Hybrid Approaches for Sentiment Analysis

Dissertation zur Erlangung des akademischen Grades
eines Doktors der Philosophie
der Philosophischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Michael Wiegand
aus Witten

Saarbrücken, 2011

Amtierender Dekan:	Prof. Dr. Erich Steiner
Berichterstatter:	Prof. Dr. Dietrich Klakow
	Prof. Dr. Hans Uszkoreit
	Prof. Dr. Iryna Gurevych
Tag der letzten Prüfungsleistung:	21. Januar 2011

Abstract

Sentiment Analysis is the task of extracting and classifying opinionated content in natural language texts. Common subtasks are the distinction between opinionated and factual texts, the classification of polarity in opinionated texts, and the extraction of the participating entities of an opinion(-event), i.e. the source from which an opinion emanates and the target towards which it is directed.

With the emerging Web 2.0 which describes the shift towards a highly user-interactive communication medium, the amount of subjective content on the World Wide Web is steadily increasing. Thus, there is a growing need for automatically processing this type of content which is provided by sentiment analysis.

Both natural language processing, which is the task of providing computational methods for the analysis and representation of natural language, and machine learning, which is the task of building task-specific classification models on the basis of empirical data, may be instrumental in mastering the challenges of the automatic sentiment analysis of written text.

Many problems in sentiment analysis have been proposed to be solved with machine learning methods exclusively using a fairly low-level feature design, such as bag of words, containing little linguistic information. In this thesis, we examine the effectiveness of linguistic features in various subtasks of sentiment analysis. Thus, we heavily draw from the insights gained by natural language processing. The application of linguistic features can be applied on various classification methods, be it in rule-based classification, where the linguistic features are directly encoded as a classifier, in supervised machine learning, where these features complement basic low-level features, or in bootstrapping methods,

where these features form a rule-based classifier generating a labeled training set from which a supervised classifier can be trained.

In this thesis, we will in particular focus on scenarios where the combination of linguistic features and machine learning methods is effective. We will look at common text classification tasks, both coarse-grained and fine-grained, and extraction tasks.

Zusammenfassung

Sentimentanalyse beschreibt die Aufgabe, Meinungen aus natürlich-sprachlichem Text zu extrahieren bzw. deren Inhalt zu klassifizieren. Übliche Teilaufgaben sind die Unterscheidung zwischen sachbezogenem Text und Meinung, die Klassifikation von Polarität (einer Meinung), sowie die Extraktion von Entitäten, die an einer Meinung beteiligt sind, d.h. der Ursprung, von dem eine Meinung ausgeht, und das Ziel, auf das sich eine Meinung richtet.

Mit dem Aufkommen des Web 2.0, das den Übergang des Internets zu einem hochgradig interaktiven Kommunikationsmedium beschreibt, ist parallel auch der Anteil an subjektiven Inhalten im Netz gestiegen. Dadurch wächst logischerweise auch der Bedarf an automatischen Verfahren, die die Aufgaben der Sentimentanalyse unterstützen.

Bei der Bewältigung der automatischen Sentimentanalyse geschriebener Sprache sind sowohl die natürliche Sprachverarbeitung, die berechenbare Modelle für die Analyse und Repräsentation natürlicher Sprache bereitstellt, als auch maschinelle Lernverfahren, die aufgabenspezifische Klassifikationsmodelle auf der Basis von empirischen Daten liefern, hilfreich.

