press releases or do not contain relevant statements.

4.3.3 Comparison with Other Resources

Table 4.1 compares different corpora for Opinion Mining with our corpus. The most similar corpus is the MPQA corpus, because it also deals with news articles. In the MPQA, the annotations cover single words and phrases, whereas our annotations belong to statements.

While it is possible to perform fine-grained word classifications about sentiments [WWH09] with the MPQA, our intention is to perform different tasks of an MRA. This requires statements which are annotated with tonality and a viewpoint. Our corpus contains 1,521 of these annotations (consisting of 3,283 sentences and 55,174 words). Table 4.2 shows the distribution of the statements on the three tonality classes and the two viewpoints. The corpus contains 992 statements for the CDU (207 positive, 319 neutral, 466 negative) and 529 statements for the SPD (155 positive, 260 neutral, 114 negative). The whole annotation process took 8 person days. The complete pressrelations dataset is available at `http://www.pressrelations.de/research/`.

Aspect	Pang [PL04]	MPQA [WWC05]	Momtazi [Mom12]	pressrelations [SCH12]
Language	English	English	German	German
Area	reviews	news	social media	news
Texts	-	523	500	617
Sentences	10,000	11,112	890	15,089
Containing Statements	no	no	no	yes (1,521)
Containing Viewpoints	no	no	no	yes (2)

Table 4.1: Comparison of resources for Opinion Mining.

Viewpoint	Positive	Neutral	Negative	All
CDU	257	265	470	992
SPD	189	227	113	529
All	446	492	583	1,521

Table 4.2: Distribution of tonality and viewpoints.

4.4 Inter Annotators' Agreement Study

This annotation process is challenging, because humans (even if they are professional experts such as our annotators) can interpret texts differently. We want to demonstrate that by analyzing the estimation of the tonality by our two annotators (annotator A and B). In 1960, Jacob Cohen introduced a measure for the agreement of two judges in such tasks. It is called the Cohen's kappa [Coh60] and well-acknowledged for annotation tasks [Mom12].

$$\kappa = \frac{p_0 - p_c}{1 - p_c} \tag{4.1}$$

It depends on the proportion of units in which the annotators agreed [Coh60], which is called p_0.

$$p_0 = \frac{\Sigma_{i=1}^{z} h_{ii}}{N} \tag{4.2}$$

And it also depends on the proportion of units for which the agreement is expected by chance [Coh60], which is called p_c.

$$p_c = \frac{1}{N^2} \Sigma_{i=1}^{z} h_{i.} h_{.i} \tag{4.3}$$

Annotator A	Annotator B			$h_{i.}$
	Positive	Neutral	Negative	
Positive	353	9	0	362
Neutral	89	478	12	579
Negative	0	10	570	580
$h_{.i}$	442	497	582	1521

Table 4.3: The agreement matrix for the calculation of Cohen's kappa.

In this two formulas, z is the number of possible categories (here $z = 3$) and N is the number of all decisions for one annotator (here $N = 1,521$). The border frequencies $h_{i.}$ and $h_{.i}$ are calculated by an agreement matrix. The matrix of our study is shown in table 4.3. Based on this matrix, the results are $p_0 \approx 0.9211$, $p_c \approx 0.3395$, and $\kappa \approx 0.8806$. The matrix also shows that the greatest possible difference in tonality (positive and negative or negative and positive, resp.) did not occur.

As a result, the inter annotators' agreement is 88.06% using Cohen's kappa. A value of over 80% is an almost perfect accordance. For the agreement of three or more annotators, the Fleiss' kappa has been introduced [Fle71], which is a further development of the Cohen's kappa. The quality assurance of an MRA ensures a annotation quality of at least 80% of correct annotations.

4.5 The Finance Dataset

In contrast to the pressrelations dataset, the Finance dataset is not created especially for the ATOM project. It represents a real world dataset, which contains an analysis for a customer of the pressrelations GmbH. The customer is a financial service provider and the analysis covers the customer itself and four of its competitors. In this thesis, the names of the companies are removed due to data protection. The complete dataset contains 5,352 articles with 8,500 statements (4,250 are neutral, 2,125 are positive, and 2,125 are negative) which were collected from 2010 until 2012. For the different tasks, we use different sizes of the dataset, because some examples are not useful for all tasks. Besides, the dataset has grown during our project, because the analysis was still running. So, the number of articles and statements will increase a little bit in the different analyses. Nevertheless, there are also other reasons, why the size can differ. Of course, the reasons can be the different aspects of the analyses, because the viewpoint features (cf. chapter 8) and the polarity of sentiment classification need no neutral examples (cf. chapter 6), for example.

4.6 Conclusion

The comparison of related work shows clearly that this corpus is required to develop and evaluate new approaches of Opinion Mining for an MRA and, in addition, for other tasks of sentiment analysis in German. Beyond the improvement of tonality classification the important tasks in further research are especially the extraction of relevant statements (which we show in the next chapter) and the identification of different viewpoints of statements (we talk about this issue in chapter 8). Contrary to the experiments, this corpus can be used to create large sentiment dictionaries including positive, negative, and also neutral examples, too.

5

Extraction of Statements

The first crucial step for an automated MRA is the extraction of statements, because, as shown in the last chapters, Opinion Mining and Sentiment Analysis for an MRA is not so interesting or, one could also say, not possible on whole documents. Thus, we concern the statements extraction step in this chapter. At this stage, we collect the opinions, but we do not value them. This step as a standalone system is a large alleviation of work for media analysts.

This chapter explains a method for extracting statements for an MRA. The task of statements extraction has many things in common with Text Summarization, because statements can be interpreted as summaries of our analysis objects. As a consequence, our evaluation compares our method with a Text Summarization approach and summaries based on coreference chains of our analysis objects. Furthermore, we compare these techniques with a state-of-the-art Opinion Mining technique, which extracts the most important and most subjective sentences.

Our proposed machine learning-based technique and the belonging evaluation was published in the paper "Extraction of Statements in News for a Media Response Analysis" on the 18th International conference on Applications of Natural Language Processing to Information Systems 2013 (NLDB 2013) [SC13a] in condensed form.

5.1 Motivation for an Extraction of Statements

As already discussed, an MRA means a big human effort, especially in the first steps, when the media analysts have to read and select statements. This means that they collect approximately 300 to 1,500 statements by reading 200 to 800 news articles each week. During this analysis, they have to remember all analysis objects, which can be

<headline>Greenpeace: Platzeck blocks energy turnaround</headline>

<text>Potsdam (ots) – Today Greenpeace activists protest against the climate-damaging direction of the Brandenburg prime minister Matthias Platzeck (SPD) on closed meeting of the SPD leadership in front of the island hotel Hermannswerder in Potsdam. The activists pile up 20 tons of lignite on the approach road and hold a banner "Dear SPD, Platzeck's lignite blocks the energy turnaround". In its concept of energy strategy 2030, the red-red government of Brandenburg still put on lignite which is the most climate-damaging energy source.

"The actual concept of energy strategy leads in an impasse for climate policy", says Greenpeace's energy expert Anike Peters. "It is a mistake to think that there will be an European infrastructure for carbon dioxide and in the same way lignite will be burnt. Prime minister Platzeck harms his country, if he continues to ignore the advantages of the phase-out of lignite. Renewable energy could generate more jobs and bring a greater value added into the country", says Peters. [...]

Greenpeace demands Mr. Platzeck to extend period for a statement to six weeks, as soon as it is possible to view the complete content of any related studies.

Greenpeace study: Lignite phase out promises benefits for Brandenburg

A new Greenpeace study shows that the number of jobs can increase from 11,500 today to more than 19,000 employees in the year 2030. [...] </text>

Figure 5.1: A translated example [SCH12] of a news article with statements.

sometimes several hundreds of entities.

The extraction of relevant statements is essential for an MRA [WN07, SCH12], because these statements contain the most relevant information for the customer of an MRA [WN07]. The statements represent tonality-bearing text parts in newspaper articles for the analysis objects. For an MRA, it is much less important to analyse the tonality [SCH12] of whole documents. We want to demonstrate that and the role of statements in the following: Figure 5.1 shows a translated example of the pressrelations dataset [SCH12]. The underlined passages are annotated statements of the dataset, which represent for us a gold standard. These two statements contain the most important information for the SPD (the governing party of the region Brandenburg) and both are annotated with a negative tonality. The marked sentences are not relevant for another party, e.g., and for Greenpeace other sentences are relevant: a relevant statement would be the last two sentences of the text snippet. So, the results of an MRA depend on the analysis objects (in general the customer of an MRA and its competitors or in the case of the pressrelations dataset the German parties SPD and CDU). Also other approaches show that a well-considered selection of text parts could improve Sentiment Analysis for opinion-bearing text [SPV11]. And, of course, our contributions of polarity/tonality classification and viewpoint determination work with statements. Here, our divide and conquer principle demonstrates one of its advantages: The extraction of statements is also interesting as a standalone system, because it would save much human effort during an MRA. The analysts would not have to read the complete articles, but they could only rate proposed statements. From an Information Extraction's point of view, the statement extraction can be used for an application, which collects the most important information for customers of media monitoring services. But now,

we begin with a formal definition of the task:

Task Definition I: Statements Extraction. *Let $d \in D$ be a document and D a collection of news articles. The task is to find a partition P of the set of all sentences S_d for every $d \in D$ so that P has ν elements and $\nu-1$ elements are relevant statements. ν is unknown before analysis.*

$$f_p : d \mapsto P = \{p_1, ..., p_\nu\} = \{\underbrace{\{s_j, s_{j+1}, ...\}, ..., \{s_k, s_{k+1}, ...\}}_{\nu-1}, \{s_l, s_m, ...\}\} \tag{5.1}$$

All p_i with $i \in \{1, ..., \nu-1\}$ include the relevant statements ($p_i \neq \emptyset$) and p_ν contains all not relevant sentences. A statement p_i ($i \in \{1, ..., \nu-1\}$) is a consecutive sequence of relevant sentences.

A statement usually consists of up to four sentences. In general, documents with only one element (all sentences are not relevant) and elements with only one sentence ($p = \{s_i\}$, e.g.) are possible.

As figure 5.1 shows, the relevant statements are not only sentences, in which certain search strings (such as 'SPD', 'Platzeck', or 'Greenpeace') appear. Sometimes a coreference resolution is needed (cf. the last sentence in the first statement), but sometimes even such resolution would not help (cf. the last sentence in the second statement). In our evaluation we will show that this is often the case. Moreover, the antepenultimate sentence contains the word 'Platzeck' and is not relevant, because it contains only additional information. So, we propose a machine learning technique which is based on significant features of relevant sentences and filters misclassified sentences by a density-based clustering.

The rest of the chapter is organized as follows: We discuss related work in the next section. Then we explain our two comparison methods DegExt [LLA+11] and RSUMM [SPV11] in section 5.3 and 5.4. In section 5.5 we explain our machine learning-based method for the statement extraction. We evaluate and compare the results of our approach with these techniques in section 5.6, before we conclude in the last section of this chapter.

5.2 Related Work on Statements Extraction

The extraction of relevant statements for an MRA is related to several kinds of domains: Text Summarization [AHG99, Tur00, MT04, LLA+11], Information Extraction [IAH+08, WPS10, HZM+11] and Opinion Mining [SPV11, SCW12, SC12].

In the history of approaches for Text Summarization, an early contribution works with coreference chains [AHG99] to estimate the sentences of a summary. We take

up this idea and create summaries based on our analysis objects as a first baseline. Generally, there are two types of summaries: An automated summarization of single documents [AHG99, Tur00, MT04, LLA+11] and a summarization of multiple documents [CHT11, SB12], which is not appropriate for our task. Turney extracts important phrases by learned rules [Tur00], while Mihalcea and Tarau build graphs using Page Rank and a similarity function between two sentences [MT04]. A language-independent approach for Text Summarization proposed by Litvak et al. [LLA+11] is called **DegExt**. We explain this approach elaborately in the next section.

Turning to contributions of the Information Extraction domain, different, but somehow related tasks and definitions of statements can be found. Inui et al. [IAH+08] perform experience mining: A system analyses texts where people tell how they gain experience in their everyday life. Finally, this system collects opinions from these experiences. Another paper [WPS10] extracts statements for market forecasts. Here, a statement consists of a 5-tuple of topic, geographic scope, period of time, amount of money or growth rate, and the statement time, whereas the relation of time and money information is particularly important. Hong et al. [HZM+11] extract events from sentences. They extract the type of the event, its participants, and their roles. Unfortunately, all three definitions of statements/events and their methods do not fit in our issue.

As we have mentioned before, the approach of Sarvabhotla et al. [SPV11], called **RSUMM** (**R**eview **Summ**ary), creates summaries of reviews for Opinion Mining tasks. They weight sentences by the importance of the containing words and the subjectivity. In this way, they select the most important and subjective sentences for their subjective excerpt [SPV11]. This fits very well with our statements extraction task, so we apply two variants of this approach for our evaluation: One 'classical' variant, which is described in the paper [SPV11] and a second variant, which we expand with machine learning. The selection of sentences is explained extensively in section 5.4.

To evaluate our approaches, we use metrics of the Text Summarization area, because this field has several things in common with our task. Lin [Lin04] proposes widely acknowledged metrics to estimate the quality of text summaries. We use the ROUGE-L score to determine the quality of the extracted statements, which we are going to explain in our evaluation section of this chapter (cf. section 5.6.2). Lin also introduces the scores ROUGE-W, ROUGE-S and ROUGE-SU, which can handle different word orders. However, all methods select the sentences from the origin text, so the word orders will remain constant. Moreover, Lin introduces the ROUGE-N score which refers to the n-gram co-occurrence statistic. In our case, it would show the proportion of relevant n-grams in the extracted statements. Nevertheless, the informational content of the ROUGE-N score is lower than the informational content of the ROUGE-L score

No.	Sentence
0	Annual report: Government bank alone upon hope.
1	To the annual report 2012 of the federal government the economic policy spokesmen of SPD faction Garrelt Duin declares:
2	The German economy is strong, but not invulnerable.
3	The federal government disregards the imponderables of the international economy by its 0.7 percent growth prognosis.
4	This fact is negligence.

Table 5.1: Translated example text snippet from [SCH12].

in this case (cf. section 5.6.1).

5.3 Statement Extraction with DegExt

The Text Summarization method DegExt [LLA+11] is very language-independent, because the only required NLP resources are a tokenizer and a sentence splitter (or instead of using a sentence splitter, a list of sentence-terminating punctuation marks is provided [LLA+11]). Litvak et al. [LLA+11] report better results than TextRank [MT04] and GenEx [Tur00] on the benchmark corpus of summarized news articles of the 2002 DUC by extracting 15 keywords. So, we take DegExt as one of our comparison methods.

The approach transforms a given text into a graph representation where words become nodes. For an illustration, table 5.1 shows an example which is a text snippet of the pressrelations dataset [SCH12]. The sentences are numbered consecutively. Besides, sentences 1 to 4 form a statement in the dataset.

From this text, DegExt builds the graph which is shown in figure 5.2. DegExt creates a unique node for every word in the text (if a word appears a second time, DegExt does not create another node). For this step, DegExt can remove stopwords and stem (lemmatize) the remaining words, if the required NLP resources exist for the analyzed language [LLA+11]. These are optional steps. For this purpose, we use the TreeTagger for lemmatization and the stopword list of RapidMiner[1]. Then DegExt creates a direct edge from node A to node B when the word of node A immediately precedes (after the stopword removal) the word of node B in one sentence. Here, multiple edges from one node to another are possible (figure 5.2 shows this with multiple numbers of the sentences).

[1]RapidMiner (http://rapid-i.com/)

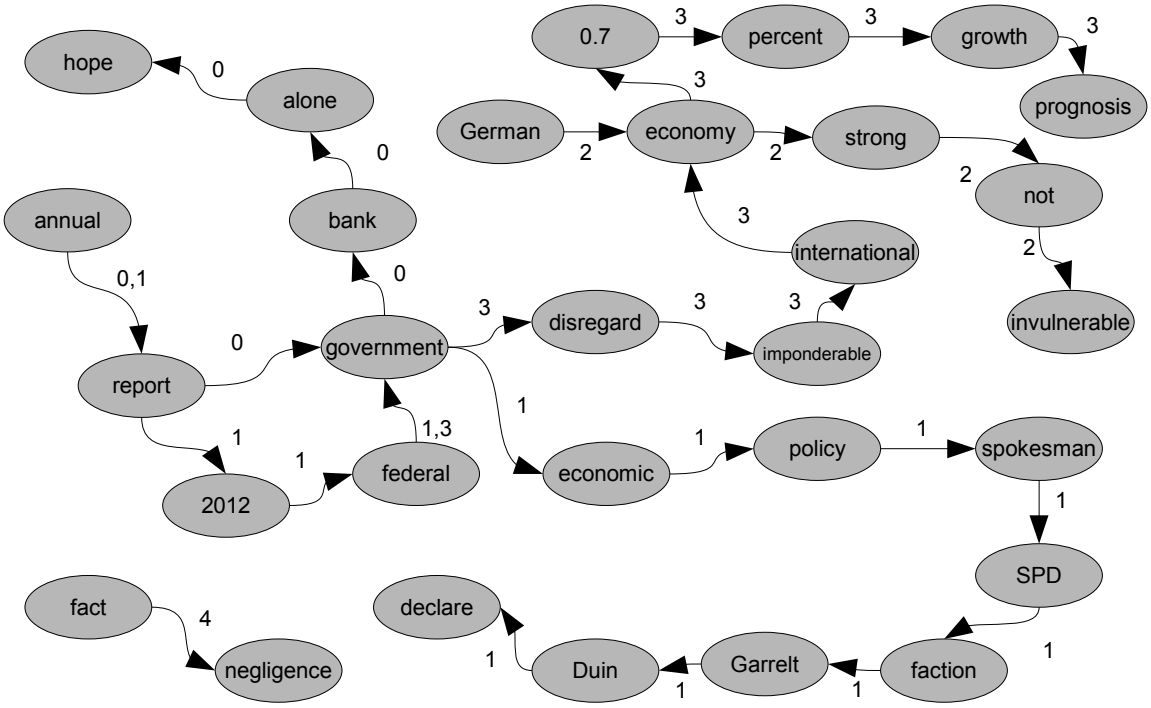

Figure 5.2: Graph representation of the example text.

Within this graph, the important words are estimated by nodes with a high connectivity. In our example, there are words such as "government", "economy", "report", and "federal". These words are extracted as keywords of the text. Litvak et al. [LLA+11] also propose a method for the extraction of keyphrases. Since this has no relevance to our tasks, we skip this point.

A summary of one document consists of all sentences which contain at least one keyword. DegExt allows to choose the number of keywords (referred to as N) and, as a consequence, the size of the summaries. We test several values for N, because the results of the experiments of Litvak et al. show that the choice of N is important for the quality of the result [LLA+11]. Consecutive sentences of a summary are combined to a statement.

5.4 Mining of Opinions with RSUMM

Sarvabhotla et al. [SPV11] propose a two-step approach for Opinion Mining. In the first step, they use a statistical technique to extract the most important and most subjective sentences. This methodology goes very well with our task of statements extraction. The second step contains a tonality classification which is based on a machine learning technique and a well-considered selection of features. In this section, we only focus on the first step. The second part will be taken up in section 7.1.4, when we talk about the tonality classification.

RSUMM uses the vector space model [SWY75]. RSUMM represents each sentence $s \in d$, where d is a document of our document collection D (cf. section 5.1), as a vector $\vec{s}$ of terms [SPV11]:

$$\vec{s} := (t_{s,1}, t_{s,2}, ..., t_{s,n}) \tag{5.2}$$

Sarvabhotla et al. [SPV11] define the two metrics ADF (average document frequency) and ASM (average subjective measure), from which they derive two term vectors $\overrightarrow{adf}$ and $\overrightarrow{asm}$ [SPV11]. They calculate the most interesting sentences by a lexical similarity between $\vec{s}$ and the vectors $\overrightarrow{adf}$ and $\overrightarrow{asm}$ [SPV11].

5.4.1 Average Document Frequency

To obtain the most relevant terms, they compute the average document frequency (ADF) of an annotated collection of texts C_{pol}:

$$ADF(C_{pol}) = \frac{\sum_{i=1}^{|V_{pol}|} df(t_i, C_{pol})}{|V_{pol}|} \tag{5.3}$$

In this equation, $|V_{pol}|$ is the size of the vocabulary of collection C_{pol} and $df(t_i, C_{pol})$ is the document frequency of term t_i in C_{pol} [SPV11]. Then they calculate the vector $\overrightarrow{adf} := (t_{adf,1}, t_{adf,2}, ..., t_{adf,n})$, where each $t_{adf,i}$ have a document frequency greater than $ADF(C_{pol})$.

5.4.2 Average Subjective Measure

To obtain the vector $\overrightarrow{asm}$, Sarvabhotla et al. [SPV11] calculate the average subjective measure $ASM(C_{sub})$ of an annotated document collection C_{sub}:

$$ASM(C_{sub}) = \frac{\sum_{i=1}^{|V_{sub}|} \Phi(t_i, C_{sub})}{|V_{sub}|} \tag{5.4}$$

Here, $|V_{sub}|$ is the size of the vocabulary of collection C_{sub}, while $\Phi(t_i, C_{sub})$ is a subjective measure of term $t_i \in C_{sub}$:

$$\Phi(t_i, C_{sub}) = \frac{subj(t_i, C_{sub})}{obj(t_i, C_{sub}) + tot(C_{sub})} \tag{5.5}$$

In this subjective measure, $subj(t_i, C_{sub})$ is the frequency of term t_i in the subjective instances of the annotated collection C_{sub}, while $obj(t_i, C_{sub})$ is the frequency in objective instances and $tot(C_{sub})$ is the total number of instances in C_{sub} [SPV11]. As an analogy to the vector $\overrightarrow{adf}$, the vector $\overrightarrow{asm}$ contains only terms of C_{sub}, which has a subjective measure greater than $ASM(C_{sub})$.

5.4.3 Final Scoring

For the final scoring, they measure the similarity between $\vec{s}$ and the vectors $\overrightarrow{adf}$ or $\overrightarrow{asm}$, respectively, for each sentence s. For this purpose, they apply the Jaccard similarity measure [Jac01] to obtain the similarity score $\sigma(\vec{a}, \vec{b})$:

$$\sigma(\vec{a}, \vec{b}) = \frac{n(\vec{a} \cap \vec{b})}{n(\vec{a} \cup \vec{b})} \tag{5.6}$$

$n(\vec{a} \cap \vec{b})$ denotes the number of matching terms in both vectors $\vec{a}$ and $\vec{b}$ and $n(\vec{a} \cup \vec{b})$ are the number of all terms in the vectors $\vec{a}$ and $\vec{b}$. The sum of both similarities represents the final score $FS(s)$ for a sentence s:

$$FS(s) = \sigma(\overrightarrow{adf}, \vec{s}) + \sigma(\overrightarrow{asm}, \vec{s}) \tag{5.7}$$

RSUMM ranks the sentences of one document d by $FS(s)$ and extracts the top X% of the ranked sentences in decreasing order as the subjective excerpt [SPV11]. After this first step, the method represents the document d as the subjective excerpt. More concrete, this means that a document is represented through a term vector of the top X% ranked sentences. In the next step, the approach would select from these terms the most important ones and classify the tonality with an SVM. We will explain this step in more detail later on, because at the moment we use exclusively the first step in order to extract the statements.

We evaluate the RSUMM method [SPV11] in two variants: The 'classical' method (denoted as RSUMM X%) selects the top X% of the sentences, which got the highest scores, as outlined above. The subjective excerpt contains the relevant statements. Consecutive sentences in the subjective excerpt are combined to one statement. In this way, we can select the most important and most subjective sentences, so this technique goes well with our task. We use 20% of our training examples (C_{pol} and C_{sub}) to create the vectors $\overrightarrow{adf}$ (average document frequency) and $\overrightarrow{asm}$ (average subjective measure) [SPV11].

As a second variant, we use both RSUMM scores as input values for a classifier (denoted as RSUMM (+SVM)) and classify every sentence. Sarvabhotla et al. use the SVMLight package[2] [Joa99], so we apply this learner. But we obtain a very low accuracy (16.43% by using 50% for training, e.g.), because the classifier tends to qualify every sentence as relevant (one reason is maybe the small number of features). As a consequence, we use the SVM of our technique (cf. section 5.5.1) which achieved better results (cf. section 5.6.2). The reason for this could be the *balance cost* parameter (also explained in section 5.5.1).

[2] http://svmlight.joachims.org/

<headline>Work of the Internet-Enquete is not yet finished</headline>

<text>Lars Klingbeil, who is the net-political speaker of the SPD Parliamentary Group, responds to the declaration of the Union fraction about the Enquete-Commission "Internet and digital society":

The establishment of a committee about network policy and digital society is right. The SPD Parliamentary Group understands network policy as fundamental and comprehensive approach which has to be reflected in different fields of politics. Network policy is social policy.

Therefore, the aim must be, that net policy is anchored in every committees of the Bundestag prominently.

The establishment of net political issues needs time, so for a transitional period we need an independent and full rights main-committee about net policy and digital society, which in charge of these topics in the Bundestag. Therefore, the SPD Parliamentary Group pressed for reinstatement of the subcommittee "new media" against the resistance of the Union at the beginning of the legislative period. Even then we demanded it as a proper committee.

It is correct that this new main-committee will be established after the end of the Internet-Enquete. [...] </text>

Figure 5.3: Second translated example of an annotated news item [SCH12].

5.5 Our Machine Learning-based Method for Statements Extraction

Now, we present our approach for the extraction of statements. It is composed of three phases:

- The first phase is learning and classifying relevant sentences.
- The second phase filters misclassified sentences by a density-based clustering.
- In the last phase, relevant and not clustered sentences are combined to statements.

We start with learning and classifying relevant sentences in news articles in the next section.

5.5.1 Learning Relevant Sentences

As shown in the examples (figures 5.1 and 5.3), statements are not just consecutive sentences or whole paragraphs, which contain certain search strings such as the name of a person or a party. In figure 2, the last sentences of each statement do not contain a keyword such as 'SPD'.

We propose an approach based on machine learning for the extraction of relevant statements. Thereby, we consider the input (new texts) as a sequence of sentences, and we decide for every sentence: Is this sentence relevant or not? For this task, we extract different features (cf. table 5.2) which indicate the importance of a sentence for an MRA: First, we count the number of important persons and organisations in sentences

Classification Features:	Clustering Features:
Entity Features:	*Word Features:*
k_1: number of important organisations	$(c_1, c_2, ..., c_{\|V\|})$:
k_2: number of important persons	the frequencies of words in
k_3: number of organisations	the statements classified
k_4: number of persons	as relevant
Importance of Words Features:	(c_i = frequency of term i
k_5: number of headline words	in the sentence; V is the set of
k_6: average TF-IDF score	all terms in all relevant sentences)

Table 5.2: Feature set for our SVM classifier and our density-based clustering.

(an organisation could also be a product; some funds are a product and a (subsidiary) company, e.g.). In an MRA, some people are of particular importance, because they are press spokesmen/spokeswomen of a relevant organisation (the customer's company or competitor) or they are an advertising medium. Media analysts collect lists of these entities in so-called codebooks [SC12], because it is very difficult for humans to remember all relevant persons and organisations.

For POS-tagging and lemmatisation, we use the TreeTagger [Sch94, Sch95]. For a Named Entity Recognition (NER), we apply our Information Extraction module (cf. section 2.4.2). Our new JAPE Rules improve our NER by handling all important entities from our codebook with the highest priority (cf. section 2.4.2). This secures that these entities are found with a very high probability. We apply our coreference resolution, too.

We count all elements of the coreference chains which belong to (important) persons and organisations/products. So, we call k_1 the number of important organisations and k_2 the number of important persons, while k_3 is the number of all organisations and k_4 the number of all persons in one sentence (cf. table 5.2). Also, it is significant how many headline words appear in the sentence. Headlines of news articles contain often compressed information about the whole article and so an occurrence of headline words can indicate a relevant statement. Therefore, k_5 is the number of headline words in a statement. Likewise, the statements themselves reflect important information about the article. For this reason, we measure the average TF-IDF score of all words in the sentence (k_6 in table 5.2).