Viele Probleme in der Sentimentanalyse können mit Standardmethoden aus dem maschinellen Lernen, die sich hauptsächlich auf elementares Merkmalsdesign stützen (wie z.B. Bag of Words, die nur sehr begrenzt linguistische Information kodieren), gelöst werden. In dieser Dissertation soll die Nutzbarkeit von linguistischen Merkmalen in unterschiedlichen Teilaufgaben in der Sentimentanalyse untersucht werden. Hierbei stützen wir uns vorwiegend auf Erkenntnisse der natürlichen Sprachverarbeitung. Linguistische Merkmale können in den unterschiedlichsten Klassifikationsmethoden Anwendung finden,

sei es in rein regelbasierten Klassifikationsverfahren, bei denen die Merkmale direkt als Klassifikator kodiert werden, in überwachten Lernverfahren, bei denen diese Merkmale Standardmerkmale (z.B. Bag of Words) ergänzen, oder aber auch in Bootstrappingverfahren, bei denen die Merkmale Bestandteil eines regelbasierten Klassifikators sein können, der ein annotiertes Trainingsset generiert, auf dem wiederum einfache überwachte Klassifikatoren trainiert werden können.

In dieser Dissertation werden wir uns vorwiegend auf Szenarien beschränken, bei denen eine Kombination aus linguistischen Merkmalen und maschinellem Lernen vorteilhaft ist. Wir werden Textklassifikationsaufgaben (sowohl grob-körnig als auch fein-körnig) und Extraktionsaufgaben betrachten.

Acknowledgements

This thesis could not have been completed without the support and direction of many people. I appreciate all the contributions they have made for this thesis.

I am indebted to my advisor, Prof. Dr. Dietrich Klakow, who offered me the opportunity to pursue a PhD. He always found time to provide me with valuable feedback and suggestions during my doctoral studies.

I am thankful to my thesis reviewers, Prof. Dr. Hans Uszkoreit and Prof. Dr. Iryna Gurevych, for their critical reading of the thesis and constructive comments.

I am grateful to my colleagues at the department of Spoken Language Systems for providing a friendly research environment.

My sincere thanks go to Alexandra Balahur, Dr. Grzegorz Chrupała, Sabrina Wilske and Dr. Theresa Wilson for giving me feedback on draft versions on various papers that are part of this thesis. Special thanks go to Dr. Josef Ruppenhofer not only for giving me feedback on my work and discussing with me the complex annotation scheme of the MPQA-corpus but also for proofreading the final version of this thesis.

Likewise, I would like to thank Dr. Caroline Sporleder not only for giving me feedback on my work but also for helping me to establish contact to other experts in natural language processing including sentiment analysis.

I am also grateful to Joo-Eun Feit and Saeedeh Momtazi for annotating the TREC Blog06 data with sentiment information. Moreover, I would like to thank Stefan Kazalski for providing a crawl from the *Rate-It-All* website. I would also like to thank Dr. Yi Zhang and Dr. Ines Rehbein for running their semantic role labelers on the MPQA-corpus. I am, in particular, grateful to Dr. Alessandro Moschitti not only for allowing

me to use his toolkit on SVM convolution kernels (*SVMLight-TK*) but also for giving valuable comments on how convolution kernels can be effectively applied to NLP tasks.

I would also like to thank the members from the International Research Training Group (IRTG) in Language Technology and Cognitive Systems for useful feedback.

I cannot go without sincerely thanking Dr. Jochen Leidner, who supervised my diploma thesis, for encouraging me to pursue a PhD and giving me useful advice even throughout my doctoral studies.

I gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) and the Bundesministerium für Bildung und Forschung (BMBF).

Last but not least, I want to thank my friends and family for their outstanding support throughout the years.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Sentiment Analysis is the task of extracting and classifying opinionated content in natural language texts. With the emerging *Web 2.0* which describes the shift towards a highly user-interactive communication medium the amount of subjective content on the World Wide Web is steadily increasing. Thus, there is a growing need for automatically processing this type of content which is provided by sentiment analysis. Modern search engines or even more sophisticated extraction systems, such as question answering systems need to be adapted in order to be able to process subjective content in addition to factual content. The most imminent components that these applications require are:

- text classifiers distinguishing between
  - subjective and objective texts (i.e. *subjectivity classifiers*)
  - different types of polarity, most prominently, positive and negative polarity (i.e. *polarity classifiers*)
- entity extraction systems for
  - opinion sources (a.k.a. opinion holders)
  - opinion targets

Both natural language processing which is the task of providing computational methods for the analysis and representation of natural language and machine learning which is the task of building task-specific classification models on the basis of empirical data may be

instrumental in mastering the challenges of the automatic sentiment analysis of written text.