For the classification, we use an SVM. We use the SVM standard implementation of RapidMiner[3]. We force the SVM to balance the performance on the two classes

[3]RapidMiner (http://rapid-i.com/)

through the *balance cost* parameter, because there are many more irrelevant instances (cf. section 5.6.2). The balance costs parameter ensures that the hyperplane of the SVM is optimized in such a way that the precision and recall of both classes are high instead of optimising the classification accuracy only, which is independent from single classes.

5.5.2 Filtering by Density-based Clustering

After the classification of each sentence, we select all sentences which are classified as relevant. For every sentence, we count the frequency of every word in the sentence and use these frequencies as input features (cf. table 5.2) for the performance of a DBSCAN clustering [EKSX96]. We use a density-based clustering, because this procedure has a large advantage for us: We do not have to specify the numbers of clusters before the clustering. In general, we do not know the number of clusters, because we see a member of a cluster as a classification mistake here.

In a clustering approach, the clusters are usually more interesting in order to identify objects, which share commonalities. But statements are representing many different pieces of information and opinions over a large document corpus [WN07]. So, our approach works the other way round and filters out clusters of not relevant sentences because really relevant sentences tend to be noise while the same classification mistakes appear several times and thus become clusters. Thereby, we use only sentences which are noise from a clustering perspective (cf. next section). Since only the sentences classified as relevant are used for our clustering, computational time can be saved for the performance of the clustering.

We set the parameter *Eps* [EKSX96] (the radius of a neighbourhood) to 1.0 and *MinPts* [EKSX96] (the number of minimum points in an *Eps* neighbourhood including the starting point itself) is set to 3. This secures that the clusters are very similar and at the same time a similar misclassification occurs at least three times. Sometimes DBSCAN has problems to detect clusters, if the density differs from one to the other clusters. We have noticed that the variation of the density between different clusters is not so large. Thus, we do not plan to apply OPTICS [ABKS99] or other clustering techniques, which can handle this problem.

5.5.3 Statements Extraction Step

Our technique combines sentences which are classified as relevant by our SVM and do not belong to any cluster in DBSCAN clustering. The input parameters of the algorithm are the set of all sentences, the calculated classification model, and the calculated clustering model (here, it is only important, if a sentence is identified as

noise or has any cluster id). The algorithm (cf. algorithm 1) filters out sentences which are not relevant or belong to a cluster, and combines the remaining sentences, if they are consecutive.

Algorithm 1: Statements Extraction

```
Data: Sentences S, Classification Model K, Clustering Model C
Result: List of Statements R
R = create an empty list of statements;
foreach s ∈ S do
    if K(s) = RELEVANT and C(s) = NOISE then
        add s in R;
    end
end
foreach r1 ∈ R do
    foreach r2 ∈ R do
        if r1.EndOffset + 1 = r2.StartOffset then
            remove r1 and r2 from R;
            add (combine(r1, r2)) in R;
        end
    end
end
```

The method `combine` takes two consecutive statements and appends the second one to the first one. `R` contains all p_i with $i \in \{1, ..., \nu - 1\}$ and p_ν are all sentences which are not a part of an element in `R`.

5.6 Evaluation

5.6.1 Baselines and Codebooks

We compare DegExt [LLA+11] and RSUMM [SPV11] with our approach in this evaluation. Furthermore, we apply two other baselines: We construct simple bags of words for every sentence to classify the sentences by our classifier (denoted as TSF-Matrix 5%, where TSF stands for term sentence frequency and the size of the training data is 5%). Likewise, we use only the extracted coreference chains of our important entities to identify statements (denoted as Coreference Chains): If one element of a chain of an important entity appears in the sentence, the sentence is relevant and consecutively relevant sentences are combined to statements.

We test the methods on our datasets: The pressrelations dataset [SCH12] has 617 articles with 1,521 gold statements and Finance dataset with 5,000 articles of an MRA about a financial service provider and four competitors. The articles include 7,498 statements.

Method	Data for Training	Accuracy	Not Relevant		Relevant	
			Precision	Recall	Precision	Recall
RSUMM (+SVM)	2.5%	0.6403	0.2591	0.5923	0.8854	0.6503
RSUMM (+SVM)	5%	0.6938	0.8579	0.7556	0.2501	0.3943
RSUMM (+SVM)	10%	0.659	0.8659	0.6969	0.2434	0.4246
RSUMM (+SVM)	15%	0.6525	0.8661	0.6866	0.2443	0.4882
our approach	2.5%	0.4918	0.8315	0.4872	0.1694	0.5143
our approach	5%	0.8172	0.8912	0.8885	0.4597	0.4667
our approach	10%	0.8178	0.8914	0.8892	0.4601	0.4656
our approach	15%	0.8173	0.8111	0.8890	0.4479	0.4633

Table 5.3: Results of the sentence classification on the pressrelations dataset.

The codebook for the Finance dataset includes 384 persons, 19 organisations, and 10 products, while the codebook for the pressrelations dataset contains 386 persons (all party members of the 17th German Bundestag[4], the German parliament), and 18 entries of organisations (names and synonyms of the parties and concepts such as 'government' or 'opposition' [SC12]). We will talk in more detail about codebooks later on, when we introduce our algorithm for the determination of viewpoints in chapter 8.

5.6.2 Results

For the step of learning relevant sentences, table 5.3 and 5.4 show the results for classifying single sentences as relevant or not. As the tables show, our classifier needs only very limited training data (5% or 0.5%, resp.) to obtain good results. There is nearly no difference between using 15% or 5% on the pressrelations dataset, for instance (cf. table 5.3). On Finance, the classifier requires even less data for good results (cf. table 5.4). The results show that it is more difficult to identify the relevant sentences, while precision and recall of not relevant examples are very high. One reason is the unequal distribution of the two classes, of course. The pressrelations dataset contains 3,283 relevant sentences and 11,806 sentences are not relevant, while Finance includes 13,084 relevant sentences and 145,219 not relevant sentences. However, the tables show that our method achieves better results on sentence level than RSUMM (+SVM).

For our further experiments, we use only 5% on the pressrelations dataset and 0.5% on the Finance dataset for training, because these values achieve good results and, for a practical solution, a technique should require as less training as possible, because more training means for our project that we can save less human effort. Here, we measure how many statements match the annotated statements of the two datasets (denoted as

[4]collected from http://www.bundestag.de

Method	Data for Training	Accuracy	Not Relevant		Relevant	
			Precision	Recall	Precision	Recall
RSUMM (+SVM)	0.25%	0.6883	0.941	0.7108	0.0831	0.3702
RSUMM (+SVM)	0.5%	0.7067	0.9407	0.7321	0.0843	0.3481
RSUMM (+SVM)	1%	0.7579	0.9395	0.7917	0.0872	0.2807
RSUMM (+SVM)	5%	0.7075	0.9408	0.7329	0.0847	0.3488
our approach	0.25%	0.5045	0.954	0.4931	0.0853	0.6653
our approach	0.5%	0.9296	0.9575	0.9675	0.4641	0.3958
our approach	1%	0.9072	0.9614	0.9383	0.3514	0.4698
our approach	5%	0.9073	0.9618	0.9384	0.3515	0.4704

Table 5.4: Results of the sentence classification on Finance.

Gold Standard Match). As well, we use the ROUGE-L score [Lin04] which is based on the idea that two summaries are similar, if the size of the longest common subsequence (LCS) [Lin04] is large:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \qquad P_{lcs} = \frac{LCS(X,Y)}{n} \qquad F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{5.8}$$

X is the annotated statement of the dataset, Y is the candidate statement, m is the length (in characters) of the gold statement X and n is the length of Y. The typical ROUGE-L score is the LCS-based F-measure, where β is set to a very high number and therefore the F-measure only depends upon R_{lcs}. We proceed not in the same way, because we are also interested in a high precision (a wrong statement can falsify the results of an MRA or means more effort to check the results). Therefore, we set $\beta = 1$ for F_{lcs}, but we report the R_{lcs} and P_{lcs} values, too.

The results of the final generation of statements are shown in table 5.5 and 5.6. The results of our approach are listed in two lines: The first line shows the results without the clustering step (denoted as our approach), which is added for the results of the second line (+ clustering). Our method achieves the best F-measure value for the identification of the perfect match of the gold statements and at the same time the best ROUGE-L values on the two datasets. The F-score of the gold match is an improvement of over 7 or 14 percentage points, resp., in comparison with the second best method (RSUMM 10%). The F_{lcs} values are over 20 or 27 percentage points higher than our second baseline (the TSF-Matrix 5%). The results show that the clustering can increase the F-score and ROUGE-L scores of F_{lcs}, especially on the pressrelations dataset by over 2 percentage points.

The DegExt method is most effective by using N=6 or N=5 on the pressrelations or

Method	Extracted Statements	Gold Standard Match			ROUGE-L		
		Prec	Rec	F1	P_{lcs}	R_{lcs}	F_{lcs}
DegExt (N=1)	758	0.0752	0.0375	0.05	0.384	0.1278	0.1918
DegExt (N=2)	1,349	0.0801	0.071	0.0753	0.3929	0.2092	0.273
DegExt (N=3)	1,869	0.0776	0.0953	0.0855	0.3749	0.2603	0.3073
DegExt (N=5)	2,679	0.0646	0.1137	0.0824	0.3382	0.3085	0.3227
DegExt (N=6)	2,948	0.0648	0.1256	0.0855	0.332	0.327	0.3295
DegExt (N=7)	3,141	0.0592	0.1223	0.0798	0.3241	0.3324	0.3282
DegExt (N=8)	3,246	0.0564	0.1203	0.0768	0.3196	0.3348	0.327
DegExt (N=10)	3,358	0.0497	0.1098	0.0685	0.3172	0.3301	0.3235
DegExt (N=15)	3,338	0.0419	0.092	0.0576	0.3081	0.2988	0.3034
RSUMM (5%)	725	0.1807	0.0889	0.1192	0.3345	0.1734	0.2284
RSUMM (10%)	1,359	0.1405	0.1297	0.1349	0.3152	0.2761	0.2944
RSUMM (15%)	1,971	0.1152	0.1541	0.1318	0.2893	0.3513	0.3173
RSUMM (20%)	2,587	0.0928	0.1629	0.1182	0.2665	0.4171	0.3252
RSUMM (25%)	3,243	0.082	0.1806	0.1128	0.2438	0.4874	0.325
RSUMM (+SVM)	3,200	0.0816	0.1716	0.1115	0.2452	0.4776	0.324
TSF-Matrix 5%	2,321	0.0866	0.1321	0.1046	0.363	0.3399	0.3511
Coreference Chains	891	0.1852	0.1085	0.1368	0.5551	0.2778	0.3703
our approach	2,233	0.1536	0.2258	0.1828	0.5545	**0.4976**	0.5245
+ clustering	1,841	**0.1896**	**0.2302**	**0.2079**	**0.6302**	0.4951	**0.5545**

Table 5.5: Results of the statements extraction on the pressrelations dataset.

Finance dataset, respectively. DegExt obtains an F-measure of 8.55% or 2.57% of the gold standard and the score F_{lcs} is 29.69% or 19.31%, respectively. RSUMM achieved better F-scores than DegExt in the match of the gold standard, but the ROUGE-L scores of F_{lcs} are nearly the same. The parameter X has less effect on both F-scores (a higher X value increases recall in the same way it decreases precision). The results show that a coreference resolution (as a preprocessing step of our approach) achieves partially precise results, but it only finds a smaller proportion of relevant statements.

But how important is a perfect match of the gold statements? If we take a look at figure 5.1 and 5.3, it is hard to decide even for humans, where a statement starts or ends. In many cases (as in example 5.1 and 5.3) it is not important, if a statement starts one sentence earlier or ends one sentence later which is often the case for the extracted statements (the ROUGE-L scores show this, e.g.). This is the reason for the low percentage values of the recall, but what is the reason for low precision values? Are so many machine-generated statements not relevant? The most approaches tend to extract more statements as in the gold annotation and we perform a deeper analysis of extracted statements in the next section.

Method	Extracted Statements	Gold Standard Match			ROUGE-L		
		Prec	Rec	F1	P_{lcs}	R_{lcs}	F_{lcs}
DegExt (N=1)	5,630	0.0238	0.0179	0.0204	0.2205	0.1077	0.1447
DegExt (N=2)	10,720	0.0207	0.0296	0.0244	0.2062	0.1655	0.1836
DegExt (N=3)	15,159	0.0181	0.0367	0.0242	0.1955	0.2042	0.1998
DegExt (N=4)	18,752	0.0175	0.0439	0.025	0.1883	0.2332	0.2084
DegExt (N=5)	21,724	0.0173	0.0501	0.0257	0.1826	0.2528	0.212
DegExt (N=6)	24,022	0.0165	0.0528	0.0251	0.1769	0.2628	0.2115
DegExt (N=7)	25,899	0.016	0.0552	0.0248	0.1725	0.2673	0.2097
DegExt (N=10)	29,518	0.0143	0.0561	0.0228	0.1655	0.2739	0.2063
DegExt (N=15)	32,117	0.0136	0.0583	0.0221	0.1596	0.2707	0.2008
RSUMM (5%)	8,214	0.0435	0.0507	0.0468	0.1775	0.1749	0.1762
RSUMM (10%)	15,090	0.0366	0.0784	0.0499	0.1649	0.2472	0.1978
RSUMM (15%)	21,905	0.0305	0.095	0.0462	0.1531	0.3045	0.2038
RSUMM (20%)	28,588	0.0271	0.1101	0.0435	0.1449	0.3524	0.2054
RSUMM (25%)	35,498	0.0243	0.1226	0.0406	0.1373	0.4023	0.2047
RSUMM (+SVM)	54,339	0.004	0.0312	0.0071	0.11	0.2343	0.1497
TSF-Matrix 5%	37,105	0.0258	0.129	0.043	0.2068	0.3877	0.2697
Coreference Chains	5,378	**0.1991**	0.1428	0.1663	0.6059	0.3572	0.4494
our approach	7,937	0.1713	0.2176	0.1917	**0.6312**	0.4754	0.5423
+ clustering	7,899	0.1707	**0.2212**	**0.1927**	0.6295	**0.4846**	**0.5476**

Table 5.6: Results of the statements extraction on the Finance dataset.

Method	Precision	Recall	F-score
DegExt (N=6)	0.4156	0.7076	0.5236
RSUMM (20%)	0.4846	0.7433	0.5867
our approach	0.7968	0.8499	0.8225

Table 5.7: Results of reconsidering the statements extraction on the pressrelations dataset.

5.6.3 Profound Analysis of the Extracted Statements

In examining the reprocessing issue in detail, the high precision of the ROUGE-L score and the low precision in the match with the gold statements are remarkable. On the one hand, the method can find the most important information in statements, but on the other hand, why did this technique (and most of the other methods) tend to extract more statements than the number of gold statements?

Two media analysts examined all extracted statements on the pressrelations dataset in a blind study (they do not know the extraction method) and reconsider all extracted statements, even those which are not a part of a gold statement. In this analysis, an extracted statement is also correct, when it is relevant (relevant information about the analysis objects), it can be rated as positive, neutral, or negative, and a viewpoint

can be estimated (for whom the statement is positive, e.g.). We use the comparison methods with the best parameters (based on F_{lcs}) and our approach (including clustering).

The findings are depicted in table 5.7. Here, the F-score is over 23 percentage points higher than the second best approach (RSUMM (20%)). This analysis shows that the approach extracts many more relevant statements which are not part of the gold annotation. There are several reasons for this: In an MRA [WN07] sometimes only a number of top-N statements are used for the analysis. So, besides the gold statements which are found exactly or partially, the machine-based approaches find more statements, which are less important, but nevertheless adequate statements. Furthermore, many of these statements are neutral, so that they are not all extracted, because too many neutral statements may dilute the tonality in a practical analysis. As well, it is even for humans a hard task, of course.

5.7 Conclusion

Our approach outperforms DegExt [LLA+11], RSUMM [SPV11] and all baseline methods on both datasets. The findings point out that the extraction of statements for an MRA could not only be solved by Text Summarization. Furthermore, our evaluation shows that our technique can find many adequate statements. On the one hand, this approach can be utilized to help media analysts who could save time by reading news articles and extracting relevant statements. And on the other hand, our method is the first step for an automated approach for an MRA, because the combination of this approach and the other modules of our divide and conquer policy (the classification of the tonality and the determination of perspectives) represents a fully automated generation of analysis data for an MRA. We explain these tasks and our solution in the next chapters.

6

Determining the Polarity of Sentiment

We want to approach the problem of an automated classification of the tonality of statements during this chapter. We believe that the classification of the tonality is the most difficult subtask of our automated approach for an MRA. Many contributions in literature [KS06a, WWH09, TBT+11] support this thesis. As we have seen in chapter 4, humans have the most problems to distinguish between neutral and subjective statements. So, we concentrate on the distinction between positive and negative statements first of all. We create sentiment dictionaries and, consequently, propose an automated approach for the distinction between positive and negative statements. In this way, we want to learn from the results of this task for the tonality classification (which is our main issue in the next chapter).

Many publications in the area of Opinion Mining talk about the sentiment [SM08, DTCY10, BWC11, FWY+11] or the polarity of sentiment [WWH09]. Besides, some publications [DLY08, TBT+11] speak about the orientation of words. The scientific literature knows several definitions of the concept sentiment. A sentiment can mean positive and negative opinions [DTCY10, BWC11], it can also include neutral utterances [PP10, KS06a], or a sentiment refers to a point-based scale (a five stars rating for customer reviews [PL05], e.g.). Similarly, sentiments can be interpreted as emotions such as joy, fear, and anger [SM08], or encouragement and sadness [FWY+11]. In this chapter, we want to focus on the polarity of sentiment [WWH09] (or in short the polarity [KK07]), which we define as positive or negative statements. An approach for the automated estimation of the polarity of sentiment can be used for trend analyses, in which it is not so important to analyse opinions about analysis objects in detail, but

it provides a coarse-grained overview and a trend about the polarity of sentiment for the objects: Are the statements positive or negative in the majority of the cases, or is the ratio balanced?

This chapter has three parts. In the first part, we try several word-based approaches by starting with a simple bag-of-words approach for machine learning. Then we expand our research by the construction of sentiment dictionaries for single words. We implement and evaluate many different methods in order to weight the words with a sentiment value. But we also design approaches based on combinations of words, because we believe that an approach based on isolated words alone is not the best way in order to tackle our problem. We believe that this is even more true in a classification of the tonality. As a consequence, we test several structures consisting of several words, which we found in literature. This part is published in the paper "Comparing Different Methods for Opinion Mining in Newspaper Articles" [SCW12].

In the second part of this chapter, we expand the best performing dictionary-based approaches by analysing the linguistic context of the words. In the linguistic context, we analyse linguistic factors such as negations, conjunctions, or modals and we identify which words are influenced by these factors and in which way. These techniques are published in the paper "Linguistic Sentiment Features for Newspaper Opinion Mining" [SC13b].

This chapter will be concluded by an examination of the limits of these combined approaches, when neutral examples are involved. Therefore, we evaluate several features, which are proposed in different contributions [BCMP11, TBT+11] for Subjectivity Analysis and try a first tonality classification.

6.1 Introduction

At this stage, we formulate our task as the determination of the polarity of sentiment, which excludes neutral examples (we will add the neutral class for the tonality y in the next task definition).

Task Definition II: Polarity of Sentiment. *In this context, Opinion Mining has the task to determine the polarity of sentiment y' for a given statement s, consisting of s_n words:*

$$t_1 : s = (w_1, w_2, ..., w_{s_n}) \mapsto y' \in \{\text{positive,negative}\} \tag{6.1}$$

So, the focus is on the statement level and not on the document level. Other research in this area supports the position that the document level is not suitable for Opinion Mining in newspaper articles [BSK+10].

For many reasons, we restrict ourselves to non-neutral statements at the beginning.

First of all, neutral statements are not on the borderline of a binary classifier [KS06a] and present a challenging problem themselves (also known as Subjectivity Analysis). The literature states that this requires other purposeful techniques [AWCM11]. Later we want to explore especially this last point. Moreover, not all methods of this chapter are designed for a classification, which includes neutral examples, and so any comparison would be misleading. Finally, a trend analysis (differentiation between positive and negative statements) would be an interesting stand-alone system for media monitoring, because it can be used as an early warning system for monitoring services: The system can raise the alarm in the case of a strong change in the polarity or if too many negative examples surface.

The main issues addressed in this chapter are: We describe the big challenges in this area and what are the specifics of the news domain for Opinion Mining (in contrast to Opinion Mining in customer reviews, e.g.). We talk about these characteristics especially in our pre-evaluation (section 6.3) and later in section 6.5 about our linguistic features. Then we explain and evaluate conventional as well as completely new methods to determine the polarity of sentiment in statements. Furthermore, we introduce linguistic features which can increase the accuracy for Opinion Mining in newspaper articles in section 6.5. Finally, we show the limits of these approaches in section 6.6, when we integrate neutral statements. But first we start with the background for the creation of sentiment dictionaries and Opinion Mining in news in the next section.

6.2 Background for the Creation of Sentiment Dictionaries and Opinion Mining in News

For the creation of sentiment dictionaries, many approaches [PLV02, KK07, DTCY10, BWC11] in the product review context use a collection of annotated data (reviews with ratings) to collect important sentiment keywords. While some of the approaches also extract verbs, nouns, and adverbs [BWC11], most of the attention is given to the adjectives [KK07, HPD+08]. One solution in this area involves the construction of powerful and partly very complex adjective patterns to handle sentiment polarity and the relationship between products and adjectives [BWC11, HPD+08]. Some approaches take only single words as unigrams, some expand their lists with bigrams [BWC11].

As we have seen in section 3.2, approaches in social media such as [PP10] collect positive and negative sentiments by a procedure which determines the sentiment of emoticons (smilies) used in the tweet. Unfortunately, news items do not contain significant icons such as emoticons.

Weighting methods for different sentiment-bearing words (in a dictionary, e.g.) come from different areas: Kaji and Kitsuregawa [KK07] have compiled a lexicon from

Japanese HTML documents. Their idea of creating adjectival phrases is unfortunately not completely adaptable due to particularities of the Japanese language and differences in context, but their methods of selecting phrases and especially words are interesting and are treated in section 6.4. Harb et al. [HPD+08] extract opinions from blogs and they use association rules for the extraction of suitable adjectives.

In contrast, other approaches avoid to construct their own dictionaries from training data and use general sentiment lexicons such as SentiWordNet [BES10] instead. We investigate the question, if a general lexicon produces sufficiently good results. Thus we apply the already mentioned dictionary SentiWS [RQH10] (cf. section 3.5.2).

In the news domain, we have already observed that many approaches on this topic only work with reported speech objects [BSG+09, BSK+10, PLS11], because news articles are less subjective [BSK+10]. But quotations in news items are often text parts where opinions and generally more subjective text can be found [BSK+10]. In addition, the opinion holder (the speaker of a reported speech object) can be identified and extracted in most cases and sometimes even the target of the opinion [BSK+10] (an entity such as another person or an organisation, for instance). However, only opinions, which are part of a reported speech object, can be found and analysed by this method. If we examine 3,120 statements of our Finance data set (the firstly analysed set of statements of our evaluation in section 6.4.4), only 21,25% of the statements contain a quotation and less than 5% have a proportion of quoted text larger than 50% of the whole of the statement. As a consequence, between 78% up to 95% of the statements could not be classified by this technique. Moreover, the focus of Park et al. [PLS11] is the extraction of groups that have different opinions and they do not estimate the sentiment of the reported speech itself from a neutral point of view. They identify speakers who agree or disagree with other speakers or organisations. Thus, these methods are not suitable for our tasks.

6.3 Pre-Evaluation

In our pre-evaluation, we show some specifics of our domain. At first we validate which word classes are the most important ones for Opinion Mining in newspaper articles. Our data set consists of 1,596 statements including 796 positive and 800 negative statements of the Finance dataset (we use a small part of it for this first evaluation). We extract the Part-Of-Speech-Tags (POS-Tags) from the statements. We use a TF-IDF matrix $\omega_{w,y'}$ [Jon72] in an adapted version to weigh the terms in the four classes: nouns, verbs, adjectives, and adverbs, which are the important categories for the polarity [BWC11, RQH10].

$$\omega_{w,y'} = \mathrm{tf}_{w,y'} * \mathrm{idf}(w) = f(w, y') * log\frac{N}{n_w} \tag{6.2}$$

In this equation $\mathrm{tf}_{w,y'}$ or $f(w, y')$, respectively, is the frequency of term w (we always use the lemma of the word) with the polarity of sentiment y', N is the number of all statements and n_w is the number of statements containing term w.

In addition, we test different classification techniques for a first overview. We use 20% of the data for training, and 80% for testing. As has been pointed out, Support Vector Machines (SVMs) and Naive Bayes are commonly used for Opinion Mining and Text Mining purposes in general. We apply also the clustering technique k-means. It creates two clusters and identifies the majority of the two classes within each cluster by the training set. The majority estimates the classes of the test set.

Table 6.1 illustrates the classification accuracies (cf. equation 6.17) of different machine learning techniques and word categories. As table 6.1 shows, the most important words do not belong to the word category of adjectives, which is a common assumption in Opinion Mining approaches which deals with customer reviews. In newspaper articles the verbs and nouns do play a more important role.

If we take a close look at examples, this behaviour is understandable. Typical sentences of a customer review are "The zoom is great." or "So, overall a great camera for the price."(taken from amazon.co.uk on 20th July 2011). Examples of newspaper articles like "Eight banks fail stress tests." or "Analysts fear the end of the eurozone." make clear that the polarity of the sentiment is created in a different way. Here, the verbs "fail" and "fear" and the noun "end" are the words which create the polarity. Note that nouns perform better than verbs by most methods of classification, but their appearance is three times more frequent than the appearance of verbs (cf. section 6.4.4).

As a result we create noun-based patterns (triplets) and verb-based patterns as new methods analogically with adjective-based patterns for reviews.

Category	SVM	Naive Bayes	Decision Tree	k-NN	k-means	Linear Regression
Adjectives	61.76%	61.44%	54.39%	52.82%	50.60%	49.76%
Verbs	73.67%	72.10%	67.01%	52.27%	51.16%	49.76%
Nouns	77.27%	70.06%	69.36%	55.72%	55.55%	77.51%
Adverbs	61.44%	61.60%	59.64%	51.65%	50.91%	59.95%

Table 6.1: Evaluation of the different word classes and learning methods.

6.4 Determination of Polarity

Turning to the determination of polarity, we examine many different methods for the estimation. Our procedure describes the following: For the word-based methods, we construct sentiment dictionaries concerning statistics over the appearance of words in positive and negative statements. Here, we use different weighting methods to calculate a polarity value $\sigma_{method}(w)$ for a word w. As in the pre-evaluation, the most important categories are nouns, verbs, adjective, and adverbs. We calculate four polarity values for one statement in order to use these values as input features for a classification process. The procedures of the lexicon-based method, the bigrams, and the pattern-based action chains operate analogically. Nevertheless, we do not create sentiment dictionaries in fact for these kinds of methods, but collections of bigrams or pattern-based action chains, for example, and the methods calculate the polarity values differently. However, we start with the section about word-based methods.

6.4.1 Word-based Methods

For the word-based methods, we compute a sentiment score σ_{method} for each single word. The words belong to the four word categories (adjectives, verbs, nouns, adverbs) and we divide the influence of each important category into one single score. Thus, we get four sentiment scores for one statement. Every score is the average of the sentiment scores in one category: The first feature is the average of the scores of all the statement's adjectives ($\sigma_{Adj}(s)$), the second of all nouns ($\sigma_{No}(s)$), the third of all verbs ($\sigma_{V}(s)$), and the fourth of all adverbs ($\sigma_{Adv}(s)$).

$$\sigma_{cat}(s) = \frac{1}{|s_{cat}|} \sum_{w \in s_{cat}} \sigma_{method}(w) \tag{6.3}$$

Here, s_{cat} are only the words in statement s which belong to one of the four important categories (adjectives, nouns, verbs, and adverbs) and σ_{method} refers to one of the five methods based single words or it is the score of the word in SentiWS [RQH10] or the score of the bigram (the score $\sigma_{cat}(s)$ would be zero for $|s_{cat}| = 0$).

Chi-square

Kaji and Kitsuregawa [KK07] use the chi-square value for a polarity value. We have adapted their method for our evaluation. First, they calculate the probability of the appearance in negative and positive statements for each candidate.

$$P(w|pos) = \frac{f(w, pos)}{f(w, pos) + f(\neg w, pos)} \qquad P(w|neg) = \frac{f(w, neg)}{f(w, neg) + f(\neg w, neg)} \tag{6.4}$$

$f(w, pos)$ is the frequency of the candidate word w in positive statements and $f(\neg w, pos)$ is the same for all candidates words without w. The higher probability sets the polarity of the score value:

$$\sigma_{\chi^2}(w) = \begin{cases} \chi^2(w) & \text{if } P(w|neg) < P(w|pos) \\ -\chi^2(w) & \text{otherwise} \end{cases} \tag{6.5}$$

The score itself is given by the statistical measure chi-square value. Here, the following thesis is assumed: The null hypothesis says that every word appears in positive statements with the same probability as in negative statements.

$$\chi^2(w) = \sum_{x \in \{w, \neg w\}} \sum_{y' \in \{pos, neg\}} \frac{(f(x, y') - \hat{f}(x, y'))^2}{\hat{f}(x, y')} \tag{6.6}$$

In this equation, $\hat{f}(x, y')$ is the expected value when the null hypothesis is supposed.

Pointwise Mutual Information

The tonality based on pointwise mutual information (PMI) [CH89, Tur02, KK07] uses the strength of the association between the word w and positive and negative statements, respectively ($y' \in \{pos, neg\}$).

$$PMI(w, y') = log_2 \frac{P(w, y')}{P(w)P(y')} \tag{6.7}$$

It is possible to calculate a polarity score based on the PMI, which is based on the difference between $PMI(w, pos)$ and $PMI(w, neg)$ [Tur02, KK07].

$$\sigma_{PMI}(w) = PMI(w, pos) - PMI(w, neg) = log_2 \frac{P(w, pos)/P(pos)}{P(w, neg)/P(neg)} = log_2 \frac{P(w|pos)}{P(w|neg)} \tag{6.8}$$

Association Rule Mining

Harb et al. [HPD+08] propose Association Rule Mining for polarity, so rules must be found: the word determines polarity. It needs minimum support (word w appears in x_1 of all cases with the polarity y') and a minimum confidence (word w appears in x_2 of the statements containing w with the polarity y').

$$\text{support}(w, y') = \frac{f(w, y')}{N} \geq x_1 \qquad \text{confidence}(w \rightarrow y') = \frac{f(w, y')}{f(w)} \geq x_2 \tag{6.9}$$

$f(w, y')$ is the number of statements, in which word w and polarity of sentiment y' appear at the same time, and N is the number of all statements. $f(w)$ is the number of statements which contain w. If the word w fulfils both conditions, the word w get the value +1 and -1, respectively.

Information Gain

Another opportunity is the use of the information gain for our polarity value. For this purpose, we designed a new weighting method. The information gain is based on the entropy [Sha48] of a set of statements S which contains positive and negative statements.

$$\text{entropy}(S) = -1 * [(P(pos) * log_2(P(pos)) + P(neg) * log_2(P(neg))] \tag{6.10}$$

If one word is chosen, then the gain of purity can be calculated by considering the two sets $S_{w,pos}$ and $S_{w,neg}$ in which w appears.

$$\sigma_{IG}(w) = \begin{cases} 1.0 - \sum_{y' \in \{pos,neg\}} \frac{|S_{w,y'}|}{|S|} * \text{entropy}(S_{w,y'}) & \text{if } P(neg|w) \leq P(pos|w) \\ -1.0 + \sum_{y' \in \{pos,neg\}} \frac{|S_{w,y'}|}{|S|} * \text{entropy}(S_{w,y'}) & \text{otherwise} \end{cases} \tag{6.11}$$

For this score, we define entropy($S_{w,y'}$) as (cf. also equation 6.13):

$$\text{entropy}(S_{w,y'}) = -1 * P(y'|w) * log_2(P(y'|w)) \tag{6.12}$$

Entropy

We have designed a polarity score based on the entropy [Sha48]. First we calculated the probability for a positive and a negative statement, if we observe w in a statement.

$$P(pos|w) = \frac{f(w, pos)}{f(w, pos) + f(w, \neg pos)} \qquad P(neg|w) = \frac{f(w, neg)}{f(w, neg) + f(w, \neg neg)} \tag{6.13}$$

$f(w, \neg pos)$ is the frequency of w in negative statements. If the probability for word w to be observed in positive statements is equal to or higher than the probability for word w to be observed in negative statements, then the score would be positive. We normalize the scores to [-1,1] by adding one or minus one.

$$\sigma_{ENT}(w) = \begin{cases} 1.0 + P(pos|w) * \log_2(P(pos|w)) & \text{if } P(neg|w) \leq P(pos|w) \\ -1.0 - P(neg|w) * \log_2(P(neg|w)) & \text{otherwise} \end{cases} \tag{6.14}$$

Lexicon-based

Many approaches [DLY08, WWH09, TBT+11], also in the news article context [DA07, BSG+09], benefit from sentiment lexicons such as SentiWS [RQH10]. Its sources include an English lexicon (translated into German), a large collection of product reviews, and a specialised German dictionary. All sources are used to define and improve a sentiment score based on the PMI method [CH89]. The English lexicon is the General Inquirer (GI) lexicon [SDSO66], in which words are listed either in a positive or a negative category. Both lists are translated into German using Google Translate[1]. Some missing words such as "insolvency" are added. Then they perform a co-occurrence analysis with the collection of product reviews. These words, which are significant for a positive or a negative review, respectively, are candidates for a manual selection.

Finally, they find additional words by looking up the collected words in the specialised German dictionary. In this dictionary, the two groups related to sentiment and not related to sentiment show new candidates based on the words of the first two steps. In that way, they obtain 1,650 positive and 1,818 negative words in lemma, which cover 15,649 different positive and 15,632 different negative forms in total. Analogue to Turney and Littman [TL03], they calculated the PMI score between the words and eleven positive or eleven negative words, respectively, based on a corpus of 100 million sentences [RQH10]. We apply this dictionary in the same way as we use our own created dictionaries, which calculate the scores based on the statistical methods explained before.

6.4.2 Bigrams

Previously, all methods compute a score based on a single word. Now, we want to introduce the first method which uses more than one word: two word sequences, also known as bigrams. The number of two word sequences can get very large in news and even in statements. On the contrary, we have to keep the size of the dictionaries (or the list of bigrams) small, because media analysts were scheduled to check the entries of the sentiment dictionaries in the practical solution (this intention was later changed, since the verification of the analysts was attached less importance). So, we propose an algorithm which uses bigrams, if they contain more information than single words.

[1] http://translate.google.com

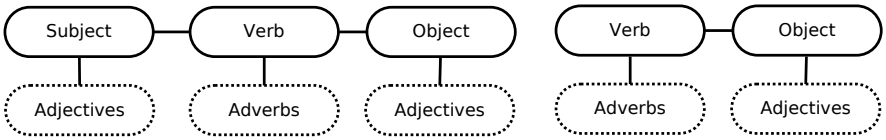

Figure 6.1: On the left side a triplet and on the right side a verb-based pattern.

The proposed algorithm tries to find two word sequences which belong to the word categories adverbs, adjectives, verbs, nouns, negation particles, or entities like persons or organisations. The approach uses the entropy to determine the quality of a certain word. If this word is not straight-forwardly positive or negative (and would therefore get exactly a +1 or -1 value in the word-based method), the algorithm searches all bigrams to which the word has a part-of relationship. If the entropy value is smaller, when the single word is replaced through all bigrams which contain this word, then the tonality scores of these bigrams are used. Thus, a word like "`loose\V`" which might occur in both types of statements, can be replaced by "`not\NEG loose\V` (positive)" and "`loose\V influence\NN` (negative)", "`loose\V strongly\ADV` (negative)".

The score itself can be calculated by all methods mentioned above. We use the entropy method. Also, we distribute the influence of the four categories on the four features: The first one is the average about all bigrams, which contain at least one adjective and so on.

6.4.3 Pattern-based Action Chains

In this section, the methods try to capture the actions of persons, organisations, etc. in the sentences and need more than one or two single words to determine a polarity value.

Triplets

In our context, these are models for word combinations in texts which are based on the grammatical structure of sentences. A triplet consists of the three basic elements: subject, verb, and object. In English, it is easy to extract these structures. In other languages such as German, you need a case-tagger to identify subject and object, for instance. Triplets are used in Information Retrieval [DRF+09] as well as in Sentiment Analysis tasks [KIM+04].

With this extraction of triplets we try to obtain the actions in the texts (which are also called action chains in literature [BHMM11]). Our triplet model does not only extract the three basic pieces, but also the adjectives, which belong to subjects and objects, and the adverbs, which belong to verbs (as shown in figure 6.1). The work of Balahur et al. [BHMM11] suggests that the inclusion of adjectives and modifica-

tions through adverbs in our action chains may be useful, because their error analysis complains the missing adjectives and adverbs in their action chains.

Verb-based Patterns

We have seen (cf. section 6.3) that verbs perform very well for the polarity classification. So, we designed a second class of action chains, which is focused on verbs. The method considers all full verbs in the first step. The TreeTagger [Sch94, Sch95] tags full verbs with \VV. In the next step, all objects (proper nouns, entities) will be extracted with the verb, if the distance of the object and the verb is lower than or equal to x words. Therefore, the pattern regards the x words before and after the verb. Likewise, the pattern includes the adjectives belonging to the objects and the adverbs belonging to the verbs (as shown in figure 6.1).

Similarity Functions for Action Chains

To find similar examples of triplets and verb-based patterns, we have designed two different similarity functions. The first one is based on the Jaccard similarity coefficient [Jac01] and the lemmas of the different elements and the second one is based on polarity values of the elements.

$$sim(c_1, c_2) = \frac{n(c_1 \cap c_2)}{n(c_1 \cup c_2)} \tag{6.15}$$

The lemmas of each element are compared to the corresponding element of the action chain by the first similarity function. If two elements have the same lemma, their similarity is one, otherwise zero. Thus, $n(c_1 \cap c_2)$ is the number of matching elements and $n(c_1 \cup c_2)$ the number of all elements. If one action chain has got more elements (through more adjectives or adverbs), $n(c_1 \cup c_2)$ is the number of the smaller chain.

The second similarity function does not compare the lemmas of the elements, but compares the polarity values of the lemmas w_1, w_2.

$$\delta(w_1, w_2) = 1 - |\sigma_{method}(w_1) - \sigma_{method}(w_2)| \tag{6.16}$$

For the tonality scoring method σ_{method} every word-based approach above can be used. This variant is more motivated by a linguistic question: Do patterns lead to a prediction about the polarity? For example, patterns such as a negative verb ("lose") with a positive subject or object ("income") could lead to a negative polarity. We investigate this question by this similarity score.

The best fitting chain of a training set delivers the value for a new one and the final score of a statement is the sum of all values from the containing chains.

Category	Δ1600	Δ1100	SentiWS	Δ600
Adjective	927	732	1,482	500
Adverb	186	164	10	134
Noun	2,663	2,011	1,270	1,315
Verb	805	634	742	450
Total	4,581	3,541	3,504	2,399

Table 6.2: The different sizes of the created dictionaries and SentiWS.

6.4.4 Evaluation

Description of the Experimental Setup

Our test corpus consists of 5,500 statements (2,750 are positive and 2,750 are negative) from 2,097 different news articles from the Finance dataset (cf. section 4.5).

We split the corpus into a part consisting of 1,600 statements (800 positive, 800 negative) for extracting dictionaries, triplets, verb-based patterns, or bigrams. Chronologically speaking, the first statements of the dataset are applied for the extractions, because this would also be the only available data in a practical solution. We obtained 4,581 words, 4,816 triplets, 4,809 verb-based patterns, and 6,924 bigrams. We have also constructed smaller collections of 600 and 1,100 statements (called Δ600 and Δ1100) from this set for our comparison with SentiWS [RQH10]. The Δ1100 collection generates a dictionary which has got approximately the same size as the SentiWS (see table 6.2). We need only the lemmas of SentiWS.

The comparison with SentiWS is interesting, because the collected words of our dictionaries are influenced by the context. The results of SentiWS are supposed to show, if a context-independent dictionary achieve sufficient results.

Table 6.3 also shows some examples of our dictionary based on 1,600 statements: Typical examples such as “benefit” and “drop” or “economic” and “known”, which are surprising entries at first sight. The examples are translated into English here and the given values are entropy-based.

After that we split again the second part into two parts. The first part of 780 statements (split ratio 0.2) is used as a training set for machine learning and the larger second part is our test set. The approach implies an SVM which gets the four different tonality values as attributes for learning. We also construct smaller training sets of 390 (called S-0.1), 195 (S-0.05) and 39 statements (S-0.01). As in the pre-evaluation, SVMs perform better than Naive Bayes et cetera. Our SVM uses the RapidMiner[2]

[2]Rapid-I: http://rapid-i.com/

Word	Category	Pos	Neg	Value
benefit	Verb	15	10	0.56
drop	Verb	0	15	-1.0
investment	Noun	184	121	0.56
very	Adverb	17	21	-0.53
known	Adjective	3	11	-0.73
economic	Adjective	3	33	-0.87

Table 6.3: Examples of the Δ1600 dictionary with entropy values.

standard implementation with default parameters. But we force the SVM to balance between precision and recall in and between the two classes because Media Response Analyses are usually not as balanced as our test set.

For our balanced test data we use the accuracy evaluation metric

$$ACC = \frac{c}{n} \tag{6.17}$$

where c are the correctly predicted statements in the test set and n the number of all statements in the test set.

Results of the Different Methods

The table 6.4 shows the results. The best performing methods (cf. table 6.4 left) are our entropy and information gain-based methods with about 70%. At the same time the performance of the chi-square method is good, PMI and TF-IDF are performing averagely. The pattern-based action chains (triplets, verb-based patterns) are performing on a lower level as the two entropy-based methods and the bigrams. Likewise, the polarity-based methods of triplets and verb-based patterns achieve nearly the same results. The bigrams obtain almost the same results as the information gain and entropy-based method.

The entropy-based method remains very stable even if the training set is decreased (table 6.4 up right). Furthermore, the entropy and information gain methods have over 7 % resp. 9 % higher performance than SentiWS using the Δ1100 set (table 6.4 bottom right). There are two possible explanations. As Remus et al. [RQH10] report: They choose the PMI method without evaluating other methods, so an implementation of our methods could increase their results. A second reason could be the construction of SentiWS. They rely on customer reviews to expand their dictionary. But this evaluation shows that Opinion Mining in customer reviews and in news differ. So, resources based

Method	Accuracy
Word-based Methods	
TF-IDF	0.6353
Chi-square	0.6837
PMI	0.6404
Association Rule Mining	0.5234
Entropy	**0.7006**
Information Gain	0.6955
Bigrams	0.6888
Action Chains	
Triplet	0.6580
Triplet(polarity-based)	0.6567
Verb-Based Pattern	0.5925
Verb-Based Pattern(pol.-based)	0.5925

Category	S-0.01	S-0.05	S-0.1	S-0.2
Adjective	0.3831	0.6224	0.6236	0.6215
Adverb	0.5804	0.5725	0.5855	0.5987
Noun	0.6674	0.6659	0.6641	0.6625
Verb	0.4996	0.6499	0.6464	0.6474
Noun&Adj.	0.3825	0.6729	0.6741	0.6718
Noun&Verb	0.6713	0.6826	0.6804	0.6808
All	0.5872	0.6804	0.6940	**0.7006**

Method	Δ600	Δ1100	Δ1600
Chi-square	0.6292	0.6548	0.6837
PMI	0.6196	0.6224	0.6404
Entropy	**0.6628**	0.6805	**0.7006**
Information Gain	0.6558	**0.6952**	0.6955
SentiWS (1 size)	-	0.6036	-

Table 6.4: Left: Comparison between the different methods. Up right: The different word classes (entropy method) by different sizes of the training set. Bottom right: Comparison to the SentiWS [RQH10] and different sizes of the created lexicons.

on customer reviews seem to be not suitable for Opinion Mining tasks in news items. Hence, a general sentiment dictionary alone does not solve our task sufficiently enough. But we will reuse it for comparison purposes or as a context-independent dictionary. As another consequence of the differences between reviews and news, the noun and adjective combination (table 6.4 up right), which is a strong compound for Opinion Mining in reviews, does not perform as well as all four important categories.

6.4.5 Conclusion for the Next Steps

The results show that our word-based methods outperform complex pattern-based approaches. On the one hand, this is an advantage for the performance of such a system because case-tagging is a very time-consuming process, for example. On the other hand, the findings are surprising because the human way of understanding and rating statements seems to consider combinations of words and does not summing up a value for each single word. But the problem is that journalists tend to write their texts in many different formulations, so that it is very difficult for a pattern-based approach to collect enough examples or bigram combinations to model all possible phrasings. So, this is probably not suitable for Opinion Mining in news articles.

We expected more from the bigrams in particular, because other approaches work well with bigrams. Nevertheless, Sarvabhotla et al. report [SPV11], that their unigram

approach works better than their bigram approach. However, the manual maintenance of a bigram solution is not easy, because the number of entries gets much larger and humans can estimate single words more easily than bigrams without context. Thus, we do not try a trigram approach.

As a consequence for our next steps, we will try to capture modifications like negations as an expansion of our existing word-based features. Again, we try to tackle this problem by a machine learning-based method (the later integration of features for the linguistic context was one of the reasons to create a method based on polarity scores for the determination of the polarity of sentiment). In the next section, we integrate features to analyse the linguistic context of sentiment-bearing words.

6.5 Integrating Linguistic Features

6.5.1 Introduction

As discussed earlier, opinions are not stated so clearly in newspaper articles [BSK+10] as, for example, in a customer review or a social media contribution (where additional pieces of information such as emoticons can be analysed [PP10]). In the news, some special features are important for the polarity of sentiment, so that a word-based method cannot solve this problem without analysing the (linguistic) context. For instance, the following negative statement from the pressrelations dataset [SCH12] contains negative words and positive words, too.

(6.1) The Higher Education Pact was and still is politically right. But deficiencies appear in several areas, which will exacerbate in the next few years. We call Federal Minister Schavan to take the initiative and to support the federal states accepting the challenges. (**Code:** negative, CDU)

Here, the conjunction "but" is the devise factor that the negative words are more important for the polarity of the sentiment. We give many examples with other linguistic features in the section 6.5.3.

In this section, we explain linguistic features which improve our methods for the polarity of sentiment, which we introduced in the last sections. For this purpose, we have created two new kinds of feature sets. The first set stands for the effect of linguistic factors (negations, conjunctions, etc.) and is called the **Linguistic Effect Features**. The second describes which opinion-bearing words are influenced (**Linguisticly Influenced Polarity Features**). We call the already explained features **Basic Polarity Features** and this set represents the polarity values of adjectives, nouns, verbs, and adverbs ($f_{\alpha_1}(s) = \sigma_{Adj}(s)$, $f_{\alpha_2}(s) = \sigma_{No}(s)$, $f_{\alpha_3}(s) = \sigma_{V}(s)$, and $f_{\alpha_4}(s) = \sigma_{Adv}(s)$) which we have already explained in section 6.4.

The rest of this section contains the following: In section 6.5.2, we analyse Related Work on linguistic features. In section 6.5.3, we present our linguistic features. After that we evaluate our methods on our two datasets, before we give a conclusion for our future steps.

6.5.2 Background on Linguistic Features

Analysing the linguistic context is one method of resolution for Opinion Mining. Many publications [CC08, WWH09, TBT+11] in this area refer to Polanyi and Zaenen [PZ06], whose contribution is a very strong theoretical work about contextual valence shifters. They use the term "valence" such as other publications speak of the sentiment [BWC11, DTCY10], the polarity [KK07] or the orientation [DLY08, TBT+11]. They start with simple lexical valences (positive and negative valences of verbs, nouns, adjectives, and adverbs) and expand their theoretical approach with negations, intensifiers, modals, presuppositional items (such as "barely", "even"), connectors, discourse structure, multi-entity, the genre, reported speech, subtopics, genre constraints, and cultural constraints. They provide sophisticated ideas, but the implementation of some ideas is very difficult or not possible with recent NLP tools (the genre constraints are very hard to estimate, for example). Nevertheless, much research operates according to this wealth of ideas. For instance, their ideas seem to be a blueprint for SO-CAL [TBT+11], which implements or adapts many of their ideas.

Methods such as Opinion Observer [DLY08] or SO-CAL [TBT+11] try to handle implicit or contextual sentiment such as negations. Negations as the maybe most important implicit factor are often treated by heuristic rules [CC08, DLY08], which reverse the polarity of sentiment words. A well-known paper [WBR+10] deals with negations in Sentiment Analysis and compares different approaches in different tasks and even different languages are treated. Unfortunately, they treat all these issues more or less in a theoretical way and do not perform an own evaluation with practical tests. Interesting techniques for the effects of negations have been introduced by Jia et al. [JYM09]. Here, the scopes of negations are derived from different rules. These ideas will be taken up later.

Some papers [BWC11, DTCY10] deal with the domain-specific context of Sentiment Analysis. They collect customer reviews about books, DVDs, electronics, or kitchen appliances. Their intention is the construction or the adaptation of sentiment dictionaries for certain contexts. But in an MRA the topics can be so numerous, so insignificant and so diverse, that we are more interested in a linguistic and grammatical context as in Zhou et al. [ZLG+11]. They show that conjunctions can be used to avoid ambiguities within sentences.

6.5.3 Linguistic Features for Machine Learning

In the following, we explain our linguistic features and their integration into a solution based on machine learning. First, we present our idea of measuring both: the presence of effects and the affected polarity. This means that we analyse which effects occur in the statement and at the same time which sentiment words are affected and how they are affected. Afterwards, we explain the different features in detail. In addition, we present an example statement of our real world dataset (the Finance dataset, see section 6.5.4) for every feature.

Effect Measurement

We propose two techniques for measuring the effect of linguistic sentiment modifications. The first technique only measures, whether or not the linguistic effects are present in a given statement and stores it as one feature for every aspect (**Linguistic Effect Features** β). In this way a machine learning approach can recognize these effects and train its classification model with the additional information (cf. section 6.5.3).

The second technique tries to capture an area of this effect (which in itself is in some cases a very problematic question) and it takes the polarity of the area as the feature value for this aspect (resulting in **Linguistically Influenced Polarity Features** γ). For example, a statement has positive and negative words, but the modals only reduce the effect of negative words (as in section 6.5.3) or the effect of some positive words like in the following example of the pressrelations dataset [SCH12]:

(6.2) Finish the zig-zag course of Merkel's government. Germany needs a master plan for the energy turnaround and for the support of future technologies. The energy turnaround can only succeed with a long-term plan, in which all concerned actors of politics, companies, environmental groups, and unions participate. (**Code:** negative, CDU)

This is a very difficult example, because it contains very few negative words in contrast to positive words (such as "support" or "succeed"). This makes it all the more important to recognize modals and their effect. The modal "can" reduces the positive polarity of the word "succeed".

The feature value is the average of the polarity of the influenced words (this normalization by the average is not absolutely necessary; our machine learning techniques could also scale the feature values). We implement techniques from Jia et al. [JYM09], who are trying to capture different effect areas for negations. We adapt their *candidate scope* and *delimiter rules* [JYM09] using static and dynamic delimiters for the German language and expand them also for our non negation features: The static *delimiters*

[JYM09] remove themselves and all words after them from the scope. Static *delimiters* are words such as "because", "when" or "hence" [JYM09]. A *conditional delimiter* [JYM09] becomes a delimiter if it has the correct POS-tag, is inside a negation scope, and leads to opinion-bearing words. Examples are words such as "who", "where" or "like". The scope for the conjunction "but" in example 6.1 would be "deficiencies appear in several areas" and the scope for the modal "can" in example 6.2 would be "only succeed with a long-term plan".

In addition, we have designed a second method which is simpler. It creates a scope around an effect word. All words in the scope have a smaller distance to all other effect words (in number of words between them). In the simple sentence "John is nice, but they do not like his bad company." the method would associate "nice" with "John" and "bad" with "company". This method helps, for example, in assigning adjectives to the entity which they belong to and is used in the first of the following types of features.

Type of Entities

The first two features indicate whether the statements refer to persons or to things (such as products, organisations, or companies). The sentiment of words can change depending on whether the statements concern persons or companies, for example.

(6.3) The old traditional company XY will offer no more certificate products in future.

(6.4) Many employees like John Blogg, fund manager, are old by contrast to other banks. This evokes challenges in recruitment.

In the first example the word "old" has a positive polarity, whereas the sentiment of the word is negative in the second example. The well-disposed reader will maybe find quickly an example, which mentions the word "old" in a context with a person and has a positive polarity. But this is not our point: Here, we will try to find a separating polarity in a concrete context of an entity. And the word "old" has a more negative polarity for persons in the Finance dataset, for example. So, the first two features represent the proportion of persons and organisations:

$$f_{\beta_1}(s) = \frac{p(s)}{p(s) + o(s)} \qquad f_{\beta_2}(s) = \frac{o(s)}{p(s) + o(s)} \tag{6.18}$$

In equation 6.18 for the first two β features, $p(s)$ and $o(s)$ are the number of persons and organisations, respectively, in the statement s. If a statement do not contain any person and organisation, both features would be zero.

Conjunctions and Polarity Values

word	ν_c	word	ν_c	word	ν_c	word	ν_c
whereas	-0.5	as well	1.0	but	-1.0	or	0.5
however	-0.5	though	-1.0	and	1.0	by	1.0

Hedging Words

can	may	could	might
would	shall	should	ought to
will	must		

Table 6.5: Left: Conjunctions and polarity values. Right: Hedging auxiliary verbs.

$$f_{\gamma_1}(s) = \frac{1}{|P_w|} \sum_{w \in P_w} \sigma(w) \qquad f_{\gamma_2}(s) = \frac{1}{|O_w|} \sum_{w \in O_w} \sigma(w) \tag{6.19}$$

For the two type γ features, P_w and O_w are the sets of words which belong to persons' and organisations' scope, respectively (cf. previous section). All word-based methods (cf. section 6.4.1) can be applied for the σ function (more precisely, it should be entitled as σ_{method}, but we use the abbreviated form in this section).

Negation

Many approaches in Opinion Mining indicate that the effect of a negation is important [CC08, DLY08, JYM09, WWH09, TBT+11]. Thus, our approach also has features for the presence and the effect of a negation in a statement.

(6.5) "We think that it is not sensible to accept lawsuits for years", said John Blogg of the banking company XY.

$$f_{\beta_3}(s) = \begin{cases} 1.0 & \text{if } \exists w \in s : w \text{ is a negation} \\ 0.0 & \text{otherwise} \end{cases} \qquad f_{\gamma_3}(s) = \frac{1}{|N_w|} \sum_{w \in N_w} \sigma(w) \tag{6.20}$$

The negation feature shows, whenever a negation is present in statement s. N_w are the affected words. At this point, the area of affected words is determined by the *candidate scope* and *delimiter rules* [JYM09].

Conjunction Polarity

The use of conjunctions can also indicate a polarity and this fact is made use of in other work. The early contribution of Hatzivassiloglou and McKeown [HM97] concerns the conjunctions "and", "or", and "but" to predict the polarity of an adjective's sentiment through co-occurrence and uses bootstrapping to collect new words. However, there are also other conjunctions which can contain an implicit sentiment.

(6.6) A profit of nine percent in the last twelve months does not sound bad, however company XY is running behind the competition with such a result.