Many problems in sentiment analysis have been proposed to be solved with machine learning methods exclusively using a fairly light-weight and task-unspecific feature design, such as bag of words, containing little linguistic information. In this thesis, we examine the effectiveness of linguistic features in various subtasks of sentiment analysis. Thus, we heavily draw from the insights gained by natural language processing.

The application of linguistic features can be applied on various classification methods, be it in rule-based classification, where the linguistic features are directly encoded as a classifier, but also in supervised machine learning, where these features complement basic low-level features, or in bootstrapping methods, where these features form a rule-based classifier generating a labeled training set from which a supervised classifier can be learned.

In this thesis, we will in particular focus on scenarios where the combination of linguistic features and machine learning methods is effective. We consider this incorporation of linguistic heuristics in a machine learning context as a kind of *hybrid* approach. We will look at common text classification tasks, both coarse-grained and fine-grained, and extraction tasks.

1.2. Contributions

This thesis contributes to the following aspects:

- **Supervised Polarity Classification at Sentence Level.** I present a set of features helping to discriminate between positive and negative sentences. Since sentence-level classification suffers more severely from data-sparseness than document-level classification, some more advanced feature engineering than bag of words is required. I focus on two types of features being structural features relying on the sentence structure and knowledge-based features which incorporate polarity lexicons. This work is also described in (Wiegand & Klakow, 2009b).

- **Feature Engineering for Detecting Indefinite Polar Sentences.** I present a set of linguistic features helping to discriminate between definite polar sentences and indefinite polar sentences. These features are tested as part of a rule-based classifier which does not require any training data. In a cross-domain evaluation, the classifier produces a competitive performance to simple machine learning classification using bag of words. This work is also described in (Wiegand & Klakow, 2010c).

- **Topic-Related Polarity Classification.** I present a study on the viability of including topic information to sentence-level polarity classification. In an evaluation on blog data, distance features and other linguistic features modeling the structural relationship between topic and polar expressions (i.e. words containing a prior polarity) are compared. This work is also described in (Wiegand & Klakow, 2009c).

- **Bootstrapping Algorithms for Document-Level Polarity Classification.** I present a cross-domain study on bootstrapping algorithms for document-level polarity classification. I compare two different methods: semi-supervised learning in which classifiers are bootstrapped with the help of at least few labeled data instances and a learning method where the classifiers are bootstrapped with the help of rule-based polarity classifiers. Moreover, for each learning method I will discuss what parameters need to be taken into consideration in order to obtain optimal performance. During that study, we will particularly address the importance of linguistic knowledge and their relevance to classification performance. This work is also described in (Wiegand & Klakow, 2009a, 2010a).

- **Convolution Kernels for Opinion Holder Extraction.** I present how convolution kernels can be tailored to opinion holder extraction allowing fairly complex but also expressive structures, such as parse trees, being directly provided to a learning method rather than manually deriving features from them. I will formulate several kernels using various scopes and levels of information. I will, in particular, show how important the consideration of linguistic insights is for the formulation of ker-

nels and kernel combination. This work is also described in (Wiegand & Klakow, 2010b).

1.3. Outline of the Thesis

Chapter 2: In the second chapter of this thesis, I will give background information to sentiment analysis. I will describe the most important applications for this discipline. Moreover, I will present the main subtasks of this area and describe state-of-the-art methods that are employed in order to solve them. I will also outline the main challenge in sentiment analysis.

Chapter 3: The third chapter focuses on experiments on supervised polarity classification at sentence level using linguistic features.

Chapter 4: In Chapter 4, I will examine a set of linguistic features designed to detect indefinite polarity.

Chapter 5: In the fifth chapter, I will describe experiments on topic-related polarity classification.