Here, the conjunction "however" implies a negative sentiment and the opposite polarity, in general. It expresses a contrast, while conjunctions such as "and" or "as well" express a support. We create a test dataset of 1,600 statements of the Finance dataset, collect the conjunctions and associate them with a polarity value ν_c by their appearance in positive and negative statements. For this, we have defined five classes, so every conjunction gets a value from -1.0 to 1.0 (often appears in negative statements $\rightarrow \nu_c =$ -1.0, appears a little more often in negative $\rightarrow \nu_c =$ -0.5, equal $\rightarrow \nu_c = 0.0$, and so on). Table 6.5 (left) shows the different conjunctions and their values to influence the polarity. It shows also that contrasts appear more often in negative statements and a support is expressed more often in positive statements.

$$f_{\beta_4}(s) = \frac{1}{|C_s|} \sum_{c \in C_s} \nu_c \qquad f_{\gamma_4}(s) = \frac{1}{|C_w|} \sum_{w \in C_w} \nu_c * \sigma(w) \tag{6.21}$$

The type β feature for conjunctions is the sum of all polarity values ν_c of all conjunctions C_s of the statement s. The conjunction influenced words are C_w. The scope is determined by the *candidate scope* and *delimiter rules* [JYM09], but only words after the conjunction are concerned because the conjunction itself is a delimiter. The multiplication with ν_c indicates which type of conjunction influences the affected words. If the conjunction expresses a contrast (e.g. "but" with $\nu_c = -1.0$), the polarity of the words will be inverted. A conjunction such as "as well" ($\nu_c = 1.0$) will keep the original polarity.

Quoted Text

The next aspect under investigation is the proportion of quoted text in a statement. It is important to analyse this proportion for two reasons. On the one hand a short part of quoted text can be a hint for irony in written texts [CSSdO09].

(6.7) "Array of products optimisation." So the company calls the closing and merger of many funds.

On the other hand, a long part can stand for a reported speech object. In a reported speech object, a person gives his/her opinion in his/her point of view, which in most cases supports the overall polarity of a statement.

(6.8) "As a consequence the share prices will be on the decrease and this will put the pressure on the finance systems", said he.

As a result, a machine learning approach can better differentiate between irony and reported statements, if the length and the affected words of quoted text are measured.

$$f_{\beta_5}(s) = \frac{l(q(s))}{l(s)} \qquad f_{\gamma_5}(s) = \frac{1}{|Q_w|} \sum_{w \in Q_w} \sigma(w) \tag{6.22}$$

$q(s)$ is the part of a statement s, which appears in quotation marks. $l(x)$ is the length (in characters) of a text x. Q_w are the words inside a quotation.

Modals

Modal verbs like "can" or "would" can weaken the strength of the polarity. The method analyses how the statement is affected by hedging expressions.

(6.9) A loss could impend, if the subsidiary company would cleave to its strategy. So, the administrative management reconsiders the investment.

The full list of auxiliary verbs for hedging expressions is shown in table 6.5 (right).

$$f_{\beta_6}(s) = \frac{h(s)}{v(s)} \qquad f_{\gamma_6}(s) = \frac{1}{|H_w|} \sum_{w \in H_w} \sigma(w) \tag{6.23}$$

The method counts how often full verbs are influenced by hedging expressions $h(s)$ in comparison to all full verbs $v(s)$. H_w is the set of words affected by hedging. Here again, the *candidate scope* and *delimiter rules* [JYM09] are used.

Polarity Classification by Linguistic Features and Machine Learning

As before, we use an SVM (RapidMiner[3] standard implementation) for the classification. The SVM receives the feature sets β and γ as input values for learning, as well as it obtains the **Basic Polarity Features** α. In this way, our machine learning approach is able to learn from the polarity features and the implicit features. For instance, a negative statement has positive opinion-bearing words, but it contains a negation and the score of the negated words is almost the score of the words within the statements. Thereby, our approach can learn that this is typical for a negative statement.

[3]Rapid-I: http://rapid-i.com/

6.5.4 Evaluation

Experiment Design

For this evaluation, the Finance dataset contains 5,500 statements (2,750 are positive, 2,750 are negative) from 3,452 different news articles. The second dataset is the pressrelations dataset [SCH12]. We use approx. 30% of the dataset to construct a sentiment dictionary. This means that 1,600 statements (800 are positive, 800 are negative) are used for Finance and 308 statements for the pressrelations dataset (we use only the positive and negative statements of the dataset). The sentiment dictionaries contain words (4,581 words for Finance, e.g.) which are weighted by the methods explained in section 6.4. In addition, we use the SentiWS [RQH10] as another baseline. The remaining set of 3,900 statements (1,950 positive and 1,950 negative) and 721 statements, respectively, are used for the evaluation of the linguistic features where we use 20% to train a classification model and 80% for testing. The evaluation shows the results in different combination of the features. So, $\alpha+\beta$ is the combination of set α and the feature set β and so on. The combination of α, β, and γ is indicated as all.

Text Preprocessing

For better results, we analyse not only the statements, but also the whole text from which a statement is taken. The basic framework for our approach is our Information Extraction Module (cf. section 2.4.2). Thus, we can resolve persons and organisations in statements, even if they are only mentioned by he/she/it in the statement, but mentioned by an identifiable name in the rest of the article. Thereby, words can be associated with persons or organisations, if they are only mentioned by pronouns or ortho-matches. For POS-tagging and lemmatisation, we apply the TreeTagger[4].

Experiment Results

The results are depicted in table 6.6. The left side shows the results on the Finance dataset and on the right side the results of the pressrelations dataset are illustrated. As table 6.6 shows, the features β and γ improve polarity allocation. The features increased performance of all methods, except the information gain method on the pressrelations dataset (the accuracy is over 0.5 percentage points lower by using all features in comparison with only using the baseline features α). However, in all other cases, the methods achieved the best results by using all features. SentiWS, as the lexicon-based approach, got the highest improvement in contrast to all other baselines (over 7 percentage points on Finance and over 14 percentage points on pressrelations). All features in combination with the entropy-based method got the highest accuracy

[4]TreeTagger: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

	Finance dataset				pressrelations dataset			
Method	α	α+β	α+γ	all	α	α+β	α+γ	all
SentiWS	0.6036	0.6590	0.6311	**0.6792**	0.5526	0.5604	0.615	**0.6943**
PMI	0.6174	0.6586	0.6317	**0.6881**	0.6245	0.6057	0.634	**0.6887**
χ^2	0.6872	0.7071	0.6981	**0.7234**	0.6453	0.6453	0.6717	**0.6868**
Entropy	0.7006	0.7221	0.7428	**0.7528**	0.6642	0.6604	0.6774	**0.6943**
Information Gain	0.6955	0.7186	0.7243	**0.7349**	**0.6912**	0.6761	0.6811	0.6828

Table 6.6: Results of the linguistic features.

with 75.28% on Finance, which is an improvement of over 5 percentage points to the baseline.

By comparing all results, the influence of feature set β seems to be bigger than the influence of feature set γ on Finance, while the γ features seems to be stronger than the β features on the pressrelations dataset. The reason for this is the nature of the two domains. The political texts are more complicated so that a deeper analysis, which exploits values of the influenced sentiment-bearing words, provides more benefit.

Nevertheless, except for the information gain method, the combination of all linguistic features achieved an increase to the baselines of at least over 3 percentage points.

6.5.5 Conclusion

In conclusion, linguistic features are useful for the classification of polarity in newspaper articles. The evaluation shows that the linguistic features can be integrated into existing solutions and thereby improve the computation of the polarity of sentiment. The improvement is especially large and therefore interesting for approaches, which use a general sentiment lexicon. The gain is large for general sentiment dictionaries, because the words are linguistically context-independent in these dictionaries. The information about this context can be added by analysing it. Moreover, this approach achieved high accuracies of over 70% and in one case an accuracy of over 75%.

Now, we want to integrate neutral statements. Research in this area [KS06a] promises an improvement, as far as this integration considers neutral statements as an independent class and does not treat neutral statements as data points which lie somewhere close to the positive-negative boundary. The work [KS06a] suggests that the accuracy of the sentiment determination could be increased in such a way. Thus, we will investigate features which are significant for neutral example and not neutral examples, respectively.

6.6 Integration of Neutral Examples and Limits

In the last sections, we have learnt much about the creation of sentiment dictionaries, the classification of the polarity of sentiment, and how linguistic features can increase this performance. Now, we want to expand this feature-based approach to neutral examples and, in this way, we establish a tonality classification. Some authors such as Esuli and Sebastiani remark that the Subjectivity Analysis (the distinction between neutral and subjective examples) is a harder task as the identification of polarity of sentiment [ES06].

Task Definition III: Tonality Classification. *Let $s \subseteq d$ be a statement and document d represents a newspaper article. The task is to determine the tonality y for a given statement s, consisting of s_n words:*

$$t_2 : s = (w_1, w_2, ..., w_{s_n}) \mapsto y \in \{\text{positive,neutral,negative}\} \tag{6.24}$$

Furthermore, we define the subjectivity as the distinction between subjective (positive and negative) statements and neutral statements.

6.6.1 Subjectivity Features for Neutral Examples

Many of the explained methods can be adapted to differentiate between positive and negative words by calculating four values for the subjectivity classification (one value for each word category: adverbs, adjectives, verbs, and nouns). In order to distinguish between subjective (positive and negative) statements and objective (neutral) statements, we use the same methods by changing the positive class to the subjective class and the negative to the objective class. This means for the chi-square method for example:

$$\bar{\sigma}_{\chi^2}(w) = \begin{cases} \chi^2(w) & \text{if } P(w|neu) < P(w|sub) \\ -\chi^2(w) & \text{otherwise} \end{cases} \tag{6.25}$$

In equation 6.25, the values for $P(w|sub)$ and $P(w|neu)$ are defined analogically to equation 6.4:

$$P(w|sub) = \frac{f(w, sub)}{f(w, sub) + f(\neg w, sub)} \qquad P(w|neu) = \frac{f(w, neu)}{f(w, neu) + f(\neg w, neu)} \tag{6.26}$$

Here, $f(w, sub)$ is the frequency of the word w in positive and negative statements ($f(w, sub) = f(w, pos) + f(w, neg)$) and $f(w, neu)$ is the frequency of w in neutral statements.

Method	SVM	Naive Bayes	Neural Net	Decision Tree	k-means
Chi-square	53.23%	54.71%	46.31%	43.13%	43.51%
PMI	46.31%	49.26%	43.02%	45.29%	46.23%
Entropy	53.23%	**55.39%**	48.47%	37.68%	52.04%
TF-IDF	48.92%	47.90%	52.44%	48.92%	38.96%
SentiWS	40.75%	34.62%	39.73%	42.00%	35.15%

Table 6.7: Results of the classification of the tonality on the pressrelations dataset [SCH12].

So, we get four additional values for Subjectivity Analysis. In this way, we treat neutral examples as an independent class as Koppel and Schler demand [KS06a]. As another baseline, we use the values (polarity classification) and the absolute values (Subjectivity Analysis) of the SentiWS dictionary [RQH10].

By adding these features for subjectivity, we can integrate neutral statements in our evaluation and perform a tonality classification. Since we do not want to make this section longer as necessary, we show the results of our experiments only on the pressrelations dataset [SCH12] here. We show results of the Finance dataset in section 6.6.3.

To classify the statements, we use different machine learning techniques[5] (see table 6.7). After the creation of the dictionaries for the polarity and the subjectivity, we select the first 220 statements of pressrelations dataset [SCH12] for training and 881 for testing. The accuracies of the tonality classification are shown in table 6.7. For the SVM, we choose a two-way classification: First we differentiate between subjective and objective statements, before we classify the subjective statements as positive or negative. Naive Bayes and the entropy method achieve the best result (55.39%).

So, the Naive Bayes classification performs a little bit better than an SVM, thus the results are based on this technique for Subjectivity Analysis (cf. table 6.8). We use only the four subjectivity values as input for this analysis. The chi-square method can differentiate best between subjective and objective examples (71.74%).

6.6.2 Linguistic Features for Neutral Examples

We implement different features which we found in literature. The paper "Towards Context-Based Subjectivity Analysis" of Benamara et al. [BCMP11] presents many linguistic and contextual features for the differentiation between subjective and neutral opinions in film reviews. The approach is not completely adaptable, because they also

[5]All techniques use the RapidMiner standard implementation with default parameters (http://rapid-i.com).

Method	Accuracy	Subjective		Objective	
		Precision	Recall	Precision	Recall
Chi-square	**71.74%**	73.51%	90.90%	62.76%	31.82%
PMI	69.35%	69.70%	12.59%	64.29%	96.64%
Entropy	69.47%	71.11%	22.03%	57.80%	92.27%
TF-IDF	67.99%	69.34%	94.29%	52.78%	13.29%
SentiWS	36.55%	80.00%	8.07%	33.37%	95.80%

Table 6.8: Results of the Subjectivity Analysis on the pressrelations dataset [SCH12].

analyse domains specifics such as emoticons and their approach works on elements, which they call elementary discourse units (EDU) [BCMP11]. An EDU represents a verbal clause in general, but an EDU can also correspond to other syntactic units. However, we adapt their ideas and add some own ideas in order to create new features analogous to our linguistic features. We call these features the **Contextual Subjectivity Features** ζ. Our hope was that the features should increase the performance of the classification accuracy as our linguistic features do (cf. section 6.5.3).

In detail, Benamara et al. [BCMP11] analyse syntactic features such as comparatives and superlatives of adjectives, so that we measure how large is the proportion of comparatives and superlatives relative to the number of adjectives (features ζ_1 and ζ_2, cf. table 6.9). In addition, we measure the proportion of adjectives and adverbs relative to all words (features ζ_3 and ζ_4, cf. table 6.9), because a higher number of adjectives and adverbs portends a subjective text [TBT+11].

Furthermore, Benamara et al. [BCMP11] analyse speaker verbs. They identify reporting or non-polar advice verbs by an own created lexicon, which has less than 270 verbs of this category. Unfortunately, this lexicon is on French and not publicly available. But verba dicendi (Latin for words of speaking) exist in many languages [Cru02]. A verbum dicendi communicates speech or starts a quotation such as reported speech. The verba dicendi [Cru02] can be categorized into positive verbs (such as "welcome", "support", "boast", or "agree"), neutral verbs (such as "say" or "explain"), and negative verbs (such as "refuse", "criticise", or "condemn"). We create and annotate a list of 473 verba dicendi (35 positive verbs, 304 neutral verbs, and 134 negative verbs).

We calculate the polarity and the subjectivity of verba dicendi in our approach by these two formulas:

$$f_{\zeta_5}(s) = \frac{vd_{pos}(s) - vd_{neg}(s)}{vd_{pos}(s) + vd_{neg}(s)} \tag{6.27}$$

Adjectives and Adverbs Features	*Contextual Features*
ζ_1: proportion of comparatives	ζ_5: RSV polarity
ζ_2: proportion of superlatives	ζ_6: RSV subjectivity
ζ_3: proportion of adjectives	ζ_7: location / ζ_8: date
ζ_4: proportion of adverbs	ζ_9: proportion of future tense

Table 6.9: Feature set for neutral statements.

Method	$\alpha_{pol+sub}$	$\alpha_{pol+sub}+\zeta$	$\alpha_{pol+sub}+\beta+\zeta$	$\alpha_{pol+sub}+\gamma+\zeta$	$\alpha_{pol+sub}+\beta+\gamma+\zeta$
Chi-square	0.504	0.4938	0.5142	0.4858	0.5108
PMI	0.4597	0.4677	0.4858	0.437	0.4711
Entropy	0.5471	0.5199	0.5221	0.5028	0.5085
SentiWS	0.4075	0.4234	0.4291	0.3927	0.4211
TF-IDF	0.4188	0.4677	0.4904	0.4779	0.4938

Table 6.10: Results of the linguistic features for neutral examples.

$$f_{\zeta_6}(s) = \frac{vd_{pos}(s) + vd_{neg}(s) - vd_{neu}(s)}{vd_{pos}(s) + vd_{neu}(s) + vd_{neg}(s)} \tag{6.28}$$

Benamara et al. [BCMP11] calculate two features which show, if a location or date is present. We use these features also for our evaluation (feature ζ_7 or ζ_8). Likewise, we count how often the future tense is used in contrast to all full verbs (feature ζ_9). We observe many verbs in future tense in neutral statements and we hope to capture a similar effect as irrealis blocking [TBT+11] (irrealis blocking will be explained in detail in section 7.1.3).

6.6.3 Evaluation

We perform our experiments on our Finance dataset with the same configurations as in section 6.5.4, the evaluation of our linguistic features. But now we also include neutral statements within this evaluation. This means that we use 8,500 statements (4,250 are neutral, 2,125 are positive, and 2,125 are negative) of the Finance dataset. Here, SVMs achieve again higher accuracies than a Naive Bayes classification. As before, the results of the methods are similar on both datasets and we do not benefit very much from a comparison at this point. Table 6.10 shows the accuracies. $\alpha_{pol+sub}$ stands for the 8 basic features; 4 of them for distinction between positive and negative statements (α_{pol} are the old α features) and 4 for Subjectivity Analysis (α_{sub}) as described in section 6.6.1. Again, $\alpha_{pol+sub} + \zeta$ indicates the combination of $\alpha_{pol+sub}$ and ζ features.

As table 6.10 shows, the neutral features ζ do not increase the accuracy very much. For the chi-square method, the $\alpha + \zeta$ combination achieves a lower accuracy. The accuracy only increases above the accuracy of the baseline by adding the β features. The entropy-based method achieves only worse accuracies in combination with the ζ features. The PMI method gets an improvement of under 3 percentage points ($\alpha + \zeta$). The results are similar for the SentiWS. The improvement is higher for TF-IDF (over 7 percentage points), but the accuracy is the second lowest (after SentiWS).

6.7 Final Conclusion

As seen in the last evaluation, linguistic features are hard to fathom, when they should be applied to a tonality classification, which includes neutral examples. Analogous to Esuli and Sebastiani [ES06], we found out that Subjectivity Analysis is more complicated than the polarity of sentiment. On the contrary, we found a machine learning-based approach for the polarity classification, which uses polarity features calculated by single words. It can be expanded with linguistic features for the polarity, which increases the classification accuracy to very well results of over 70% and partially even over 75%. Thereby, an automated trend analysis is possible.

The words from the four important categories seems to be more important, or to put it more succinctly, are the features which achieve a more accurate classification result for the tonality. So, for the tonality classification, we will make a step back, concentrate on this fact, and pick up a previous idea again: As mentioned before, we believed that an approach based on combinations of words should be more suitable for the tonality classification. However, techniques from literature and own ideas already failed to classify the polarity. Nevertheless, relationships of words among each other are missing. And we wish to start right there, but we try to find a more flexible model than triplets, for instance. Thus, we created a new model which connects important words of the four categories plus negations and model their combination with respect to the tonality. As a result, we developed a graph-based approach, which we will explain in the next chapter.

Also, lexicon-based approaches can be expanded by analysing the context [DLY08, WWH09, TBT+11]. These state-of-the-art approaches are also explained in detail during the next chapter. Furthermore, our pre-evaluation has shown that a bag-of-words as an input for machine learning is also possible and generate good results at least on small datasets. This is also an approach based on combinations of words. We neglect this approach a little bit in this chapter, because our research project has some constraints: The media analysts should be able to maintain the created sentiment dictionaries. A term document matrix representing the bags-of-words is very large

and sparse. Therefore, it is not easy to maintain, so that the analysts can delete wrong entries or insert new ones. Here, analysts encounter difficulties, because it is not clear what a wrong entry is. Is it a whole row? On the one hand, deleting a whole row seems to be the most appropriate solution to maintain a bag-of-words approach, because editing single values of the row would be very laborious and overwhelming for the media analysts. On the other hand, a row is a concretely learned example which has this tonality. This effectively means: An analysts has assigned an incorrect tonality to a statement.

By the way, the problem of the maintenance would be also an advantage of a graph-based approach, because word pairs such as “large” and “profit” can be displayed for a user. But this consideration was becoming less and less important during the project, because the manual maintenance of the system was considered less necessary. So, we evaluate also the partially explained RSUMM method (Review Summary) [SPV11] as another state-of-the-art technique. RSUMM operates with bag-of-words as the underlying model in order to select word features for machine learning.

7

Tonality Classification

In the last chapter, we have seen how challenging it is to establish a well working tonality classifier. The findings of our results have shown that it is especially hard to differentiate between subjective and neutral examples. The difference between positive and negative seems to be bigger in contrast.

One surprising result was that the approaches, which use multiple words to create a sentiment score or features for machine learning, achieve worse accuracies than the approach based on single words. This is surprising, because we still believe, that word combinations should provide more information about the tonality than single words. Although linguistic features can improve the distinction between positive and negative statements, the integration of linguistic features for the subjective and objective distinction was a failure. The failure brought us finally to make a step backwards and to design a new approach based on word combinations, which consists of a more flexible model to extract and collect tonality-bearing words.

The outcome of this reasoning leads to a graph-based approach, which we introduce in this chapter. One basic idea behind this approach replaces sentiment dictionaries through a model, which learns the tonality-indicating word combinations precisely. We apply the edges of the graph, which we call word connections, in order to calculate tonality features for machine learning.

As we have seen the first chapters, viewpoints play a significant role in a newspaper and we are aware of this fact, but since we concentrate on the determination of the tonality, the extraction of viewpoints can be solved in a separate step (cf. chapter 8). This is possible, because the tonality of a statement can be determined without knowledge of the viewpoint in almost all cases. The only exception is a statement with multiple viewpoints and different tonalities, but these statements are very rare

(see next paragraph). At the beginning of this thesis it was an open question whether a viewpoint has an influence on the tonality or the divide and conquer principle is a better solution in order to tackle the two problems independently of each other. For this chapter, we assume that a divide and conquer solution is the best way, but we will investigate this question in the next chapter (note that the results of the research of the next chapter was obtained before we develop the graph-based approach).

Across the board, it is possible that one statement has two or more viewpoints. This is the case for 116 statements (approx. 7.62%) of the pressrelations dataset and 279 statements of the Finance dataset (approx. 3.28%). The tonality can be different for the different viewpoints in general. This means, consequently, that statements can have two or more different tonalities for different viewpoints, but this is rarely the case (for less than 3.56% of the pressrelations statements and less than 0.17% of the statements of the Finance dataset in its largest version). One of these examples is the following statement, which is a translated statement of the pressrelations dataset:

(7.1) The logical consequence would be a substantial increase of the subsidies, which the SPD fraction has demanded several times. But the government has limited the funding for 2011 and a too slight rise is planned for 2012. (**Code A:** positive, SPD; **Code B:** negative, CDU)

So, we keep them as two statements with different tonalities within the dataset, because this case can occur in an MRA. However, we will show that this situation does not irritate our approach too much.

Likewise, we also explain four state-of-the-art methods in this chapter, because they are our comparison methods for our approach. Besides, two of the methods, namely SO-CAL [TBT+11] and Opinion Observer [DLY08], are also recommendations of conference reviewers. The approaches are implemented and evaluated in their initial form and also in several variants in order to try to improve the classification accuracies of the tonality. Some approaches get an additional dictionary in one variant, for instance. We want to derive robust conclusions from the results in this way. So, first we describe the four state-of-the-art approaches in detail. Then we explain how we construct our graph and learn word connections for a tonality classification. Then we evaluate all methods in detail, before we draw conclusions.

Our graph-based approach and the belonging study of its comparison with state-of-the-art methods was published in the paper "Opinion Mining in Newspaper Articles by Entropy-based Word Connections" on the 2013 Conference on Empirical Methods in Natural Language Processing [SC13c].

7.1 State-of-the-Art Approaches

7.1.1 Wilson

Our first state-of-the-art method is published by Wilson et al. [WWH09]. They use machine learning for a word-based sentiment classification, which measures contextual influences profoundly through well selected features. They propose two types of features: one type for the subjectivity and one type for polarity of words.

Features for Neutral-Polar Classification

For the Subjectivity Analysis, Wilson et al. [WWH09] estimate *Word Features*, *General Modification Features*, *Polarity Modification Features*, *Structure Features*, *Sentence Features*, and *Document Features* (32 features in total).

The *Word Features* consists of the word token itself, the POS-tag of the word, of the previous word and of the next word. In addition, a feature indicates the prior polarity as *positive*, *negative*, *both*, or *neutral* and another feature indicates the reliability class as *strong subjective* or *weak subjective*. The MPQA corpus [WWC05], which was created during the same research project, has word-based annotations in the same range for the prior polarity and reliability class. To get this information, they look up subjective clues in a dictionary [RW03], which also belongs to this project.

The *General Modification Features* show, whether the word is preceded by an adjective, an adverb, or an intensifier, or whether the word itself is an intensifier. They use a list of Quirk et al. [QGLS85] for their intensifiers. Furthermore, they estimate four features, if the word modifies a strong subjective clue, a weak subjective clue, or is modified by a strong subject clue or a weak subjective clue. To calculate these relations, they parse a sentence [Col97] and they convert the parse tree into a dependency representation [XP01] (cf. figure 7.1). The modifiers of a word are its children. All these features are binary features.

The three *Polarity Modification Features* show the polarity of a modified object, of the modifier, and of the conjunction with values *positive*, *negative*, *neutral*, *both*, or *not modified*.

For the three binary *Structure Features*, they climb the dependency tree from the node of the word to the root and observe, if they pass a subjective (subj) relationship, a node, which is a main and a copular verb, or nodes representing a passive verb pattern on the climb. In figure 7.1, the subjective relationship is true for the first four words of the sentence, for example.

The *Sentence Features* [WR05] are numbers of strong subjective clues or weak subjective clues in the current, previous, or next sentence as $0, 1, 2$, or $>= 3$ (six features). Also, the numbers of adjectives and adverbs are counted in the current

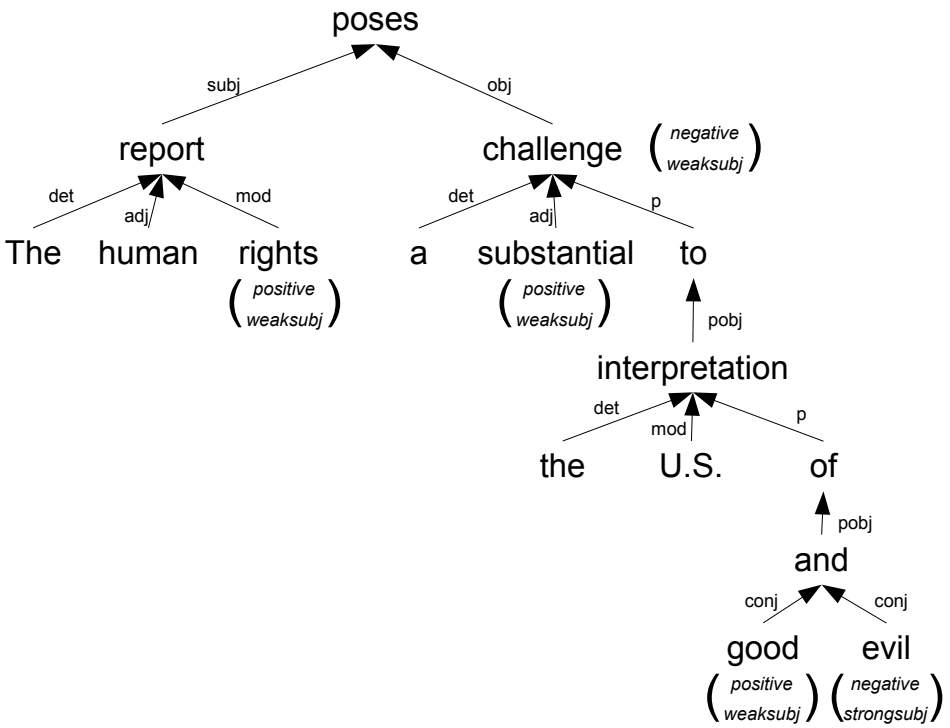

Figure 7.1: This example is taken from [WWH09] and shows the dependency tree for the sentence "The human rights report poses a substantial challenge to the U.S. interpretation of good and evil." [WWH09].

sentence (as $0, 1, 2$, or $>= 3$) and three binary features show the existence of a cardinal number, a pronoun, or a modal (except will) in the current sentence.

The last feature shows the topic or the domain of the document (*Document Feature*). This is obtained from the ten topics of the MPQA corpus (*argentina*, *axisofevil*, *guantanamo*, *humanrights*, *kyoto*, *settlements*, *space*, *taiwan*, *venezuela*, *zimbabwe*). Also, they introduce more general topics for documents of the MPQA corpus, which have no topic annotation: *economics*, *general politics*, *health*, *report events*, and *war and terrorism*. For our datasets, we use the two topics *economics* and *general politics*.

Features for Polarity Classification

For the classification of the polarity of sentiment, Wilson et al. [WWH09] consider 10 features. Five features are already explained, namely all *Polarity Modification Features* and the two *Word Features*: word token and word prior polarity.

The other features include, inter alia, the *Negation Features*, which are two binary features. The first feature called negated is *true*, if the word is negated in this sentence. The second feature called negated subject is *true*, if the subject of the sentence is negated like in this example:

(7.2) No proposal of the review is helpful.

The *Polarity Shifters* show the appearance of shifters in a window of four words before the word through three binary features. The general polarity shifters reverse the polarity, negative shifters change the polarity to negative and positive polarity shifters modify it to positive.

Machine Learning Techniques

Wilson et al. [WWH09] apply four different types of machine learning for their evaluation of the features: They evaluate boosting (BoosTexter [SS00], AdaBoost.MH with 2,000 rounds of boosting), memory-based learning (Ripper [Coh96]), rule learning (TiMBL [DZVdSVdB03]), and support vector learning (SVMLight [Joa99]).

7.1.2 Opinion Observer

Basic Lexicon

The basic lexicon [HL04] is created by using a bootstrapping process and WordNet [Mil95]. The bootstrapping process starts with a small list of words (these lists contain only adjectives in earlier contributions [HL04]), which are marked as positive or negative. The list grows by co-occurrences of labeled words with new words, which become the label of the marked word. In their final dictionary, adjectives and also adverbs, verbs, and nouns are listed as opinion words. In addition, they annotate a list with more than 1,000 idioms [DLY08], which they treat as opinion words. Unfortunately, it is not clear, how many entries the lexicon finally has.

Opinion Orientation Algorithm

With their basic lexicon, they identify the opinion words w_i in one sentence s and then aggregate the orientation score for every product feature f in s. Positive words have a score of +1, while negative words have a score of -1 (denoted as $w_i.SO$). They use the following formula to calculate the aggregated score:

$$score(f) = \sum_{w_i : w_i \in s \bigwedge w_i \in V} \frac{w_i.SO}{dis(w_i, f)} \tag{7.1}$$

Here, V denotes the set of all opinion words, while $dis(w_i, f)$ is the distance between feature f and opinion word w_i in the sentence (the distance is not clearly defined in the paper [DLY08], we assume that it is the number of words between them including the opinion word itself). If the final score is greater than zero, then the opinion is positive, and it is negative, if the final score is below zero. The opinion is neutral, if the score is exactly zero.

To perform this aggregation and to handle the (linguistic) context of the opinion

words, they introduce a new algorithm called *Opinion Orientation* (cf. algorithm 2). This algorithm has sub-processes as the *Word Orientation* procedure (cf. algorithm 3) or the *Handling of Context Dependent Opinion Words* (cf. algorithm 4) and different rules (such as the *Negation Rules* or the *But Clause Rules*, cf. algorithm 5).

Algorithm 2: OpinionOrientation [DLY08]

```
   Data: sentences S
   Result: rated sentences S and features F
 1 foreach sentence s_i ∈ S that contains a set of features do
 2 |   features F = features contained in s_i;
 3 |   foreach feature f_j ∈ F do
 4 |   |   orientation = 0;
 5 |   |   if feature f_j is in the "but" clause then
 6 |   |   |   orientation = apply "But" Clause Rules
 7 |   |   else
 8 |   |   |   remove the "but" clause from s_i if it exists;
 9 |   |   |   foreach unmarked opinion word ow in s_i do
10 |   |   |   |   //ow can be a TOO word or Negation word as well
11 |   |   |   |   orientation+ = WordOrientation(ow, f_j, s_i);
12 |   |   |   end
13 |   |   end
14 |   |   if orientation > 0 then
15 |   |   |   f_j's orientation in s_i = 1;
16 |   |   else
17 |   |   |   if orientation < 0 then
18 |   |   |   |   f_j's orientation in s_i = −1;
19 |   |   |   else
20 |   |   |   |   f_j's orientation in s_i = 0;
21 |   |   |   end
22 |   |   end
23 |   |   if f_j is an adjective then
24 |   |   |   (f_j).orientation += f_j's orientation in s_i;
25 |   |   else
26 |   |   |   let o_ij be the nearest adjective word to f_j in s_i;
27 |   |   |   (f_j, o_ij).orientation += f_j's orientation in s_i;
28 |   |   end
29 |   end
30 end
31 ContextDependentOpinionWordsHandling(S,F);
```

The *Word Orientation* procedure (cf. algorithm 3) calculates the orientation of an opinion word. First it applies the *Negation Rules* or the *TOO Rules*, if necessary. Otherwise, the orientation is looked up in the lexicon and weighted like in equation 7.1.

Their *Negation Rules* do the following: A negative opinion-word becomes positive,

a positive word becomes negative and a neutral one also becomes negative ("does not work"). Moreover, they apply *TOO Rules* which observe the preposition "too" in expressions such as "too large". This indicates a negative opinion and so all words influenced by "too" are turned into negative opinions (the exact method is not shown in the paper [DLY08], but the function can be deduced from the textual description).

Algorithm 3: WordOrientation [DLY08]

Data: word w, feature f, sentence s
Result: orientation
if *w is a Negation word* **then**
  orientation = apply **Negation Rules**;
  mark words in s used by Negation Rules;
else if *w is a TOO word* **then**
  orientation = apply **TOO Rules**;
  mark words in s used by TOO Rules;
else
  orientation = orientation of w in Opinion Word List;
end
$orientation = orientation/dis(w, f)$;

The *Handling of Context Dependent Opinion Words* procedure (cf. algorithm 4) tries to obtain an orientation from the context within the sentence or even from the previous or next sentence (*Inter-sentence Conjunction Rule*, see below).

Algorithm 4: ContextDependentOpinionWordsHandling [DLY08]

Data: rated sentences S and features F
Result: rated sentences S and features F
foreach *f_j with orientation = 0 in sentence s_i* **do**
  if *f_j is an adjective* **then**
    f_j's orientation in $s_i = (f_j)$.orientation;
  else
    //**Synonym and Antonym Rule** should be applied too
    let o_{ij} be the nearest opinion word to f_j in s_i;
    if (f_j, o_{ij}) *exists* **then**
      f_j's orientation in $s_i = (f_j, o_{ij})$.*orientation*
    end
  end
  if *f_j's orientation in $s_i = 0$* **then**
    f_j's orientation in s_i = apply **Inter-sentence Conjunction Rule**
  end
end

If a previous sentence or a next sentence exists, Ding et al. [DLY08] take the orientation of the last or next sentence as the orientation of the actual sentence with one

exception: Analogue to the *But Clause Rules*, the orientation is flipped, if a sentence starts with "but" or "however". This step is called *Inter-sentence Conjunction Rule* [DLY08].

In addition, a sentence which contains the word "but" requires particular processing as we have seen in the last chapter. Ding et al. [DLY08] invert the orientation of the words in the sentence before the conjunction, because the opinion before the word "but" and after it are usually the opposite of each other. Algorithm 5 (*But Clause Rules*) shows that.

Algorithm 5: "But" Clause Rules [DLY08]

```
Data: feature f, sentence s
Result: orientation
if f appears in the "but" clause then
    foreach unmarked opinion word ow in the "but" clause of s do
        //ow can be a TOO word (see below) or Negation word
        orientation+ = Word Orientation(ow, f, s);
    end
    if orientation ≠ 0 then
        return orientation;
    else
        orientation = orientation of the clause before "but";
        if orientation ≠ 0 then
            return −1 ∗ orientation;
        else
            return 0;
        end
    end
end
```

Ding et al. [DLY08] provide some more little steps and ideas (*Synonym and Antonym Rule*, *Intra-sentence Conjunction Rule*, and *Pseudo Intra-sentence Conjunction Rule*) especially for the product review domain. These aspects are not relevant for Opinion Mining in newspaper articles and they do not appear within our dataset. Their final implementation is called **Opinion Observer** [DLY08].

For our processing, we do not extract product features and handle them, but we analyse the sentences. As a result, we can simplify the process, because it is not required to identify features of products. So, we can focus on the semantic orientation of the sentences and can compute a score of the semantic orientation for a statement as a sequence of sentences. Technically, we assume that every sentence of a statement contains a feature and all words within this sentence belong to this feature and have a distance of one. In this way, we get the semantic orientation of a sentence.

Word	Category	SO Value	Word	Category	SO Value
monstrosity	noun	-5	masterpiece	noun	+5
hate	noun & verb	-4	relish	verb	+4
inexcusably	adverb	-3	good	adjective	+3
fabricate	verb	-2	purposefully	adverb	+2
delay	noun & verb	-1	determination	noun	+1

Table 7.1: Examples of the original dictionaries of SO-CAL [TBT+11].

7.1.3 SO-CAL

Dictionaries

To build the sentiment dictionary, Taboada et al. [TBT+11] collect adjectives from a corpus [TG04, TAV06] called "Epinions 1". The 400 reviews of the corpus are Epinions reviews[1], which belong to one of eight different products groups (book, car, computer, cookware, hotel, film, music, phone). They annotate the collected list of adjectives manually, because they do not trust the stability of (semi-)automatically generated lexicons [TBT+11]. They assign Semantic Orientation Value (SO-Value) on the scale from -5 (extremely negative) to +5 (extremely positive) to every word in the list.

As a result, their dictionary for adjectives includes 2,252 entries. Furthermore, they collect 1,142 nouns, 903 verbs, and 745 adverbs in three further dictionaries. Here, they combine the "Epinions 1" with 100 film reviews of the 2,000 film reviews of the Polarity Dataset [PLV02, PL04, PL05] and a list of positive and negative words of the General Inquirer Dictionary [SDSO66, Sto97], in order to be more context independent and also be able to analyse more formal texts. We list some examples of the final entries in table 7.1.

Intensification

SO-CAL [TBT+11] identifies modifiers and analyses which words are influenced by the modifiers. SO-CAL knows two kinds of modifiers: amplifiers and downtoners. The influence is calculated by a percentage value: a positive value for an amplifier and a negative value for a downtoner. Table 7.2 shows some examples.

To calculate the influence, semantic-oriented words are assigned with a new value. The old value is multiplied with 100% plus the value of the modifier. So, "really good" becomes a value of +3.45 (= +3 * (100% + 15%)). Also, multiple modifiers are possible.

[1] www.epinions.com/

Downtoner	Modifier (%)	Amplifier	Modifier (%)
slightly	-50	really	+15
somewhat	-30	very	+25
pretty	-10	most	+100

Table 7.2: Examples of the intensifiers of SO-CAL [TBT+11].

Negation

Many publications such as [HL04, PZ06, CC08] treat the negation as a switch/flip of the polarity. This would mean, that the value of "not good" turns from +3 ("good") to -3. However, Taboada et al. [TBT+11] handle negations in a different way: If a word is negated, then they shift the polarity of the word by a constant value towards the opposite polarity. The constant value of the shift is 4 in their implementation. Thus, "not good" become an SO value of -1 (= +3 - 4), instead of -3.

To estimate the scope of a negation, Taboada et al. [TBT+11] propose two approaches: The first scope is larger, because all words following the negation belong to the scope, until "a clause boundary marker" [TBT+11] (punctuations, sentential connective such as "if", "but" or "and") is reached. The second, more conservative approach is complex. Here, SO-CAL looks backwards from a potentially negated word as long as the considered words or their POS-tags belong to a skip list [TBT+11]. Each category (adjective, noun, verb, and adverb) has an own list. For example, adjectives can be skipped for nouns. In general, adjectives, copulas, determiners, and certain basic verbs [TBT+11] are on the skip lists. This conservative approach produces better results [TBT+11].

Taboada et al. [TBT+11] consider negators such as "not", "none", "nobody", "never", "nothing" and words with similar function ("without", e.g.). They do not consider any negative polarity items (NPIs) [TBT+11] for their negation search. These items (such as "any", "anything", "ever", e.g.) are treated by the next step: irrealis blocking.

Irrealis Blocking

SO-CAL identifies some special expression and constructions, which tell the reader, that this text part does not really contain an actual opinion or sentiment. The linguistic term for this situation is called irrealis. Irrealis markers can be modals, conditional markers (such as "if"), NPIs, or verbs (such as "expect" or "doubt"). Furthermore, words in questions or quotations are blocked as opinions or sentiments.

(7.3) The next movie of Spielberg should be great. (+3 → 0)

(7.4) Is this a bad situation? $(-3 \rightarrow 0)$

In both examples, the semantic oriented words "great" and "bad" and so the whole text parts are excluded from the final calculation of the SO score for the whole text. SO-CAL also tries to identify rhetorical questions which do not act as an irrealis blocker. Since this technique is not explained in detail in their article [TBT+11] and rhetorical questions are not such a large problem in our datasets, we do not consider this issue any further.

Text Level and Other Features

Taboada et al. [TBT+11] assess negative orientation higher, because humans tend to use more positive words and expressions than negative words, but their (and our) datasets/dictionaries are more equally distributed. So, they revaluate negative values by increasing the SO value of negative words by a constant factor (50% in their implementation).

Furthermore, if a word with an SO value appears more than one time in text, the SO value is decreased by the factor $1/n$, where n is the number of the appearance of the word.

Moreover, SO-CAL allows a weighting of special parts of the text (the beginning of a text, e.g.) and text areas, which are marked with a special tag. These techniques are not evaluated in this contribution [TBT+11] and they seem not to be appropriate for our task. The statements are already the important texts parts.

With all these techniques, SO-CAL provides an SO value for a text (a statement for example). The rating system allows multiple cut-offs. So, we can classify the statements based on a scale (for example, above +2 represents a positive text, a text with a value from -2 to +2 is neutral and so on).

Example

We want to illustrate SO-CAL with the following example, which is published in [Sch13] (please note that the example was published in German and so the values are created by our own, but they adapt the real values of the SO-CAL dictionaries):

(7.5) I did not enjoy this film. The performance of its actors was very bad.

This snippet could be part of a film review or a post on Twitter or Facebook. The opinion of the author is clearly negative about the film. SO-CAL identifies the verb "enjoy", the noun "performance" and the adjective "bad" as opinion-bearing words, because they have an entry in the dictionaries. Then SO-CAL looks up the Semantic Orientation values: "enjoy" has a value of +3, "performance" is weak positive with

+1, and the negative adjective "bad" has got a value of −3. At this point, the sum of the values would be $+1(= 3 + 1 - 3)$, and therefore the text would be neutral to slightly positive.

But the verb "enjoy" is negated. With a negation, the value is shifted by 4 to the opposite polarity. Since "enjoy" has got a positive value, its value is decreased from +3 to −1. In addition, SO-CAL recognize that "very" modifies the adjective "bad". The modifier "very" intensifies the SO value by 25%. In this way, "bad" with −3 turns into "very bad" with −3.75. Finally, the text snippet has a score of $-3.75(= -1+1-3.75)$, so that it is clearly negative. At this point we forego the stronger weighting of negative words in order to keep this example simple.

7.1.4 Tonality Classification with RSUMM

In section 5.4, we have explained the first part of RSUMM [SPV11]. The first part has extracted the important sentences. Now, we come to the second part, which selects the most important words as features.

Sarvabhotla et al. [SPV11] select the more relevant features by the mutual information (MI) and the Fisher discriminant ratio (FDR). These methods has already been applied to Text Categorization [YP97] and Sentiment Analysis [WLWL09]. Sarvabhotla et al. [SPV11] intend to reduce dimensionality for higher classification accuracy and faster training.

The mutual information (MI) measures how significant the feature f is for the class C.

$$MI(f;C) = P(f,C)\log(\frac{P(f,C)}{P(f)P(C)}) \tag{7.2}$$

$MI(f;C)$ is the mutual information between feature f and class C. $P(f,C)$ is the probability of feature f occurring in class C. $P(f)$ is the probability of f in the entire collection, while $P(C)$ is the probability of class C.

The fisher discriminant ratio (FDR) is applied in pattern recognition in order to reduce the dimensionality of D-dimensional points which are projected in a lower dimensional space so that the difference of the means is maximal and the variance within each class is minimal.

$$J(w) = \frac{|m_1 - m_2|^2}{S_1^2 + S_2^2} \tag{7.3}$$

Here, m_i is the mean of class i and S_i is the belonging within-class variance. Sarvabhotla et al. [SPV11] calculate "the discriminating power of a feature" [SPV11] based on this idea.

$$FDR(f) = \frac{(\frac{m_1}{m} - \frac{n_1}{n})^2}{\frac{\sum_{i=1}^{m}(d_{P,i}(f) - \frac{m_1}{m})^2}{m} + \frac{\sum_{j=1}^{n}(d_{N,j}(f) - \frac{n_1}{n})^2}{n}} \tag{7.4}$$

In equation 7.4, m and n are the numbers of documents in class P or N, respectively, while m_1 or n_1 refer to the number of occurrences of f in P or N. The two classes P and N can be positive and negative texts, but also subjective and objective texts. The presence or absence of feature f in document i of class P is denoted by $d_{P,i}(f)$, as well as $d_{N,j}(f)$ shows the presence or absence of feature f in document j of class N.

If f has a high discriminative power, the values of MI and FDR should be high. The final subset selection retains the top $Y\%$ of features for each method. In our evaluations, features ranked by FDR produce consistently better accuracies than features ranked by MI, so we show only results for FDR. Features are in general n-grams; they use only uni-grams and bi-grams in their evaluation and uni-grams perform better. To a certain extent, this is understandable, because their bag-of-words representation for a text already models word combinations. Thus, the consecutive word combinations such as bi-grams are represented, too.

In combination with the first part of RSUMM [SPV11] (explained in chapter 5.4), the most important sentences (X% of the sentences) and then the most important terms (Y% of the terms) of the sentences can be extracted from a text. RSUMM(30%, 50%) would mean that 30% of the sentences are used to extract 50% of the terms of these sentences.

7.2 Our Approach for Learning Tonality

After the detailed description of the four state-of-the-art methods, we explain our solution to tackle the tonality classification. First, we explain how we construct our graphs to learn word connections, before we describe how we delexicalize the feature space by translating recovered word combinations in unseen statements into eight tonality features for machine learning (four for the polarity and four for the subjectivity). Basically, this is the same idea as in the last chapter, when we translate words into sentiment scores by using dictionaries. To perform this translation, we propose and compare two weighting methods.

7.2.1 Graph Model for Word Connections

To solve the tonality classification task for an MRA (cf. Task Definition III: Tonality Classification, section 6.6), we propose a graph-based approach to capture the opinion-bearing words and modifiers such as negations. In this way, our approach is able to recognize tonality-indicating structures (subgraphs) which provide precise information

1)	This solves the crisis.	(positive)
2)	This solves the crisis slowly.	(neutral)
3)	This intensifies the crisis.	(negative)

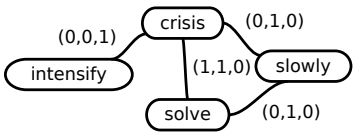

Figure 7.2: An example for different statements and a graph: The weights base on the three examples and their notation is (positive,neutral,negative).

about the tonality, even if statements have a very similar bag-of-words representation and at the same time different tonalities. One could also say that we create a graph instead of a sentiment dictionary from training examples, as other approaches [KK07, DTCY10] proceed.

In figure 7.2, simple examples are shown with a possible graph (the nodes and edges are taken from the given statements; of course, the graphs and weights become larger in practice). These simple examples are concentrated on nouns, verbs, and adverbs, but also examples with combinations of other categories are possible, such as, for example, different combinations of adjectives, nouns, and verbs: "This is a black day for the company", "The company is in the black", "The company is in the red", and "The company prevents to be in the red". Thus, even though the word representation is quite similar, the tonality can be different.

For opinion-bearing words, we use adjectives, nouns, verbs, and adverbs, which are widely acknowledged as opinion-bearing word categories [BWC11, RQH10, TBT+11]. Furthermore, we also include negation particles. Therefore, the vocabulary V is the set of opinion-bearing words in lemma for one set of statements S. Thus, for every lemma $w \in V$, the approach creates one node v in the graph. A node v also contains the type information (adjective, noun, verb, adverb, or negation).

The edge e_{ij} shows the appearance of node v_i and v_j in combination with tonality y by means of a weight $\varepsilon_{i,j}$ (the sequence of the values in equation 7.5 is also used in figure 7.2 and 7.3).

$$\varepsilon_{ij} = (y_{ij\pi}, y_{ijo}, y_{ij\nu}) \tag{7.5}$$

$y_{ij\pi}$ is the number of co-occurrences of node v_i and v_j in positive statements within the same sentence. In analogy, y_{ijo} belongs to sentences of neutral statements and $y_{ij\nu}$ to sentences of negative statements. Figure 7.2 shows a small example for this calculation, too.

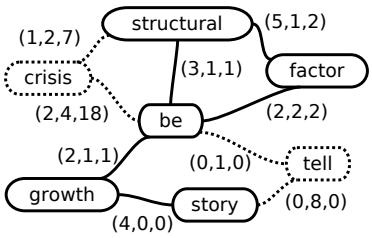

Figure 7.3: An example of a learned graph: The nodes and edges, which are drawn in solid lines, represent the recognized subgraph G_{sl} for the sentence "There are structural factors behind the African growth story.".

7.2.2 Generating Features for Learning

From a learned graph, we can combine different edges to calculate tonality features for an unseen statement s. An unseen statement is a statement, which is of course not used to learn the graph. We use all edges of the subgraph G_{sl} which contains the nodes for every lemma w_i in the l-th sentence of s.

We explain this in the following example: Assuming that our learned graph is shown in figure 7.3. It contains seven nodes and nine edges (also the nodes and edges in dashed lines). If we further assume that an unseen statement is the following example, which is a positive statement from an article in The Telegraph (8th Aug 2012) dealing with the prospects of British companies in Africa:

(7.5) There are structural factors behind the African growth story: a growing and sizeable population which is increasingly urbanised with disposable income; growing political stability; and a financial services industry that is still in its infancy. (**Code:** positive)

To keep this example short, we take the part until the colon as the first sentence of the statement: "There are structural factors behind the African growth story."

Our approach recognizes the nodes for "be", "structural", "factors", "growth", and "story". Thus, the subgraph G_{sl} for the first sentence ($l = 0$) would be the graph which is drawn in solid lines in figure 7.3. In this example, it is a connected graph, but it does not have to be.

We could also look for complete or connected graphs in the statement instead of using all edges. The largest complete graph would consist of the nodes "structural", "factor", and "be" in our example. But using all edges achieves better results, because this method provides all information. In addition, this method is quicker (search for largest complete or connected graph can be omitted, which would be an additional check).

If we have found our subgraphs G_{sl}, we can then compute the vectorial sum of all edges for one node v_i and we get the probability for a tonality y, if we observe v_i in

the l-th sentence. For this purpose, the corresponding edge weights are added up to calculate four probabilities:

$$P(pos|v_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi}}{\sum\limits_{e_{ij} \in Gsl} (y_{ij\pi} + y_{ij\nu})} \tag{7.6}$$

$$P(neg|v_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ij\nu}}{\sum\limits_{e_{ij} \in Gsl} (y_{ij\pi} + y_{ij\nu})} \tag{7.7}$$

$$P(sub|v_i) = \frac{\sum\limits_{e_{ij} \in Gsl} (y_{ij\pi} + y_{ij\nu})}{\sum\limits_{e_{ij} \in Gsl} (y_{ij\pi} + y_{ijo} + y_{ij\nu})} \tag{7.8}$$

$$P(neu|v_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ijo}}{\sum\limits_{e_{ij} \in Gsl} (y_{ij\pi} + y_{ijo} + y_{ij\nu})} \tag{7.9}$$

For the subjective class (sub), we add the appearance in positive statements ($y_{ij\pi}$) and negative statements ($y_{ij\nu}$). Otherwise we take the appearances in statements of the same class. The denominators of the polarity refer only to positive and negative appearances, while the denominators for the subjectivity refer to every tonality.

By calculating the vectorial sum, we combine several edges in order to estimate precise tonality scores. In this way, we can get the correct tonality score for the noun "crisis", if a sentence contains also "solve" and "slowly" ($\rightarrow$ more neutral) or "intensify" ($\rightarrow$ more negative) (cf. figure 7.2). And we get the correct tonality score for the adjective "structural", if a sentence includes also "crisis" ($\rightarrow$ negative) or the nodes "factor", "be", "growth", and "story" ($\rightarrow$ positive) (cf. figure 7.3).

We distinguish between different word categories (analogous to chapter 6, we have noticed that this creates better results than just having a single feature for one statement). Thus, every category gets its own feature and every node only has a tonality value, if it belongs to the category of the feature. This does not mean that we only consider edges which connect two nodes with the same category; we divide the influence of different categories into different features:

$$T_{cat,z}(v_i) = \begin{cases} f_z(v_i) & \text{if } v_i \in cat \\ 0 & \text{if } v_i \notin cat \end{cases} \tag{7.10}$$

$cat \in \{adj, adv, n, v\}$ indicates the category of the node (adjectives, adverbs, nouns, or verbs) and z specifies the type of feature. One type shows the difference between positive and negative polarity ($z = pol$), for the other type we replace the positive class

Polarity Features	Subjectivity Features
$T_{v,pol}$: polarity for edges with verbs	$T_{v,sub}$: subjectivity for edges with verbs
$T_{n,pol}$: polarity for edges with nouns	$T_{n,sub}$: subjectivity for edges with nouns
$T_{adv,pol}$: polarity for edges with adverbs	$T_{adv,sub}$: subjectivity for edges with adverbs
$T_{adj,pol}$: polarity for edges w. adjectives	$T_{adj,sub}$: subjectivity for edges w. adjectives

Table 7.3: Polarity and subjectivity features based on word connections.

by the subjective one (the sum of positive and negative) and the negative by a neutral one in order to differentiate between neutral and non-neutral examples ($z = sub$). As a result, we calculate eight features (see table 7.3) for the tonality, two for each important word category. For the weighting, we apply and compare two methods, presented in the next sections.

Kullback-Leibler Weighting

For the final score, we can use the Kullback-Leibler divergence (relative entropy) [KL51] of P_2 from P_1:

$$D_{KL}(P_1||P_2) = \sum_{x \in X} P_1(x) \log \frac{P_1(x)}{P_2(x)} \tag{7.11}$$

To measure the information about tonality, we can define our tonality scores based on the divergence between the two category pairs:

$$f_{pol}(v_i) = D_{KL}(P(pos|v_i)||P(neg|v_i)) \tag{7.12}$$

$$f_{sub}(v_i) = D_{KL}(P(sub|v_i)||P(neu|v_i)) \tag{7.13}$$

Here, we measure the information lost, if $P(neg|v_i)$ approximates $P(pos|v_i)$, for example. The Kullback-Leibler is an asymmetric measure, so a switch of the distributions would give a different result. This is one reason why we prefer our second method (which is symmetric except in one point), but we evaluate both in order to find out how important the choice of the weighting method is.

Entropy-summand Weighting

Also, the basic idea of the entropy [Sha48] can be applied to extract the importance of the edges for the tonality.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)) \tag{7.14}$$

Here, the $p(x_i)$ refer to the probabilities in the equations 7.6 to 7.9. We add or subtract the entropy-summand of the assumed tonality class for one node v_i to/from a perfect state (normalized to 1 and -1):

$$f_{pol}(v_i) = \begin{cases} 1 + P(pos|v_i) * \log_2(P(pos|v_i)) & \text{if } P(neg|v_i) \leq P(pos|v_i) \\ -1 - P(neg|v_i) * \log_2(P(neg|v_i)) & \text{otherwise} \end{cases} \tag{7.15}$$

$$f_{sub}(v_i) = \begin{cases} 1 + P(sub|v_i) * \log_2(P(sub|v_i)) & \text{if } P(neu|v_i) \leq P(sub|v_i) \\ -1 - P(neu|v_i) * \log_2(P(neu|v_i)) & \text{otherwise} \end{cases} \tag{7.16}$$

In this way, we measure how much disorder one node v_i provides for a certain tonality class. For a clearly positive node (appears only in positive statements), e.g., the disorder will be 0 and so $f_{pol}(v_i) = 1$ and also $f_{sub}(v_i) = 1$.