Chapter 6: The sixth chapter presents experiments on bootstrapping algorithms for document-level polarity classification.

Chapter 7: The seventh chapter describes how convolution kernels have to be designed in order to use them for opinion holder extraction.

Chapter 8: In the last chapter, I will draw some general conclusions from the results obtained in the previous chapters. I will also show possible directions for future work.

2. Background

2.1. What is Sentiment Analysis?

In this section, I will discuss the notion of *sentiment analysis*. I will first give an intrinsic definition of the expression. Pang and Lee (2008) define sentiment as the:

> reference to automatic analysis of evaluative text and tracking of predictive judgments.

In the research community the expression sentiment analysis is often (almost) synonymously used with *subjectivity analysis* and *opinion mining*.

Subjectivity can be described as a type of *private state* (Wiebe, 1994). A private state is a state that is not open to objective observation and verification (other types are evaluations, emotions or speculations) (Quirk, Greenbaum, Leech, & Svartvik, 1985).

The term *opinion mining* originally had a more restricted meaning. It was mostly understood as web-search (for products) and aggregating opinions about each of them (poor, mixed, good) (Dave, Lawrence, & Pennock, 2003). In recent years, however, the term has been given a more general sense making it hard to distinguish from sentiment analysis (B. Liu, 2006). Pang and Lee (2008) claim that the only difference between these two terms is that they are used by two different communities. While *opinion mining* is mostly used in *information retrieval*, *sentiment analysis* is the preferred term in *natural language processing (NLP)*. Following this trend, I will use the two terms *opinion* and *sentiment* synonymously in the remainder of this thesis.

In summary, one can describe sentiment analysis as the automatic analysis of opinions while opinions (in this thesis) are understood as evaluating and judgmental utterances.

The type of analysis that is going to be considered in this thesis primarily focuses on text classification (i.e. does a text express an opinion or not, and if so, what type of opinion is it) and entity extraction (i.e. given a text expressing an opinion which is the entity that expresses the opinion or which is the entity towards which the opinion is directed).

It should be noted, however, that although there is some general agreement in the research domain on what an opinion is, there are many differences when it comes to the annotation of concrete text. There exists a plethora of different annotation standards and corpora for English for this task (Pang, Lee, & Vaithyanathan, 2002; Wiebe, Wilson, & Cardie, 2003; Hu & Liu, 2004; Ounis, Rijke, Macdonald, Mishne, & Soboroff, 2007; Seki et al., 2007; Stoyanov & Cardie, 2008; Dang, 2009; Kessler, Eckert, Clarke, & Nicolov, 2010; Toprak, Jakob, & Gurevych, 2010). Even though some of these corpora appear to contain common annotation, they are not always compatible when it comes to actually using them (Li, Bontcheva, & Cunningham, 2007).

In the following, I will give an extrinsic definition of sentiment analysis by distinguishing it from related disciplines:

Flame detection is the task of detecting abusive messages (Spertus, 1997). There are similarities to sentiment analysis as flames are usually highly subjective and contain a negative polarity. Thus, flames are just a very specific type of subjectivity.

Hedging is defined as the linguistic means used to indicate a lack of complete commitment to the truth value of a proposition or a desire not to express that commitment categorically (Hyland, 1998). Thus, hedging is similar to subjective language in that neither of them can be assigned a truth value. Unfortunately, there are only few attempts to discriminate these two terms. Medlock and Briscoe (2007) state that the domain of interest between the two concepts differs. Hedging is mostly examined on scientific articles, in particular, on the biomedical domain (Light, Qiu, & Srinivasan, 2004; Medlock & Briscoe, 2007; Kilicoglu & Bergler, 2008) whereas sentiment analysis is carried out on the most diverse forms of text, most predominantly news (Wiebe et al., 2003) and reviews (Pang et al., 2002). We assume that due to these different domains the phenomena in focus vary. While in scientific texts mostly neutral subjective texts, such as

Sentence (2.1) play an important role, in sentiment analysis there is also much work done on subjective texts containing a value judgment, such as Sentence (2.2).