These functions are not continuous, because $f_{pol}(v_i)$ is 0.5 for $P(pos|v_i) = 0.5$, but the left-hand limit for $P(pos|v_i) \to 0.5$ is -0.5, for example. This is also the reason, why these functions are not completely symmetric. But we think this is an advantage for the final classification, because it separates the classes more strongly.

7.2.3 Final Scores and Classification

To compute the eight final features values (four for each z-class), we calculate the average scores of all nodes, which share the same category, over all sentences of the statement. If no nodes/edges could be recognized in an unseen statement, all features would be zero. We use an SVM[2] to classify the statements by the extracted features. This works according to the one-versus-all strategy for a non-binary classification, which achieved slightly better results than the one-versus-one strategy or a subjective-objective classification first and then a positive-negative classification.

By using only eight features, we actually achieve better results if compared with the use of one edge as a feature, because we abstract from individual word combinations in order to prevent overfitting. We will demonstrate that in section 7.3.1, where this method of using all edges as features is denoted as the **graph edges** method. Another

[2]RapidMiner standard implementation (http://rapid-i.com/)

positive aspect of restricting the number of features to a constant limit is that we save computing time (for the calculation of distances within machine learning, e.g.), because the graphs can be large (cf. section 7.3.1).

7.3 Experiments

7.3.1 Data and Experimental Setup

We use our two datasets for the evaluation: The pressrelations dataset with 1,521 statements (446 positive, 492 neutral, 583 negative) and the Finance dataset containing 8,500 statements (2,125 positive, 2,125 negative, 4,250 neutral) from 5,352 news items.

Again, we use approx. 30% of the statements, that is 420 statements (the first 140 positive, neutral, or negative statements) or 2,500 statements (the first 625 positive or negative and the first 1,250 neutral statements) in order to create our graph (the graph has 41,470 or 154,001 edges, resp.). For POS-tagging, identification of negations, and lemmatisation, we apply the TreeTagger [Sch95]. Unless otherwise stated, 20% of the remaining statements (220 and 1,200 statements) are once more the training set for the SVM and the rest is test set. Again, the size of the test is so large, because we are aiming at a real significance of the solution which can actually be operated in practice.

7.3.2 Adapting the State-of-the-Art Approaches for a German MRA

For the approaches of Ding et al. [DLY08], Wilson et al. [WWH09], and Taboada et al. [TBT+11] we need a sentiment dictionary. Thus, we use the same statements which we use for the creation of our graphs for the creation of a dictionary as one variant.

To create the lexicon of subjectivity clues for the method of **Wilson** et al. [WWH09], all words which appear more often in neutral statements get the *prior polarity* neutral. For all other words, we calculate the number of appearances in positive statements minus the appearances in negative statements divided by all appearances. A positive word has a value of over 0.2, a negative word has a value of less than -0.2 and the rest has the prior polarity *both*. A positive word with a value above 0.6 belongs to the reliability class *strongsubj*, the other positive words are *weaksubj*. We treat the negative words analogously. We use the Stanford Parser for German [RM08] to calculate the dependency trees for the sentences [WWH09], in order to extract the *General Modification Features*, the *Polarity Modification Features* and the *Structure Features*. The lists of intensifiers, copular verbs, modals, negations, and polarity shifters are translated by us. We also added such elements which are not direct translations, but have the same function. The result of this method is a classification of words and phrases. Thus,

for a statement classification, we classify the words of the statements and the class of the most frequently used words is the class of the statement (ambiguous statements are classified as the most frequent class). According to the authors, we apply the best machine learning techniques for the word classification (BoosTexter [SS00] for tonality classification and Ripper [Coh96] for Subjectivity Analysis with parameters as in [WWH09]).

For **Opinion Observer** [DLY08], we also identify neutral words if they appear more often in neutral than in subjective statements and subjective words are positive if they appear more often in positive than in negative statements and vice versa for negative words. In contrast to Opinion Mining in customer reviews, we exchange product features through statements and calculate the orientation of opinions for all statements with their opinion orientation algorithm. For this purpose, we adapt the *Negation Rules*, the *But Clause Rules*, the *Inter-sentence Conjunction Rule*, and the *TOO Rules* for German (by translating important words such as "but" or the negations).

SO-CAL [TBT+11] needs a dictionary with sentiment values from -5 to +5 with intervals of one. Thus, we use the same scores as the Wilson method and a word with a value above 0.818 to 1 gets a sentiment score of +5 and so on. This means, that neutral words also exist. We translated the list of intensifiers (amplifiers and downtoners) and negations, as well as we also added missing elements. The authors propose two approaches for the negation search. We use the second, more conservative approach, because this approach works better according to the authors. Also, we use the value 4 for the negation shift. Furthermore, we implement the algorithm of irrealis blocking and translate the list of irrealis markers (modal verbs, conditional markers, negative polarity items, private-state verbs [TBT+11]).

For all dictionary-based methods (Wilson, Opinion Observer, SO-CAL), we also evaluate an additional variant which use a sentiment dictionary and not the statements which we use to construct the graphs on each fold. We apply the SentiWS [RQH10] for this purpose. As the SentiWS has sentiment values between -1 and 1, we apply similar procedures to construct the method-specific dictionaries as described above: For SO-CAL, it is the same procedure by using the SentiWS values, positive words have a score above 0.33 for Wilson and Opinion Observer, *strongsubj* words have an absolute value above 0.66 and so on. The methods are denoted as *method* (dictionary).

RSUMM [SPV11] needs less specific adaptation, because only a sentence splitter and a tokenizer are needed. So, RSUMM is very language-independent. We test two versions of this method: one includes the optimization step to estimate the best values for X and Y (notated as RSUMM(X%, Y%)) and the other version (RSUMM(100%)) does without this step, because we believe that every sentence is important in the statements and also because more words mean more information about the tonality

in our domain. We use the sets for the creation of the graphs and lexicons as the validation dataset (VDS) [SPV11] and the subjectivity dataset (SDS) [SPV11]. As in [SPV11], we apply the SVMLight package [Joa99] for classification.

Opinion Observer [DLY08] and SO-CAL [TBT+11] do not use supervised learning. Therefore, we have also added the RapidMiner SVM in order to classify the statements based on the scores of Opinion Observer and SO-CAL (as shown in tables with (+ SVM)).

7.3.3 Results

Table 7.4 and 7.6 show the results on the pressrelations dataset and table 7.5 and 7.7 show the results on the Finance dataset. Table 7.4 and 7.5 present the tonality classification (positive, neutral, negative) and tables 7.6 and 7.7 displays the Subjectivity Analysis (subjective, neutral).

Word connections (Entropy-summand) achieve the best results with 63.45% accuracy on the pressrelations dataset (more than 15 percentage points better than Wilson, which is the best of the 'classical' state-of-the-art methods) and best results on the Finance dataset with 65.17% (more than 4 percentage points better than RSUMM(90%,95%), which comes in second). The weighting of the edges through the Entropy-summand performs better than the Kullback-Leibler weighting on both datasets, so we use the Entropy-summand weighting for all further experiments.

Also, the improved methods (RSUMM(100%), Opinion Observer (+ SVM), and SO-CAL(+ SVM)) get better accuracies in the majority of cases (the improvement of SO-CAL is more than 13 percentage points on pressrelations and more than 4 percentage points on Finance, e.g.). Furthermore, the variants of the methods, which are expanded by a general sentiment dictionary, perform rather worse. The 'classical' Opinion Observer performs better with a general sentiment dictionary, while Wilson tends to achieve worse results in this variant (one reason could be the missing of words with the prior polarity *both* in this variant).

Wilson et al. [WWH09] (without an additional dictionary) achieve an accuracy of 42.91% on the pressrelations dataset (Subjectivity Analysis 69.36%) and 48.67% on Finance (Subjectivity Analysis 60.96%) for their word classification. The accuracy of the dictionary variant is 43.44% on pressrelations and 40.12% on Finance. Therefore, the tonality classification by the most frequent word class seems appropriate for this task and method, because this method achieves better results in the classification of statements than on the word level.

The findings of RSUMM are ambiguous. The 'classical' RSUMM with parameter optimization does not perform very well on pressrelations, but it performs well on Finance with a high proportion of sentences and words (RSUMM(90%,95%)). Also,

Method	Accuracy	Positive		Neutral		Negative	
		prec	rec	prec	rec	prec	rec
Wilson	0.4784	0.358	0.5	0.5423	0.5054	0.5540	0.4444
Wilson (dictionary)	0.4609	0.377	0.3366	0.3664	0.2963	0.5346	0.6223
Opinion Observer	0.3806	0.3732	0.1732	0.3481	0.8267	0.6098	0.1693
Opinion Observer (dictionary)	0.4468	0.5083	0.1993	0.4005	0.8693	0.576	0.2822
RSUMM(80%,20%)	0.403	-	0.0	-	0.0	0.403	1.0
SO-CAL	0.3279	0.3676	0.7353	0.2626	0.3551	0.8461	0.0248
SO-CAL (dictionary)	0.2852	0.2987	0.8464	0.2072	0.1307	0.0075	0.0002
Opinion Observer (+ SVM)	0.3825	-	0.0	0.252	0.1084	0.4037	0.8743
Opinion Observer (dictionary + SVM)	0.3235	0.52	0.2122	0.1322	0.0804	0.346	0.6
RSUMM(100%)	0.4801	0.4586	0.3025	0.8298	0.1354	0.4609	0.8789
SO-CAL (+ SVM)	0.4608	0.463	0.3061	0.3543	0.5699	0.6486	0.48
SO-CAL (dictionary + SVM)	0.3995	0.8235	0.0571	0.3559	0.9371	0.6306	0.2
graph edges	0.5482	0.4313	0.551	0.6578	0.5175	0.5831	0.5714
our approach (Kullback-Leibler)	0.5778	0.5	0.302	0.6642	0.6154	0.5534	0.74
our approach (Entropy-summand)	**0.6345**	0.5346	0.4735	0.6989	0.6818	0.6442	0.7086

Table 7.4: Results of the tonality classification on the pressrelations dataset.

Method	Accuracy	Positive		Neutral		Negative	
		prec	rec	prec	rec	prec	rec
Wilson	0.5602	0.4206	0.188	0.6358	0.7329	0.4706	0.5872
Wilson (dictionary)	0.4088	0.3678	0.3291	0.5618	0.339	0.3367	0.6132
Opinion Observer	0.4357	0.3641	0.0947	0.5033	0.713	0.2449	0.222
Opinion Observer (dictionary)	0.4583	0.3275	0.186	0.5325	0.664	0.3404	0.3193
RSUMM(90%,95%)	0.6092	0.4433	0.4840	0.731	0.6145	0.5866	0.7233
SO-CAL	0.3478	0.2992	0.5993	0.384	0.373	0.8519	0.046
SO-CAL (dictionary)	0.2905	0.2669	0.9207	0.4429	0.1203	0.001	0.0007
Opinion Observer (+ SVM)	0.4852	0.3384	0.0914	0.496	0.9269	-	0.0
Opinion Observer (dictionary + SVM)	0.4577	0.3299	0.187	0.5118	0.6649	0.3384	0.3177
RSUMM(100%)	0.6088	0.4428	0.4823	0.731	0.6145	0.5854	0.7233
SO-CAL (+ SVM)	0.3921	0.2986	0.7479	0.4573	0.1074	0.599	0.601
SO-CAL (dictionary + SVM)	0.4762	0.3862	0.341	0.544	0.6206	0.3878	0.3244
graph edges	0.5875	0.4437	0.3633	0.6444	0.7096	0.5816	0.5708
our approach (Kullback-Leibler)	0.561	0.3868	0.5445	0.7659	0.5524	0.5201	0.5951
our approach (Entropy-summand)	**0.6517**	0.53	0.5675	0.7714	0.6527	0.5946	0.7351

Table 7.5: Results of the tonality classification on the Finance dataset.

Method	Accuracy	Subjective		Objective	
		prec	rec	prec	rec
Wilson	0.6818	0.7251	0.8602	0.4970	0.2975
Wilson (dictionary)	0.7029	0.7742	0.8636	0.2871	0.179
Opinion Observer	0.4496	0.7698	0.2724	0.3481	0.8267
Opinion Observer (dictionary)	0.5422	0.8635	0.3885	0.4005	0.8693
RSUMM(80%,20%)	0.3269	-	0.0	0.3269	1.0
SO-CAL	0.5250	0.7373	0.4686	0.3632	0.6449
SO-CAL (dictionary)	0.4378	0.7928	0.235	0.3481	0.8693
Opinion Observer (+ SVM)	0.6061	0.6636	0.8454	0.252	0.1084
Opinion Observer (dictionary + SVM)	0.4109	0.88	0.1479	0.3508	0.958
RSUMM(100%)	0.7083	0.7014	0.9865	0.8298	0.1354
SO-CAL (+ SVM)	0.5153	0.7485	0.4252	0.3702	0.7028
SO-CAL (dictionary + SVM)	0.3598	0.878	0.0605	0.3345	0.9825
graph edges	0.7037	0.6983	0.9882	0.8205	0.1119
our approach (Kullback-Leibler)	0.7662	0.8215	0.8353	0.6449	0.6224
our approach (Entropy-summand)	**0.7707**	0.8478	0.8050	0.6329	0.6993

Table 7.6: Subjectivity Analysis on the pressrelations dataset.

Method	Accuracy	Subjective		Objective	
		prec	rec	prec	rec
Wilson	0.6307	0.6228	0.6649	0.6399	0.5966
Wilson (dictionary)	0.5247	0.5296	0.7944	0.5069	0.2305
Opinion Observer	0.5047	0.508	0.2963	0.5033	0.713
Opinion Observer (dictionary)	0.5405	0.5538	0.417	0.5325	0.664
RSUMM(90%,95%)	0.6919	0.7307	0.6170	0.6630	0.7682
SO-CAL	0.6127	0.616	0.5983	0.6095	0.627
SO-CAL (dictionary)	0.5155	0.5571	0.1513	0.509	0.8797
Opinion Observer (+ SVM)	0.494	0.4665	0.0636	0.496	0.9269
Opinion Observer (dictionary + SVM)	0.5327	0.5732	0.2667	0.5204	0.8003
RSUMM(100%)	0.6975	0.7424	0.6137	0.6654	0.7829
SO-CAL (+ SVM)	0.6231	0.7415	0.3814	0.582	0.8663
SO-CAL (dictionary + SVM)	0.511	0.5481	0.1421	0.5055	0.8822
graph edges	0.6302	0.7821	0.3639	0.5840	0.898
our approach (Kullback-Leibler)	0.7006	0.6753	0.7761	0.735	0.6247
our approach (Entropy-summand)	**0.739**	0.7179	0.7898	0.7649	0.6878

Table 7.7: Subjectivity Analysis on the Finance dataset.

Method	0.05	0.1	0.2	0.4	0.8	210	420	840
Wilson	0.4388	0.4743	0.4784	0.5514	0.5795	**0.5275**	0.4784	0.5553
Opinion Observer	0.3403	0.3683	0.3825	0.3979	0.3591	0.3585	0.3806	0.3822
SO-CAL	0.4579	0.439	0.4608	0.4402	0.4818	0.3509	0.3279	0.2702
RSUMM(80%,20%)	0.4063	0.4046	0.403	0.3949	0.3636	0.3226	0.403	0.4557
RSUMM(100%)	0.2964	0.448	0.4801	0.5265	**0.6318**	0.489	0.4801	0.5529
our approach (Entropy-summand)	**0.5717**	**0.5883**	**0.6345**	**0.6278**	0.5818	0.5224	**0.6345**	**0.6452**

Table 7.8: Different sizes of the training set and the dictionaries/graphs.

if we use all sentences and all features (RSUMM (100%)) we obtain very good results on both datasets. This fits in with our assumption that every sentence of a statement is important and that more words lead to more tonality information. The number of word features for RSUMM(100%) is 4,985 features for one statement on pressrelations and 13,608 features on Finance. After the parameter optimization the size is 974 word features on pressrelations (RSUMM(80%,20%)) and 12,248 features on Finance (RSUMM(90%,95%)).

The outcomes of this study suggest that methods which include machine learning techniques tend to perform better than unsupervised techniques. The results of the approaches which we expand with an SVM support this conclusion. As mentioned before, the graph edges without delexicalization and weighting obtain a not so high accuracy. This shows the importance of the aggregation of the edges and entropy-based weighting.

We evaluate the influence of the different input sizes and so we performed experiments with 5%, 10%, 40%, and 80% training for machine learning as well as 210 and 840 statements for the creation of dictionaries/graphs on pressrelations (0.17% training for 210 statements and 0.32% training for 840 statements in order to create the same size of training according to the results of 420 statements). The results are shown in table 7.8. Opinion Observer and SO-CAL are written in italics, because the results on the left side (size of the training set) belongs to their (+ SVM) variants and the results on the right side are the 'classical' methods with no supervised learning.

These experiments show that our word connections remain very stable if the training set is decreased. However, it does not benefit from more training, especially when the training set is very large (80%). Opinion Observer and RSUMM(80%,20%) has the same problem. Nevertheless, it still receives the second-best results, even if another method gets a higher accuracy. However, in our opinion, it is more important to obtain good results on small training sizes, because over 75% for training would mean that a possible practical implementation would save not much human effort.

7.3.4 Statistical Significance of the Features

We perform a 10-fold cross validation with our method, Wilson (as the best 'classical' state-of-the-art-method) and SO-CAL (+ SVM) on the pressrelations dataset in order to evaluate the contribution of single tonality features. Our approach (Entropy-summand with all features) achieves an accuracy of 61.94%, while Wilson gets 56.36% and SO-CAL 46.68%. As an analogy to Wilson et al. [WWH09], we carry out a two-sided t-test with Wilson and SO-CAL (+ SVM) as baselines. The results are shown in table 7.9. The plus signs indicate a significant increase to the baseline, the minus signs show a significant decrease. For one sign, changes are significant at the level $p \leq 0.1$,

Features	Level(Wilson)	Level(SO-CAL)	Features	Level(Wilson)	Level(SO-CAL)
$T_{v,pol}$	− − − − −	nsc	$T_{v,sub}$	nsc	+ + + + +
$T_{n,pol}$	− − − − −	− − −	$T_{n,sub}$	− − − − −	++
$T_{adv,pol}$	− − − − −	−	$T_{adv,sub}$	− − − − −	nsc
$T_{adj,pol}$	− − − − −	− − − − −	$T_{adj,sub}$	− − − − −	nsc
$T_{cat,pol}$	− − − − −	nsc	$T_{cat,sub}$	−−	+ + + + +
$T_{cat,z}$(all)	+ + + + +	+ + + + +			

Table 7.9: Significance of the tonality features T to the baselines Wilson and SO-CAL.

two signs mean $p \leq 0.05$, three signs $p \leq 0.025$, four signs $p \leq 0.01$ and five signs indicate $p \leq 0.005$. "nsc" stands for no significant change.

As shown in table 7.9, the features with type $z = sub$ are more important than the polarity features. In the categories, the nouns and verbs are more significant than adjectives and adverbs (adverbs are a little stronger in the polarity difference). Combining all features produces a very significant increase against both baselines.

7.4 Conclusion

We have shown that the word connections outperform state-of-the-art-methods in most cases of tonality classification for an MRA. As a major advantage, our approach does not need much training data. The combination of all tonality features is a significant increase against both baselines, too. The findings show that the word connections in combination with the entropy weighting allow to learn the tonality structure of different word combinations accurately, even though the training size is small. This is a major advantage for a solution, which operates in practice for media analysts, which have to analyse articles for an MRA.

Therefore we have finally found an approach for learning the tonality of statements, which is based on combinations of words and produces the best classification accuracies in contrast to all comparison methods under our conditions.

8

Integrating Viewpoints

In this chapter, we want to concentrate on the perspective of statements and examine the viewpoints in given statements. As mentioned before, an MRA requires the viewpoint for each extracted statement. The following statement (translated example from the pressrelations dataset [SCH12]) is negative for the former Federal Minister Röttgen and his party CDU, e.g.:

(8.1) Even the CDU notices Federal Minister Röttgen's careless neglect of the problems in nuclear repository Asse in the meantime. The comments of different CDU politician from the Asse region demonstrate, how this situation is becoming critical and that he "have lacked leadership". (**Code:** negative, CDU)

As we have shown in chapter 4, the datasets of our MRAs have an annotation that the statement has the specified tonality under this viewpoint. Without viewpoints, an MRA does not have enough information. For example, a collection of news documents, which are collected for a vehicle manufacturer, could contain many positive statements. However, most or all statements could only belong to competitors and are only positive for them. So, the vehicle manufacturer as the client of the MRA would not know how it shall assess the results of the analysis without any viewpoint information, because this collection of news could be very negative for the vehicle manufacturer, for example.

We propose an ontology-based algorithm for the determination of viewpoints and analyse perspective features. Furthermore, we explain also two state-of-the-art techniques, called DASA [QHZ+10] and OPUS [GR09], in detail and compare them with our method. This algorithm and the belonging case study was published in the contribution "Integrating Viewpoints into Newspaper Opinion Mining for a Media Response Analysis" [SC12] on the 11th Conference on Natural Language Processing (KONVENS 2012).

8.1 Problem Definitions

As has been pointed out, media analysts annotate extracted statements with a tonality and an assignment of the viewpoint (this could be the client's organisation or a competitor) in an MRA. So, a suitable solution is an automated determination of viewpoints for given statements. But we also want to try a different way: Can we analyse the tonality under a certain perspective? For both tasks, we give a formal definition.

Task Definition IV: Viewpoint Tonality. *Given a statement s which consists of the words w_i with $i \in \{1, ..., s_n\}$, the task is to determine a tonality y and a viewpoint g for the statement s.*

$$t_3 : s = (w_1, ..., w_{s_n}) \mapsto (y, g); y \in \{\text{positive,neutral,negative}\}, g \in \{g_1, ..., g_m\} \quad (8.1)$$

The tonality y could be determined by our tonality classification (cf. last chapter). The m different views are known before the analysis. For example, the pressrelations dataset has $m = 2$ different viewpoints (here $g \in \{\text{CDU,SPD}\}$). The following statement has a positive tonality for the German Chancellor Merkel and her political party CDU, e.g.:

(8.2) That itself illustrates how important it was that Chancellor Merkel has implemented reforms in Europe. (Code: positive, CDU)

This is the way companies in media monitoring code their MRA results. A harder task (even for humans and therefore not the common procedure) is the determination of the polarity from a certain point of view.

Task Definition V: Viewpoint Modified Polarity. *Given a statement s and a viewpoint g, the task is to determine the polarity y_g from the given viewpoint g.*

$$t_4 : (s, g) \mapsto y_g \in \{\text{positive,negative}\} \quad (8.2)$$

In this example, the polarity y_g is modified by a given point of view g. This would solve the problem of different polarities through different viewpoints (cf. chapter 7). For example, a statement that is considered positive from a certain viewpoint A, can be negative for another viewpoint B.

(8.3) The government quarrels, the SPD acts: The time has come to establish legal rules for a quota of women in commercial enterprises. (**Code A:** positive, SPD; **Code B:** negative, CDU)

The SPD as the biggest opposing party in Germany acts in this statement as the political competitor of the CDU. While the point of view should be extracted from the

statement itself for the **Viewpoint Tonality**, this task needs more external or world knowledge (the SPD is a competitor of the CDU, e.g.). Nevertheless, we have seen in the last chapter that this case of two viewpoints with different tonalities does not appear often.

For this task, we leave out neutral statements, because they are not so interesting. Neutral statements would not often change their tonality (neutral for A means also neutral for B or the statements are not relevant for B in many cases). Of course, if we would include neutral examples, we could compare the results better with our tonality classification of chapter 7, but, as in chapter 6, we will start with positive and negative statements and find out, whether the outcome is good enough or whether Task Definition IV is more appropriate for our solution.

In this chapter, we examine these problems in detail. We present a new ontology-based approach for the determination of viewpoints. In addition, we explain viewpoint features which improve current methods in Opinion Mining for statements which are modified through a viewpoint. We evaluate our approach against two state-of-the-art methods. Furthermore, we want to analyse the influence of the viewpoint to the tonality.

This chapter contains the following: In section 8.2, we analyse the related work about perspectives and viewpoints in opinionated texts. This section also includes a pre-evaluation of OPUS [GR09]. We present our viewpoint assignment algorithm and viewpoint features in section 8.3. After that we evaluate our methods on our two MRA datasets in comparison with DASA [QHZ+10], before we give a conclusion in the last section.

8.2 Related Work on Viewpoints

The point of view aspect plays less important role in customer reviews, because a customer expresses only one view (his/her own view) and different viewpoints occur actually only by comparisons of different products [Liu10]. As shown in the examples, the viewpoints are almost essential in news articles and some statements do not have any tonality without a viewpoint.

Nevertheless, far too little attention has been paid to the integration of viewpoints. Only a few publications tackle the problem of viewpoints or perspectives for Opinion Mining. Likewise, many of these approaches do not fit to the task of an MRA.

Devitt and Ahmad [DA07] work with news articles about a company takeover. Their approach computes graph-based features which require a sentiment dictionary (SentiWordNet [ES06]) as well as a common dictionary (WordNet [Mil95]) to create the graph. The nodes are concepts of words and the edges represent relations between

the words in WordNet, but this approach does not handle different viewpoints.

Park et al. [PLS11] extract groups for a certain topic. The groups have contrasting views about this topic. To extract these groups, they identify the speaker of a reported speech object who agrees or disagrees with other speakers or organisations of the same group and the opposing group, respectively. Unfortunately, this approach does not fit in with the requirements of an MRA, because the determination of tonality and not the different opinion groups are interesting for such an analysis. The different groups are commonly known from the beginning, anyway. Thomas et al. [TPL06] headed in the same direction by using a graph-based approach with same speaker links and different speaker agreement links in congressional debates.

8.2.1 Perspective with OPUS

A study [GR09] of Greene and Resnik introduces **OPUS**. OPUS stands for **O**bservable **P**roxies for **U**nderlying **S**emantics. The authors argue that not only the choice of words, but rather the structure of sentences is important for the perspective. Their prominent example is Ronald Reagan's use of the passive form: "Mistakes were made" (due to the Iran-Contra affair) [Bro07]. This syntactic choice is a hint for the perspective.

So, the words get a syntactic role for their analysis. As a consequence, their approach needs syntactic parsing. The resulting features are used by a classification technique (SVM). The following example shows the extraction of the features from a sentence step-by-step:

- Example (taken from [GR09]): The prisoner will murder a guard.
- Results of dependency parsing: nsubj(murder, prisoner); aux(murder, will); dobj(murder, guard)
- Corresponding OPUS features: TRANS:murder, murder:nsubj, nsubj:prisoner, murder:aux, aux:will, murder:dobj, dobj:guard

Greene and Resnik work in their study with an own corpus about the death penalty and the Bitter Lemons corpus, which contains 297 documents about Israeli and Palestinian viewpoints on different topics. We want to start with test runs of OPUS, which was performed by Robert Höck, who worked as a student within the ATOM project. He implemented and evaluated OPUS for his bachelor thesis [Höc12].

He evaluated OPUS [GR09] on the pressrelations dataset, because OPUS is complex in terms of calculation and representation of features and differs strongly in its methodology from the other techniques. At first, we wanted to investigate, if OPUS would achieve convincing results, before we adapt its complex structure to the processing pipeline.

Test No.	Grammar	Separation	OPUS-Blank	Correct	Rateable
1	PCFG	no	no	**0.6465**	0.1302
2	Factored	no	no	0.5556	0.0296
3	PCFG	yes	no	0.6392	**0.4063**
4	Factored	yes	no	0.5821	0.0881
5	PCFG	yes	yes	0.6432	0.2643
6	Factored	yes	yes	0.5862	0.0572

Table 8.1: Results of the OPUS test run [Höc12].

Robert Höck tested several configurations of OPUS (shown in table 8.1). The grammar shows which parameter file and therefore which parsing strategy (PCFG or Factored [KM03, RM08]) was used for the Stanford Parser [RM08], which performed the syntactic parsing. Separation means a separation of tonality and perspective (Robert Höck's test run also covers the tonality). If he used an OPUS-Blank then he filtered out verbs which have an unambiguous tonality and did not provide any information about the viewpoints (the verb "win", e.g.).

But the results of OPUS were not very convincing in this test. Although OPUS reaches an accuracy of 64.65% for the rateable statements, only 40.63% of all statements are rateable and, as a consequence, over 59% of the statements cannot be classified by this technique. This means an accuracy of only 25.97% of all statements. Table 8.1 shows an excerpt of the results on the pressrelations dataset.

8.2.2 Perspective with DASA

Qiu et al. [QHZ+10] propose an approach called **DASA** (**D**issatisfaction-oriented **A**dvertising based on **S**entiment **A**nalysis) for an online advertising strategy. They analyse web forums for an advertising selection based on the consumer attitudes. Although their intention is something different, the basic problem is the same: the extraction of a viewpoint. They use syntactic parsing and a rule-based approach to extract topic words which are associated with negative sentiment to propose products from rivals. We also expand this approach to positive and neutral sentiments to create topic words and use this approach for our comparison.

For the Sentiment Analysis step of the DASA algorithm [QHZ+10], we use the tonality annotation of the datasets to identify the opinion words with the same tonality, because we are more interested in the perspective component than in the Sentiment Analysis component, which identifies basically the negative words of the General Inquirer dictionary [SDSO66].

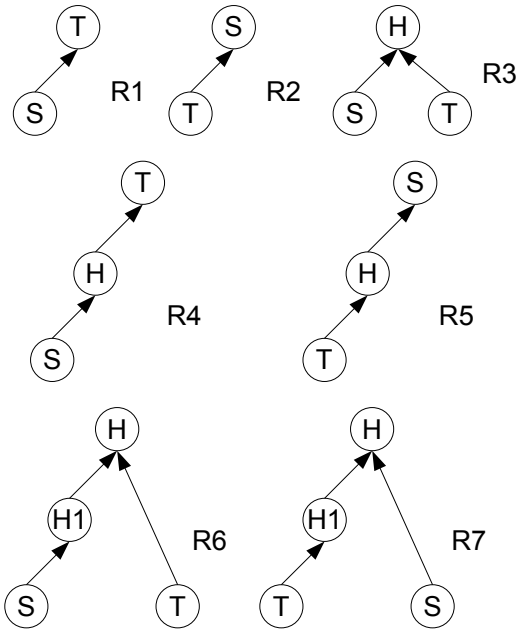

Figure 8.1: The different relations used by DASA [QHZ+10].

We calculate the dependencies between opinion and topic words with the Stanford Parser for German [RM08]. Then DASA uses rules to combine the correct relations between sentiment and topic words. They handle two kinds of relation types: direct relations (R1 to R3) and indirect relations (R4 to R7). All seven rules are shown in figure 8.1. *S* stands for a sentiment word, *T* for a topic word, and an arrow shows a dependency between two words. In direct relations, one word depends directly on the other kind of word or both depend directly on a third word *H*. Indirect relations are more complex. Here, *S* and *T* are connected over a third word *H* or even a fourth word *H1*. We apply the rules R1 to R7 in descending order as described in Qiu [QHZ+10].

Furthermore, we also expand their approach to positive and neutral sentiments. For their application of selecting advertisements, it is very reasonable to search for negative words and then to recommend rivals' products, but we also need a viewpoint for positive and neutral statements. We use the same RDF ontologies (cf. section 8.3.1) to create the input information for the DASA method, so that DASA knows the rivals and can use the entities as topic words for the assignment.

8.3 Viewpoint of Statements

Now, we explain our algorithm for the viewpoint determination and features for viewpoints. For the viewpoint of an MRA it is very important to know which entities (persons, organisations, or products) play a role in a given statement. We propose an ontology-based approach to recognize viewpoints based on entities.

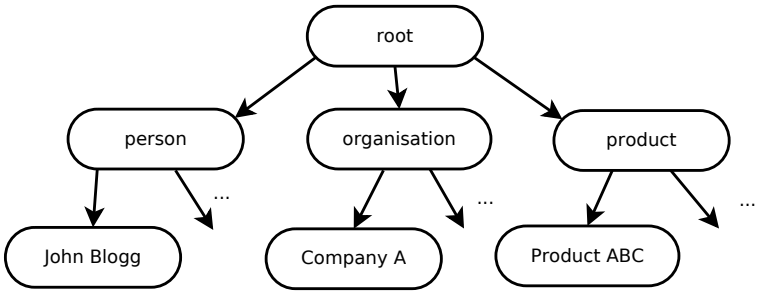

Figure 8.2: A sample ontology.

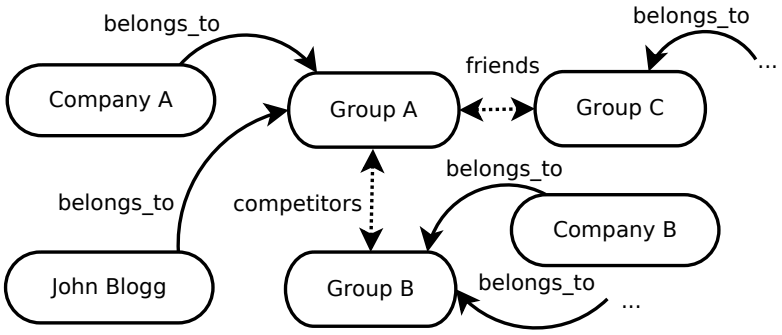

Figure 8.3: Sample ontology relationships.

8.3.1 Ontology-based Approach

Ontologies are very helpful in structuring knowledge. For our ontology-based approach of viewpoint determination, we organise our different entities in a hierarchical structure (shown in Figure 8.2). The entities are persons, organisations, and products. Persons can have an important role in an organisation, e.g. a special function such as press agent or chairman. Organisations can be companies, political parties, and so on. Products can be something the client's companies or the competitors are producing and/or selling.

All these entities have a group attribute. In other words, they belong to one certain group. Also, each group stands in relationship to every other group. These relationships can be neutral, friendly, or competitive. Hence, every entity has one of these relationships to each of the other entities. Figure 8.3 shows an example.

It is not very time consuming to extract this information, because media analysts keep this information in so-called code books. These code books provide information about the organisations, the people, and the products, which the MRA should analyse. It also tells the media analysts about the different viewpoints in the analysis. These pieces of information are used during the human analysis process (as tooltips, e.g.), because it is very hard for humans to remember all the names and relationships between the entities. A code book can contain hundreds of persons, for instance.

8.3.2 Viewpoint Features

The approach considers how often the article mentions friendly entities and how often the competitors are mentioned. As a result, two features represent the persons and two features the organisations/products (**Viewpoint Features** δ).

$$f_{\delta_1}(s) = \frac{pf(s,g)}{p(s)} \qquad f_{\delta_2}(s) = \frac{of(s,g)}{o(s)} \tag{8.3}$$

In equation 8.3, $p(s)$ or $o(s)$ are the numbers of persons or organisations, respectively, in the statement s. $pf(s,g)$ or $of(s,g)$ are the friendly persons or friendly organisations/products, respectively, for group g. Friendly persons could be members of the initiator of the MRA (the managing director of the analysis customer, e.g.) or a member of a cooperation organisation.

$$f_{\delta_3}(s) = \frac{pc(s,g)}{p(s)} \qquad f_{\delta_4}(s) = \frac{oc(s,g)}{o(s)} \tag{8.4}$$

$pc(s,g)$ or $oc(s,g)$ are persons or organisations/products, respectively, of group g's competitors. Friends and competitors are deduced by the relationships in the ontology.

Furthermore, the influenced tonality is stored by **Viewpoint Tonality Features** ϵ, which can apply every word-based method σ (or more precise σ_{method}) such as chi-square or entropy.

$$f_{\epsilon_1}(s) = \frac{1}{|F_w|} \sum_{w \in F_w} \sigma(w) \qquad f_{\epsilon_2}(s) = \frac{1}{|R_w|} \sum_{w \in R_w} \sigma(w) \tag{8.5}$$

F_w and R_w are the sets of words which belong to friend or competitor entities, respectively, and are determined in this way: Our method creates a scope around an entity. All words in the scope have a smaller distance to all other entities (in number of words between them). In the sentence "Merkel is acclaimed for her government's work, so the SPD is still performing relatively poorly in polls." the method would associate "acclaimed" with "Merkel" and "poorly" with "SPD". We avoid syntactic parsing with this simple solution, which creates also the scope for the entities in section 6.5.3 and produces not so bad results.

8.3.3 Determination of the Assignment

To assign a viewpoint for the statements, our algorithm determines the probability of one entity belonging to one group (we use this algorithm also for the friend and

competitor assignment). As a consequence, a statement belongs to one or more specific groups, if the probability is maximal under the assumption that all entities within this statement belong to this group (if the two probabilities are equal and maximal, the statement belongs to two groups and so on).

$$g = \underset{g_i \in \{g_1,\ldots,g_m\}}{\arg\max} P(s|g_i) \tag{8.6}$$

$$P(s|g_i) = \sum_{e \in E_s} P(e|g_i) \tag{8.7}$$

The probability of one statement belonging to one specific group is the sum of all single probabilities with which entity e belongs to group g_i (E_s are all entities in statement s).

For this purpose, an entity of statement s is compared to all entities E_{g_i} which belong to group g_i in the ontology.

$$P(e|g_i) = \max_{e_g \in E_{g_i}} (sim(e, e_g)) \tag{8.8}$$

The similarity function $sim(e, e_g)$ compares the name e of the entity in the text and the name of member e_g of group g. If they are the same, the value is 1.0. If e consists of the same tokens and only one token is different, the value is 0.9 (see the pseudo code in algorithm 6). A token is one part of a name, e.g. the surname of a person. So, a name of a person could, for example, consist of two tokens: The first name and the surname. The method *equals* in algorithm 6 checks, if two tokens have the same string representation (e.g. both names start with "John").

This is useful, when persons are only mentioned by their surname or when a product's or organisation's name is not mentioned in its entirety (company XY → company XY Inc., product ABC → product ABC international).

This also means that two persons, who share the same first name, could have a similarity of at least 0.9. This might sound unpleasant, but we have noticed that the persons are almost always mentioned by their full names in the complete articles, so we apply again our Information Extraction module (cf. section 2.4.2) in order to get the information about the full name.

8.4 Evaluation

8.4.1 Experiment Design

We use our two datasets for the experiments: The first test corpus is the **Finance** dataset (with 4,000 statements). The client of the MRA is a financial service provider

Algorithm 6: Entity Similarity

```
Data: Statement Entity e, Group Entity e_g
Result: Similarity σ
if e and e_g are the same type then
    l_1 ← getListOfTokens(e);
    l_2 ← getListOfTokens(e_g);
    m = max(getSize(l_1), getSize(l_2));
    foreach token t_1 ∈ l_1 do
        foreach token t_2 ∈ l_2 do
            if t_1 equals t_2 then
                m = m − 1;
            end
        end
    end
    σ = 0.9^m;
end
else
    σ = 0.0;
end
```

and the collected articles and statements refer to this company and its four competitors (we explain the different relations between these companies in the following). Our second corpus is the pressrelations dataset [SCH12] which consists of two viewpoints (the two competitive parties: CDU and SPD).

For the **Finance** corpus, we create an RDF ontology by extracting all entities of the code book from customer group A (see table 8.2; all companies' names are made anonymous for reasons of data protection). Group A has four competitors (group B to E) and the groups D and E have a friendly relationship, because D has taken over E in the first few days of the MRA. All other relationships are competitive. For the pressrelations dataset, we create an ontology in which CDU and SPD are competitors. We add all party members of the seventeenth German Bundestag[1] (the German parliament) and add some synonyms of the party and concepts such as "government" or "opposition" as organisations (see table 8.2).

For evaluation of the **Viewpoint Features** δ and **Viewpoint Tonality Features** ϵ, we use 30% of the statements to construct sentiment dictionaries which are weighted by the introduced methods (such PMI, Entropy, Information Gain, and so on) and the remaining statements as training and test set (20% training and 80% test; we use again a small training and big test set to guarantee that this approach will also work in practice). In addition, we use SentiWS [RQH10] as another baseline.

We change the tonality according to viewpoint: the tonality is changed to the neg-

[1] collected from http://www.bundestag.de

Group	Organis.	Persons	Products	all
Finance				
A	6	178	6	190
B	2	58	1	61
C	4	53	2	59
D	5	79	1	85
E	2	16	0	18
pressrelations dataset				
CDU	9	237	0	246
SPD	9	149	0	158

Table 8.2: Size of the evaluation ontologies.

ative, if a statement is exclusively positive for a competitor and negative statements for a competitor become positive. For the classification, we use an SVM (RapidMiner standard implementation with default parameters[2]) which performed better than other machine learning techniques (k-means, Naive Bayes) in this evaluation task. The evaluation shows the results in different combination of the features. So, $\alpha+\delta$ is the combination of set α and the feature set δ and so on and all means the selection of all features ($\alpha+\delta+\epsilon$).

8.4.2 Experiment Results

Table 8.3 and 8.4 show the results of the viewpoint assignment. $|s|$ is the number of statements, c are the correctly assigned statements, w are the ones incorrectly assigned, and nc could not be classified (the probability of all viewpoints is zero or DASA does not find a viewpoint, respectively). The average performance for statements are correctly classified in more than 79% or 80% of cases, less than 15% or 12% are classified incorrectly and over 6% or 8% are not classified at all. This is an improvement about 24 or 22 percentage points against the DASA algorithm.

We use the accuracy evaluation metric for the evaluation of the **View Modified Tonality** (cf. equation 6.17). In contrast to the view assignment, we only use the subjective statements from the pressrelations dataset in this task (1,029 statements). But we use all 4,000 statements from this version of Finance, because all statements are subjective (positive or negative).

Table 8.5 and 8.6 show the statement results of which the tonality is modified by a

[2]Rapid-I: http://rapid-i.com/

Method	$\|s\|$	c	nc	w
DASA				
Viewpoint A	994	0.5613	0.2374	0.2012
Viewpoint B	1173	0.5303	0.2293	0.2404
Viewpoint C	812	0.5123	0.3067	0.1810
Viewpoint D	682	0.5967	0.2038	0.1994
Viewpoint E	339	0.6018	0.2655	0.1327
All	**4000**	**0.5518**	**0.2458**	**0.2025**
our approach				
Viewpoint A	994	0.7606	0.0986	0.1408
Viewpoint B	1173	0.7894	0.0648	0.1458
Viewpoint C	812	0.7906	0.0370	0.1724
Viewpoint D	682	0.8372	0.0440	0.1188
Viewpoint E	339	0.8348	0.0590	0.1062
All	**4000**	**0.7945**	**0.0635**	**0.1420**

Table 8.3: Results of the group assignment on **Finance** in comparison with DASA [QHZ+10].

Method	$\|s\|$	c	nc	w
DASA				
CDU	992	0.5927	0.2258	0.1815
SPD	529	0.5350	0.2155	0.2495
All	**1521**	**0.5726**	**0.2222**	**0.2051**
our approach				
CDU	992	0.8700	0.0766	0.0534
SPD	529	0.6759	0.0964	0.2287
All	**1521**	**0.8021**	**0.0835**	**0.1144**

Table 8.4: Results of the group assignment on **pressrelations** in comparison with DASA [QHZ+10].

Method	α	α+δ	α+ϵ	all
SentiWS	0.5066	0.5678	0.5200	0.5888
PMI	0.6094	0.6450	0.5909	0.6406
Chi-square	0.6275	0.6388	0.6272	0.6281
Entropy	0.6319	0.6378	0.6319	0.6334
Information Gain	0.6328	0.6394	0.6297	0.6397

Table 8.5: Results of view based modified tonality on **Finance**.

Method	α	α+δ	α+ϵ	all
SentiWS	0.5342	0.5259	0.5259	0.6177
PMI	0.6294	0.6077	0.6260	0.6427
Chi-square	0.6694	0.6494	0.6678	0.6761
Entropy	0.5492	0.5993	0.6043	0.6411
Information Gain	0.5159	0.5676	0.6277	0.626

Table 8.6: Results of view based modified tonality on the **pressrelations dataset**.

given point of view. The improvement of the accuracy expands from over 0.5 (Entropy, α+δ combination on Finance) to over 11 percentage points (Information Gain, α+ϵ combination on pressrelations). SentiWS achieves an improvement of over 8 percentage points on both datasets with all features. All methods achieved the best results or at least the second best results, if all features are combined.

In contrast, table 8.7 shows the results using the **Basic Tonality Features** α, if the tonality is not modified through a given view. The results are of course higher, because this task of a not modified tonality is simpler and the results do not provide any information about the viewpoint. But the combination of all viewpoint features approach these results (the PMI method or SentiWS with all features achieved even a better result on Finance or pressrelations, respectively).

8.4.3 Error Analysis

If we examine the statement set of Finance for the group assignment task, the reason for the worst performance of the customer group is the type of the collected statements. The customer is given more attention during an MRA and so articles are collected which do not directly contain known entities, but they include general messages (which should not be included necessarily in an MRA): “Markets are suffering from the consequences of the economic crisis.” This is also the reason why this group has the highest rate of

Method	not mod. **Finance**	not mod. **pressrelations**
SentiWS	0.6455	0.5526
PMI	0.6393	0.6528
Chi-square	0.6848	0.6878
Entropy	0.7006	0.6811
Information Gain	0.6945	0.6945

Table 8.7: Results of both datasets which are not modified through a viewpoint.

not classified statements.

One reason for the better performance of our approach against the DASA algorithm is the entity similarity and assignment of the most likely viewpoint which decreases the number wrongly and especially not classified statements.

Both methods (DASA and our approach) perform only slightly better on pressrelations, although pressrelations has only two different viewpoints, because this area is more characterized by comparative statements. Often articles, which are talking much about the CDU, also mention the SPD and vice versa, what it makes more difficult to extract the correct viewpoint of a statement.

The results of the view modified tonality are not on such a high level. But this is a very hard task even for humans. As a consequence, this way does not represent the state-of-art method to code statements in a real media analysis. Besides, the improvement of the viewpoint features approach the existing solutions of the not modified results which do not provide any information about the viewpoint.

8.5 Conclusion

In conclusion, our ontology based approach provides a tonality based on a specific viewpoint. The group assignment algorithm and the viewpoint features allow the coding of statements into certain groups and tonality mutation based on viewpoint. Although results of the evaluation suggest that both options (viewpoint assignment and viewpoint features) are more or less possible for a view-based approach, we believe that a complete divide and conquer strategy is a more elegant way. We will describe this in more detail in the next chapter, when we sum up our conclusions.

9

Conclusion

In this final chapter, we want to conclude our research concerning Opinion Mining for an MRA. Furthermore, we want to show results of a first test of our solution in the practice. We complete this chapter by introducing starting points for the future work.

9.1 Summarizing Conclusion

In this thesis, we work out the three major challenges of Opinion Mining for an MRA: The statements extraction, the tonality classification, and the determination of the viewpoints.

We have presented a machine learning-based algorithm for the automated extraction of statements in chapter 5. The findings of our profound analysis (cf. section 5.6.3) show that our statements extraction technique pulls out the relevant text parts in news articles very well (over 82% F-Score) and achieves a more than 23 percentage points higher F-score in contrast to all comparison methods.

Our main challenge is the classification of the tonality. As we have noticed in chapter 6 and chapter 7, approaches based on analysing the linguistic context achieve good results (cf. section 6.5.3), but they are also somehow limited in the classification of the tonality (as shown in section 6.6), even if the methods apply detailed algorithms such as SO-CAL [TBT+11] or Opinion Observer [DLY08] or calculate features for machine learning based on profoundly linguistic analyses as Wilson et al. [WWH09] (cf. chapter 7). Maybe the analysis of the linguistic context requires new NLP tools, which provide more than POS-tagging and syntactic parsing. RSUMM [SPV11] shows that a bag-of-words approach needs a larger quantity of training examples, even though the features are well selected (cf. section 7.3). At this point, our graph-based approach

(cf. section 7.2) is considerably better, because it learns tonality information about word combinations more accurately and very fast through its word connections (cf. section 7.3).

The advantages of our divide and conquer policy (cf. section 3.6.3) is also reflected in the determination of viewpoints (cf. chapter 8). Nevertheless, the viewpoint features (cf. section 8.3.2) improve the accuracy. The results are even not far away from the result, when the tonality is not modified through a view (cf. section 8.4.2). But the assignment algorithm provides a concrete viewpoint determination with a high accuracy (approx. 80%, cf. section 8.4.2) and this approach can be combined with our graph-based approach for the tonality classification. In this way, we can integrate also neutral examples. The single results generate additionally a better understanding, because media analysts understand the process and the results more easily, if the results of tonality and viewpoints are not combined as one annotation. The tonality and the viewpoints should be rather provided as two separate pieces of information.

Thus, a two-process solution of calculating tonality and viewpoint fits the way, in which media monitoring companies code their MRA today. Likewise, the viewpoint information itself is valuable, because the estimated viewpoints of statements can be applied to calculate how much documents or whole collections talk about clients of an MRA or competitors. This value can be provided as another MRA result.

9.2 Our Solution in a Practical Environment

We evaluate our solution in a testing environment, in which four media analysts get proposals for the tonality and the viewpoints by our graph-based approach and our algorithm for viewpoint determination. The accuracy of their inter-annotator agreement is 81.8% using the simple accuracy metric, if they do not get any suggestions from the system [Wol12]. They change the proposals of the machine in 36.14% of the cases [Wol12]. But the system can classify a part of the statements with a high accuracy.

RapidMiner[1] calculates a confidence value for the classification. For Naive Bayes, it is simply the probability of the class and, for k-nearest neighbours, it is the number of the k-nearest neighbours with the same class divided by k. For their SVM, the distance between the object and the hyperplane determines the confidence.

The hyperplane of an SVM can be defined as:

$$H(\vec{w}, b) = \{\vec{x} \in FS | 0 = \langle \vec{w}, \vec{x} \rangle + b\} \tag{9.1}$$

The vector $\vec{w}$ represents the normal vector to the hyperplane and $\vec{x}$ is the feature vector from the feature space FS (the vector consists of the eight features $T_{cat,z}$ (cf.

[1] RapidMiner (http://rapid-i.com/)

section 7.2) in our case) and $\langle \vec{w}, \vec{x} \rangle$ is the scalar product of the vectors $\vec{w}$ and $\vec{x}$. b (also called the bias) is the offset of the hyperplane. Then, the distance can be described as:

$$\text{dist}(\vec{x}, H(\vec{w}, b)) = |\frac{1}{\sqrt{\langle \vec{w}, \vec{w} \rangle}} \langle \vec{w}, \vec{x} \rangle + b| \tag{9.2}$$

The confidence for the predicted class c is defined in the following equation:

$$\text{confidence}_c(x) = \begin{cases} \frac{1}{1 + e^{-dist(\vec{x}, H(\vec{w}, b))}} & \text{if } c \text{ is the prediction of } x \\ \frac{1}{1 + e^{dist(\vec{x}, H(\vec{w}, b))}} & \text{otherwise} \end{cases} \tag{9.3}$$

But our task is not a binary problem and, as described in section 7.2.3, the tonality classification applies the one-versus-all strategy. This means that the approach creates three classification models: positive vs. not positive, neutral vs. subjective, negative vs. not negative. So, the confidence for a tonality y is the confidence of the model y vs. not y divided by the sum of all confidence values:

$$\text{confidence}_y(x) = \frac{\text{confidence}_y(x)}{\sum_{i \in Y} \text{confidence}_i(x)} \tag{9.4}$$

Here, $Y = \{\text{positive,neutral,negative}\}$ and, of course, the maximum of the three values determines the predicted class.

If we demand a confidence value of above 0.575 for the predicted class, then 30.6% of the statements can be classified with an accuracy of 75.1%, for example [Wol12]. Or if the confidence should be higher than 0.624, than 19.9% of the statements can be classified by an accuracy of 81.8% [Wol12], which meets the human accuracy exactly. If these findings will be validated in practice, approx. 20% of the statements could be annotated automatically with a human accuracy, which means that a large human effort would be saved.

9.3 Future Work

Future Work could cover a closer look to the final classification step for the tonality. In our experiments we use very different machine learning techniques. But the use of specially adapted techniques for the tonality would be interesting. A Self-Organizing Map (SOM) could be applied for the classification, for instance. As mentioned before, we have obtained good results in predicting the authorship attribution of a document or the genre of a web page by SOMs [SC11]. In small pre-tests, we achieve even good results by using a 3-dimensional SOM for the tonality classification by applying the same classification algorithm as in [SC11]. It seems that feature objects based on word connections for positive, neutral, and negative statements can be separated in this way.

Also, it would be interesting, how neural nets would perform on this task. First tests show an accuracy of under 60% in the determination of the polarity, but maybe we do not have found the best implementation or good parameters for neural nets in our context.

Another part of Future Work could be the creation of a solution for the problem discussed in the first section of chapter 7: Statements can have two or more viewpoints and different tonalities for different viewpoints. Maybe the tonality classification can be expanded to solve this problem, when the system recognizes a statement with more than one viewpoint. For this purpose, the viewpoint determination algorithm could also be adapted, so that it also assigns several viewpoints, if the probability for these viewpoints does not strongly differ from each other. These ideas are not implemented yet, because it promises only a small improvement (less than 0.2% of the statements have more than one viewpoint and more than one tonality in the Finance dataset, e.g.).

In addition, the application in practice requires further research. How does our solution behave in a productive system? Opportunities for an iterative learning are embedded. New edges can be inserted. Edges, which are created a long time ago, can be deleted and/or the weighting of the edges can be adjusted, even by hand. It should be further assessed how new edges can be inserted and how old edges can be forgotten automatically for a long-term learning. And another interesting question concerns the influence of the system to the analysts. Our first test about this question suggests that the analysts tend to agree to the proposals of the system. The accordance with our system is over 6 percentage points higher with proposals [Wol12]. But is this a positive or a negative influence for the overall results of an MRA? This aspect requires further research and analysis.

Moreover, it would be interesting to analyse the progress of the tonality on the application side. The observation of shifts in the tonality over time would be attractive during a more extensive practical use. The earlier mentioned warning systems could be realized in this way (cf. chapter 6). It would be especially interesting, if our approach would be combined with a well working topic tracking, so that conclusions could be drawn about the impact of topics for the results of an MRA.

List of Own Publications

2013

- Thomas Scholz, Stefan Conrad. Opinion Mining in Newspaper Articles by Entropy-based Word Connections. In Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), Seattle, WA, USA, pages 1828-1839, 2013.
- Thomas Scholz. Opinion Mining für verschiedene Webinhalte. In Konrad Scherfer and Helmut Volpers, editors, Methoden der Webwissenschaft, LIT Verlag, Münster, pages 63-81, 2013.
- Thomas Scholz, Stefan Conrad. Extraction of Statements in News for a Media Response Analysis. In Proc. of 18th International Conference on Applications of Natural Language Processing to Information Systems 2013 (NLDB 2013), Manchester, United Kingdom, pages 1-12, 2013.
- Thomas Scholz, Stefan Conrad. Linguistic Sentiment Features for Newspaper Opinion Mining. In Proc. of 18th International Conference on Applications of Natural Language Processing to Information Systems 2013 (NLDB 2013), Manchester, United Kingdom, pages 272-277, 2013.

2012

- Thomas Scholz, Stefan Conrad. Integrating Viewpoints into Newspaper Opinion Mining for a Media Response Analysis. In Proc. of 11th Conference on Natural Language Processing (KONVENS 2012), Vienna, Austria, pages 30-38, 2012.
- Thomas Scholz, Stefan Conrad, Lutz Hillekamps. Opinion Mining on a German Corpus of a Media Response Analysis. In Proc. of 15th International Conference on Text, Speech and Dialogue (TSD 2012), Brno, Czech Republic, pages 39-46, 2012.
- Thomas Scholz, Stefan Conrad, Isabel Wolters. Comparing Different Methods for Opinion Mining in Newspaper Articles. In Proc. of 17th International Conference on Applications of Natural Language Processing to Information Systems 2012 (NLDB 2012), Groningen, The Netherlands, pages 259-264, 2012.

2011

- Thomas Scholz, Stefan Conrad. Style Analysis of Academic Writing. In Proc. of the 16th International Conference on Applications of Natural Language to Information Systems 2011 (NLDB 2011), Alicante, Spain, 2011.

- Thomas Scholz. Ein Ansatz zu Opinion Mining und Themenverfolgung für eine Medienresonanzanalyse (An Approach to Opinion Mining and Topic Tracking for a Media Response Analysis). In Proc. of the 23th GI-Workshop on the Foundations of Databases, Obergurgl, Austria, pages 7-12, 2011.

2009

- Thomas Scholz, Sadet Alcic, Stefan Conrad. Automatic Annotation of Web Images combined with Learning of Contextual Knowledge. In Proc. of the International Workshop on Semantic Multimedia Database Technologies (SeMuDaTe 2009), Graz, Austria, 2009.

2008

- Johanna Vompras, Thomas Scholz and Stefan Conrad. Extracting Contextual Information from Multiuser Systems for Improving Annotation-based Retrieval of Image Data. In Proc. of the ACM International Conference on Multimedia Information Retrieval (MIR 2008), Vancouver, Canada, pages 149-155, 2008.

Bibliography

[ABKS99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *Proc. of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 49–60, 1999.

[AHG99] Saliha Azzam, Kevin Humphreys, and Robert Gaizauskas. Using coreference chains for text summarization. In *Proc. of the Workshop on Coreference and its Applications*, CorefApp '99, pages 77–84, 1999.

[AWCM11] Cem Akkaya, Janyce Wiebe, Alexander Conrad, and Rada Mihalcea. Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proc. of the 15th Conference on Computational Natural Language Learning*, CoNLL '11, pages 87–96, 2011.

[AZ12] Charu C. Aggarwal and ChengXiang Zhai, editors. *Mining Text Data*. Springer-Verlag New York, Inc., New York, NY, USA, 2012.

[BCMP11] Farah Benamara, Baptiste Chardon, Yannick Mathieu, and Vladimir Popescu. Towards context-based subjectivity analysis. In *Proc. of 5th International Joint Conference on Natural Language Processing*, pages 1180–1188. Asian Federation of Natural Language Processing, 2011.

[BES09] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Multi-facet rating of product reviews. In *Proc. of the 31th European Conf. on Information Retrieval*, ECIR '09, pages 461–472, 2009.

[BES10] Andrea Baccianella, Stefano Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of the 7th intl. conf. on Language Resources and Evaluation*, LREC '10, pages 2200–2204, 2010.

[BFK04] Yves Bestgen, Cédrick Fairon, and Laurent Kerves. Un baromètre affectif effectif: Corpus de référence et méthode pour déterminer la

valence affective de phrases. In *Journées internationales d'analyse statistique des donnés textuelles (JADT)*, pages 182–191, 2004.

[BHMM11] Alexandra Balahur, Jesús M. Hermida, Andrés Montoyo, and Rafael Muñoz. Emotinet: A knowledge base for emotion detection in text built on the appraisal theories. In *Proc. of the 16th International Conference on Applications of Natural Language to Information Systems*, NLDB '11, pages 27–39, 2011.

[BMZ11] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1– 8, 2011.

[Bro07] John Broder. Familiar fallback for officials: 'mistakes were made'. *New York Times*, 2007. March 14.

[BSG+09] Alexandra Balahur, Ralf Steinberger, Erik van der Goot, Bruno Pouliquen, and Mijail Kabadjov. Opinion mining on newspaper quotations. In *Proc. of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, pages 523–526, 2009.

[BSK+10] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. In *Proc. of the 7th intl. conf. on Language Resources and Evaluation*, LREC '10, pages 2216–2220, 2010.

[BWC11] Danushka Bollegala, David Weir, and John Carroll. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 132–141, 2011.

[CC08] Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proc. of the conf. on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 793–801, 2008.

[CH89] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proc. of the 27th Annual Meeting of the ACL*, ACL '89, pages 76–83, 1989.

[CHT11] Asli Celikyilmaz and Dilek Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 491–499, 2011.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[CMBT02] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. Gate: an architecture for development of robust hlt applications. In *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 168–175, 2002.

[Coh60] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Coh96] William W. Cohen. Learning trees and rules with set-valued features. In *Proc. of the 13th National Conference on Artificial Intelligence*, AAAI'96, pages 709–716, 1996.

[Col97] Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '97, pages 16–23, 1997.

[Cru02] Alan Cruse. *Lexicology: an international handbook on the nature and structure of words and vocabularies.* Handbooks of Linguistics and Communication Science Series. Mouton de Gruyer, 2002.

[CSSdO09] Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's so easy ;-). In *Proc. of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, TSA '09, pages 53–56, 2009.

[DA07] Ann Devitt and Khurshid Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 984–991, 2007.

[DDH10] Zhendong Dong, Qiang Dong, and Changling Hao. Hownet and its computation of meaning. In *Proc. of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 53–56, 2010.

[DLY08] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proc. of the Intl. Conf. on Web search and web data mining*, WSDM '08, pages 231–240, 2008.

[DLZ09] Xiaowen Ding, Bing Liu, and Lei Zhang. Entity discovery and assignment for opinion mining applications. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 1125–1134, 2009.

[DRF+09] Lorand Dali, Delia Rusu, Blaz Fortuna, Dunja Mladenic, and Marko Grobelnik. Question answering based on semantic graphs. In *Proc. of the Workshop on Semantic Search in Conjunction with the 18th Intl. World Wide Web Conference*, 2009.

[DT09] Weifu Du and Songbo Tan. An iterative reinforcement approach for fine-grained opinion mining. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 486–493, 2009.

[DTCY10] Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proc. of the 3rd ACM intl. conf. on Web search and data mining*, WSDM '10, pages 111–120, 2010.

[DVDM01] Nigel Dewdney, Carol VanEss-Dykema, and Richard MacMillan. The form is the substance: classification of genres in text. In *Proc. of the workshop on Human Language Technology and Knowledge Management - Volume 2001*, HLTKM '01, pages 7:1–7:8, 2001.

[DZVdSVdB03] Walter Daelemans, Jakub Zavrel, Ko Van der Sloot, and Antal Van den Bosch. Timbl: Tilburg memory based learner, version 5.0. Technical report, ILK Research Group, 2003.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, KDD '96, pages 226–231, 1996.

[ERS+09] Shady Elbassuoni, Maya Ramanath, Ralf Schenkel, Marcin Sydow, and Gerhard Weikum. Language-model-based ranking for queries on rdf-graphs. In *Proc. of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 977–986, 2009.

[ES06] Andrea Esuli and Fabrizio Sebastiani. Determining term subjectivity and term orientation for opinion mining. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 193 – 200, 2006.

[Fle71] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[FSSY12] Yi Fang, Luo Si, Naveen Somasundaram, and Zhengtao Yu. Mining contrastive opinions on political texts using cross-perspective topic model. In *Proc. of the 5th ACM international conference on Web search and data mining*, WSDM '12, pages 63–72, 2012.

[FWY+11] Shi Feng, Daling Wang, Ge Yu, Wei Gao, and Kam-Fai Wong. Extracting common emotions from blogs based on fine-grained sentiment clustering. *Knowl. Inf. Syst.*, 27(2):281–302, 2011.

[GKV08] Venkatesh Ganti, Arnd C. König, and Rares Vernica. Entity categorization over large document collections. In *Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 274–282, 2008.

[GL08] Murthy Ganapathibhotla and Bing Liu. Mining opinions in comparative sentences. In *Proc. of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 241–248, 2008.

[GR09] Stephan Greene and Philip Resnik. More than words: syntactic packaging and implicit sentiment. In *Proc. of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 503–511, 2009.

[GWMA09] Yaw Gyamfi, Janyce Wiebe, Rada Mihalcea, and Cem Akkaya. Integrating knowledge for subjectivity sense labeling. In *Proc. of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 10–18, 2009.

[HBF+13] Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. Exploiting emoticons in sentiment analysis. In *Proc. of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 703–710, 2013.

[HC09] Xuanjing Huang and W. Bruce Croft. A unified relevance model for opinion retrieval. In *Proc. of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 947–956, 2009.

[HF12] Louisa Ha and Ling Fang. Internet experience and time displacement of traditional news media use: An application of the theory of the niche. *Telematics and Informatics*, 29(2):177–186, 2012.

[HFC+08] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 179–186, 2008.

[HL04] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proc. of the 10th ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining*, KDD '04, pages 168–177, 2004.

[HM97] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, ACL '97, pages 174–181, 1997.

[Höc12] Robert Höck. Opinion Mining mit OPUS. Bachelor thesis, 2012.

[HPD+08] Ali Harb, Michel Plantié, Gerard Dray, Mathieu Roche, François Trousset, and Pascal Poncelet. Web opinion mining: how to extract opinions from blogs? In *Proc. of the 5th Intl. Conf. on Soft computing as transdisciplinary science and technology*, CSTST '08, pages 211–217, 2008.

[HZM+11] Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. Using cross-entity inference to improve event extraction. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1127–1136, 2011.

[IAH+08] Kentaro Inui, Shuya Abe, Kazuo Hara, Hiraku Morita, Chitose Sao, Megumi Eguchi, Asuka Sumida, Koji Murakami, and Suguru Matsuyoshi. Experience mining: Building a large-scale database of personal experiences and opinions from web documents. In *Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pages 314–321, 2008.

[Jac01] Paul Jaccard. Etude comparative de la distribution orale dans une portion des alpes et des jura. In *Bulletin de la Société Vaudoise des Sciences Naturelles*, volume 37, pages 547–579, 1901.

[JG10] Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1035–1045, 2010.

[JL08] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proc. of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 219–230, 2008.

[Joa99] Thorsten Joachims. Advances in kernel methods. chapter Making large-scale support vector machine learning practical, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[Jon72] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[JYM09] Lifeng Jia, Clement Yu, and Weiyi Meng. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proc. of the 18th ACM Conference on Information and knowledge management*, CIKM '09, pages 1827–1830, 2009.

[KED+07] Vladimir Khoroshevsky, Irina Efimenko, Grigory Drobyazko, Polina Kananykina, Victor Klintsov, Dmitry Lisitsin, Viacheslav Seledkin, Anatoli Starostin, and Vyacheslav Vorobyov. Ontos solutions for semantic web: text mining, navigation and analytics. In *Proc. of the 2nd international conference on Autonomous intelligent systems: agents and data mining*, AIS-ADM'07, pages 11–27, 2007.

[KH06] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proc. of the*

Workshop on Sentiment and Subjectivity in Text, SST '06, pages 1–8, 2006.

[KH07] Soo-Min Kim and Eduard Hovy. Crystal: Analyzing predictive opinions on the web. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 1056–1064, 2007.

[KIM+04] Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. Collecting evaluative expressions for opinion extraction. In *Proc. of the 1st International Joint Conference on Natural Language Processing*, IJCNLP '04, pages 584–589, 2004.

[KK07] Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive collection of html documents. In *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 1075–1083, 2007.

[KKE+08] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami, Pooya Khosravyan Dehkordy, and Asghar Tajoddin. Optimizing text summarization based on fuzzy logic. In *Proc. of the 7th IEEE/ACIS International Conference on Computer and Information Science*, ICIS '08, pages 347–352, 2008.

[KL51] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KLC06] Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. Tagging heterogeneous evaluation corpora for opinionated tasks. In *Proc. of the 5th International Conference on Language Resources and Evaluation*, LREC '06, pages 667–670, 2006.

[KM03] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proc. of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, 2003.

[KMS+10] Milos Krstajic, Florian Mansmann, Andreas Stoffel, Martin Atkinson, and Daniel A. Keim. Processing online news streams for large-scale semantic analysis. In *ICDE Workshops*, pages 215–220, 2010.

[KRKK09] Won Young Kim, Joon Suk Ryu, Kyu Il Kim, and Ung Mo Kim. A method for opinion mining of product reviews using association rules.

In *Proc. of the 2nd Intl. Conf. on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, pages 270–274, 2009.

[KS06a] Moshe Koppel and Jonathan Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109, 2006.

[KS06b] Moshe Koppel and Itai Shtrimberg. Good news or bad news? Let the market decide. In James G. Shanahan, Yan Qu, and Janyce Wiebe, editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 297–301. Springer Netherlands, 2006.

[LBC09] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. Adapting svm for data sparseness and imbalance: A case study in information extraction. *Natural Language Engineering*, 15(2):241–271, 2009.

[LBK09] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proc. 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 497–506, 2009.

[LCDZ11] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proc. of the 20th international conference on World wide web*, WWW '11, pages 347–356, 2011.

[LGW09] Hiep Phuc Luong, Susan Gauch, and Qiang Wang. Ontology learning through focused crawling and information extraction. In *Proc. of the 2009 International Conference on Knowledge and Systems Engineering*, KSE '09, pages 106–112, 2009.

[Lin04] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proc. of the ACL-04 Workshop*, pages 74–81, 2004.

[Liu10] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, 2010.

[LK08] Sungjick Lee and Han-Joon Kim. News keyword extraction for topic tracking. In *Proc. of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management - Volume 02*, pages 554–559, 2008.

[LLA+11] Marina Litvak, Mark Last, Hen Aizenman, Inbal Gobits, and Abraham Kandel. Degext - a language-independent graph-based keyphrase extractor. In *Proc. of the 7th Atlantic Web Intelligence Conference*, AWIC '11, pages 121–130, 2011.

[LLLW05] Baoli Li, Wenjie Li, Qin Lu, and Mingli Wu. Profile-based event tracking. In *Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 631–632, 2005.

[LLX07] Raymond Y. K. Lau, Yuefeng Li, and Yue Xu. Mining fuzzy domain ontology from textual databases. In *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 156–162, 2007.

[LM02] Yong-Bae Lee and Sung Hyon Myaeng. Text genre classification with genre-revealing and subject-revealing features. In *Proc. of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 145–150, 2002.

[LSL+09] Raymond Y. K. Lau, Dawei Song, Yuefeng Li, Terence C. H. Cheung, and Jin-Xing Hao. Toward a fuzzy domain ontology extraction method for adaptive e-learning. *IEEE Trans. on Knowl. and Data Eng.*, 21(6):800–813, 2009.

[MB10] Eugenia Mitchelstein and Pablo J Boczkowski. Online news consumption research: An assessment of past work and an agenda for the future. *New Media & Society*, 12(7):1085–1102, 2010.

[MG05] David Michaelson and Toni L. Griffin. A new model for media content analysis. Technical report, The Institute for Public Relations, 2005.

[Mil95] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[ML10] Arjun Mukherjee and Bing Liu. Improving gender classification of blog authors. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 207–217, 2010.

[MO06] Craig Macdonald and Iadh Ounis. The trec blogs06 collection: Creating and analysing a blog test collection. Technical report, Department of Computer Science, University of Glasgow, 2006.

[Mom12] Saeedeh Momtazi. Fine-grained german sentiment analysis on social media. In *Proc. of the 9th Intl. Conf. on Language Resources and Evaluation*, LREC '12, pages 1215–1220, 2012.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MST94] Donald Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine learning, neural and statistical classification*. Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[MT04] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 404–411, 2004.

[Niv10] Joakim Nivre. Statistical parsing. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, 2010.

[NKM01] Tetsuji Nakagawa, Taku Kudoh, and Yuji Matsumoto. Unknown word guessing and part-of-speech tagging using support vector machines. In *Proc. of the 6th Natural Language Processing Pacific Rim Symposium*, pages 325–331, 2001.

[NLM99] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[OBRS10] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. of the International AAAI Conference on Weblogs and Social Media*, pages 122–129, 2010.

[PK13] Oliver Plauschinat and Florian Klaus. Web Monitoring - Methodik zur Beobachtung von Social Media für die Meinungsanalyse. In Konrad Scherfer and Helmut Volpers, editors, *Methoden der Webwissenschaft*, pages 43–61. LIT Verlag, Münster, 2013.

[PL04] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of the 42nd Meeting of the ACL*, ACL '04, pages 271–278, 2004.

[PL05] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proc. of the 43rd Annual Meeting of the ACL*, ACL '05, pages 115–124, 2005.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[PLS11] Souneil Park, KyungSoon Lee, and Junehwa Song. Contrasting opposing views of news articles on contentious issues. In *Proc. of the 49th Annual Meeting of the ACL: Human Language Technologies - Volume 1*, HLT '11, pages 340–349, 2011.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, 2002.

[PP10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proc. of the 7th conference on International Language Resources and Evaluation*, LREC '10, pages 1320–1326, 2010.

[PSB07] Bruno Pouliquen, Ralf Steinberger, and Clive Best. Automatic detection of quotations in multilingual news. In *Proc. of Recent Advances in Natural Language Processing*, pages 487–492, 2007.

[PZ06] Livia Polanyi and Annie Zaenen. Contextual valence shifters. In *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. 2006.

[QGLS85] Randolph Quirk, Sidney Greenbaum, Geoffry Leech, and Jan Svartvik. *A Comprehensive grammar of the English language*. General Grammar Series. Longman, 1985.

[QHZ+10] Guang Qiu, Xiaofei He, Feng Zhang, Yuan Shi, Jiajun Bu, and Chun Chen. Dasa: Dissatisfaction-oriented advertising based on sentiment analysis. *Expert Systems with Applications*, 37(9):6182 – 6191, 2010.

[QLBC11] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.*, 37(1):9–27, 2011.

[RA12] Seonggi Ryang and Takeshi Abekawa. Framework of automatic text summarization using reinforcement learning. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 256–265, 2012.

[RDF+07] Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. Triplet extraction from sentences. In *Proc. of the 10th International Multiconference Information Society-IS*, pages 8–12, 2007.

[RFM+09] Delia Rusu, Blaž Fortuna, Dunja Mladenic, Marko Grobelnik, and Ruben Sipos. Document visualization based on semantic graphs. In *Information Visualisation, 2009 13th International Conference*, pages 292–297, 2009.

[RM08] Anna N. Rafferty and Christopher D. Manning. Parsing three german treebanks: Lexicalized and unlexicalized baselines. In *Proc. of the Workshop on Parsing German*, PaGe '08, pages 40–46, 2008.

[RQH10] Robert Remus, Uwe Quasthoff, and Gerhard Heyer. SentiWS - a publicly available german-language resource for sentiment analysis. In *Proc. of the 7th Intl. Conf. on Language Resources and Evaluation*, LREC '10, pages 1168–1171, 2010.

[RW03] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proc. of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 105–112, 2003.

[SAC09] Thomas Scholz, Sadet Alcic, and Stefan Conrad. Automatic annotation of web images combined with learning of contextual knowledge. In *Proc. of the 10th International Workshop on Semantic Multimedia Database Technologies*, SeMuDaTe '09, 2009.

[SB12] Sara Botelho Silveira and António Branco. Using a double clustering approach to build extractive multi-document summaries. In *Proc. of the 15th International Conference on Text, Speech and Dialogue*, TSD '12, pages 298–305, 2012.

[SC11] Thomas Scholz and Stefan Conrad. Style analysis of academic writing. In *Proc. of the 16th Intl. conf. on Applications of Natural Language Processing to Information Systems*, NLDB '11, pages 246–249, 2011.

[SC12] Thomas Scholz and Stefan Conrad. Integrating viewpoints into newspaper opinion mining for a media response analysis. In *Proc. of the 11th conf. on Natural Language Processing*, KONVENS '12, pages 30–38, 2012.

[SC13a] Thomas Scholz and Stefan Conrad. Extraction of statements in news for a media response analysis. In *Proc. of the 18th Intl. conf. on Applications of Natural Language Processing to Information Systems 2013*, NLDB '13, pages 1–12, 2013.

[SC13b] Thomas Scholz and Stefan Conrad. Linguistic sentiment features for newspaper opinion mining. In *Proc. of the 18th Intl. conf. on Applications of Natural Language Processing to Information Systems 2013*, NLDB '13, pages 272–277, 2013.

[SC13c] Thomas Scholz and Stefan Conrad. Opinion mining in newspaper articles by entropy-based word connections. In *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1828–1839, 2013.

[Sch94] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of the International Conference on New Methods in Language Processing*, 1994.

[Sch95] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Proc. of the ACL SIGDAT-Workshop*, pages 47–50, 1995.

[Sch11] Thomas Scholz. Ein Ansatz zu Opinion Mining und Themenverfolgung für eine Medienresonanzanalyse. In *Proc. of the 23rd GI-Workshop "Grundlagen von Datenbanken 2011"*, volume 733 of *CEUR Workshop Proceedings*, pages 7–12, 2011.

[SCH12] Thomas Scholz, Stefan Conrad, and Lutz Hillekamps. Opinion mining on a german corpus of a media response analysis. In *Proc. of the 15th International Conference on Text, Speech and Dialogue*, TSD '12, pages 39–46, 2012.

[Sch13] Thomas Scholz. Opinion Mining für verschiedene Webinhalte. In Konrad Scherfer and Helmut Volpers, editors, *Methoden der Webwissenschaft*, pages 63–81. LIT Verlag, Münster, 2013.

[SCW12] Thomas Scholz, Stefan Conrad, and Isabel Wolters. Comparing different methods for opinion mining in newspaper articles. In *Proc. of the 17th Intl. conf. on Applications of Natural Language Processing to Information Systems 2012*, NLDB '12, pages 259–264, 2012.

[SDSO66] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, 1966.

[SEK+07] Yohei Seki, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi Chen, Noriko Kando, and Chin-Yew Lin. Overview of opinion analysis pilot task at ntcir-6. In *Proc. of the Workshop Meeting of the National Institute of Informatics (NII) Test Collection for Information Retrieval Systems (NTCIR)*, pages 265–278, 2007.

[SGS02] Young-Woo Seo, Joseph Andrew Giampapa, and Katia Sycara. Text classification for intelligent portfolio management. Technical Report CMU-RI-TR-02-14, Robotics Institute, Pittsburgh, PA, 2002.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27:379–423, 1948.

[SM08] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proc. of the 2008 ACM symposium on Applied computing*, SAC '08, pages 1556–1560, 2008.

[SM09] Fangzhong Su and Katja Markert. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 1–9, 2009.

[SMDH10] Stefan Siersdorfer, Enrico Minack, Fan Deng, and Jonathon Hare. Analyzing and predicting sentiment of images on the social web. In *Proc. of the international conference on Multimedia*, MM '10, pages 715–718, 2010.

[SPSS10] Marcin Sydow, Mariusz Pikula, Ralf Schenkel, and Adam Siemion. Entity summarisation with limited edge budget on knowledge graphs.

In *Proc. of the 2010 International Multiconference on Computer Science and Information Technology*, pages 513–516, 2010.

[SPV11] Kiran Sarvabhotla, Prasad Pingali, and Vasudeva Varma. Sentiment classification: A lexical similarity based approach for extracting subjectivity in documents. *Inf. Retr.*, 14(3):337–353, 2011.

[SS00] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning*, pages 135–168, 2000.

[ST99] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Proc. of NIPS*, pages 617–623. MIT Press, 1999.

[ST00] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Proc. of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 208–215, 2000.

[Sto97] Philip Stone. Thematic text analysis: New agendas for analyzing text content. In Carl Roberts, editor, *Text Analysis for the Social Sciences*. Lawerence Erlbaum Associates, Mahwah, NJ, 1997.

[STT95] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, IMS-CL, University Stuttgart, 1995.

[SV13] Konrad Scherfer and Helmut Volpers, editors. *Methoden der Webwissenschaft*. LIT Verlag, Münster, 2013.

[SW10] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological on-line debates. In *Proc. of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 116–124, 2010.

[SWY75] Gerard Salton, Andrew Wong, and ChungShu Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[SZ12] Heiner Stuckenschmidt and Cäcilia Zirn. Multi-dimensional analysis of political documents. In *Proc. of the 17th international conference on Applications of Natural Language Processing to Information Systems*, NLDB '12, pages 11–22, 2012.

[TAV06] Maite Taboada, Caroline Anthony, and Kimberly Voll. Methods for creating semantic orientation dictionaries. In *Conference on Language Resources and Evaluation*, LREC '06, pages 427–432, 2006.

[TBP12] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63(1):163–173, 2012.

[TBS09] Maite Taboada, Julian Brooke, and Manfred Stede. Genre-based paragraph classification for sentiment analysis. In *Proc. of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '09, pages 62–70, 2009.

[TBT+11] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, 2011.

[TDB08] Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva. A natural language query interface to structured information. In *Proc. of the 5th European semantic web conference on The semantic web: Research and applications*, ESWC'08, pages 361–375, 2008.

[TG04] Maite Taboada and Jack Grieve. Analyzing appraisal automatically. In *Proc. of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 158–161, 2004.

[TKK+09] Hideki Tanaka, Akinori Kinoshita, Takeshi Kobayakawa, Tadashi Kumano, and Naoto Kato. Syntax-driven sentence revision for broadcast news summarization. In *Proc. of the 2009 Workshop on Language Generation and Summarisation*, UCNLG+Sum '09, pages 39–47, 2009.

[TKMS03] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, 2003.

[TL03] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, 2003.

[TM08] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proc. of the 17th international conference on World Wide Web*, WWW '08, pages 111–120, 2008.

[TPL06] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 327–335, 2006.

[TSK06] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Library of Congress, 2006.

[Tur00] Peter D. Turney. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, May 2000.

[Tur02] Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, 2002.

[TYZ09] Xiangyu Tang, Chunyu Yang, and Jie Zhou. Stock price forecasting by combining news mining and time series analysis. In *Proc. 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 279–282, 2009.

[Uts96] Akira Utsumi. A unified theory of irony and its computational formalization. In *Proc. of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 962–967, 1996.

[Vap82] Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[Vap95] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[VSC08] Johanna Vompras, Thomas Scholz, and Stefan Conrad. Extracting contextual information from multiuser systems for improving annotation-based retrieval of image data. In *Proc. of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval*, MIR '08, pages 149–155, 2008.

[WBR+10] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proc. of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 60–68, 2010.

[WK10] Michael Wiegand and Dietrich Klakow. Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 795–803, 2010.

[WLWL09] Suge Wang, Deyu Li, Yingjie Wei, and Hongxia Li. A feature selection method based on fisher's discriminant ratio for text sentiment classification. In *Web Information Systems and Mining*, volume 5854 of *Lecture Notes in Computer Science*, pages 88–97. Springer Berlin, Heidelberg, 2009.

[WN07] Tom Watson and Paul Noble. *Evaluating public relations: A best practice guide to public relations planning, research & evaluation*, chapter 6, pages 107–138. PR in practice series. Kogan Page, 2007.

[Wol12] Isabel Wolters. ZIM-Kooperationsprojekt ATOM (Projektform KF) Zwischenbericht zum 10.08.2012. Technical report, pressrelations GmbH, 2012.

[WPS10] Henning Wachsmuth, Peter Prettenhofer, and Benno Stein. Efficient statement identification for automatic market forecasting. In *Proc. of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1128–1136, 2010.

[WR05] Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proc. of the 6th international conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 486–497, 2005.

[WWC05] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210, 2005.

[WWH05] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, 2005.

[WWH09] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

[WZHS07] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *Proc. 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 784–793, 2007.

[XP01] Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *Proc. of the 1st intl. conf. on Human language technology research*, HLT '01, pages 1–5, 2001.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of the 14th International Conference on Machine Learning*, ICML '97, pages 412–420, 1997.

[ZLG+11] Lanjun Zhou, Binyang Li, Wei Gao, Zhongyu Wei, and Kam-Fai Wong. Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 162–171, 2011.

[ZWW10] Jianping Zeng, Chengrong Wu, and Wei Wang. Multi-grain hierarchical topic extraction algorithm for text mining. *Expert Syst. Appl.*, 37:3202–3208, 2010.

List of Figures

List of Tables