(2.1) *I believe* that the causes of increasing natural catastrophes *can* be ascribed to global warming.

(2.2) I *find it irresponsible* that some people still deny the existence of global warming given the notable increase of natural catastrophes in recent years.

Affect computing deals with the design of systems that can recognize human emotions (Picard, 1997). While sentiment analysis is usually restricted to verbal utterances, emotions can also be expressed on several other modes. As far as verbal utterances are concerned, there is no universal agreement upon the distinction between emotions and sentiment. A common distinction is that an emotion is a state of mind (Sentence (2.3)) whereas a sentiment or opinion is an evaluation or judgment towards some entity (Sentence (2.4)).

(2.3) I am happy.

(2.4) I think that X is nice.

Another definition suggests that sentiment is an umbrella term that includes both emotions (as a state of mind) and evaluations or judgments (Wilson, 2008b). I will follow the second definition since the corpora I use have been annotated according to that notion.

Creative language, such as humour, irony, idioms, proverbs, puns, and figurative language, bears some similarity to subjectivity in the sense that they often coincide (Wiebe, Wilson, Bruce, Bell, & Martin, 2004), however creative language (e.g. irony) is only a means to express subjectivity or a side-effect of it. Though the interrelation between these two items might appear to be compelling to look into in a thesis about linguistic aspects of sentiment analysis, I will mostly neglect this issue, since the computational approaches towards the detection of creative language is still in their infancy (Sarmento, Carvalho, Silva, & Oliveira, 2009).

2.2. Applications of Sentiment Analysis

Rather than being justified on its own, sentiment analysis is a task that can be used in several applications. Given that the web is currently the resource containing the greatest amount of publicly available opinions, it comes as no surprise that many of these applications are related to the web.

One of the most prominent applications are *search engines* which instead of merely retrieving any web content that is topically related to a query just retrieve subjective content. Ideally, the user formulating the query should even be able to specify the target polarity of subjective content that is to be retrieved.

One step beyond such an opinion-related search engine would be an opinion *question answering* system. While in traditional factual question answering an answer snippet to a natural language question, such as Question (2.5), is extracted, an opinion question answering system should be able to answer questions asking for entities that are involved in an opinion, such as Question (2.6). In addition, similar to definition questions which ask for general information about a specific topic, such as Question (2.7), opinion-based definition questions, such as Question (2.8), i.e. questions asking about the general sentiment towards a particular topic, should be answered.

(2.5) When was Mozart born?

(2.6) Who likes Mozart's music?

(2.7) Who is Mozart?

(2.8) What do people think about Mozart?

The scenario that is represented by the latter question type is of course very similar to the task that is performed by opinion-related search engines; unlike the other opinion question type (Question (2.6)), statements rather than entities are to be returned for this type. Depending on how the output for such a question is to be formatted, the task might also become very similar to opinion-related summarization, as a user may just want the essence of the general sentiment towards a topic and not the mere concatenation of

actual relevant texts that could be found (as it might be much too verbose and, thus, difficult to grasp).

Another major type of applications for sentiment analysis are tools for *social media monitoring*. By that one understands systems that observe a particular part of the web for a longer period of time and try to detect new developments on these data. With regard to sentiment analysis this could mean observing the public opinion (as represented by a certain part of the web) towards a particular item. Such a monitoring system might be attractive for businesses that want to observe the impact of their products on the market. It should enable the detection of early signs of discontent allowing the businesses to take counteraction at a very early stage preventing a negative sentiment regarding their products to spread. Similarly, political institutions, like political parties in a general election might be interested to obtain an immediate feedback on their latest campaign.

Finally, sentiment analysis may also be used as an additional filter in *recommendation systems* to exclude content receiving too much criticism from being recommended. This additional filter might be useful since the algorithms applied to select items to be recommended are usually not based on sentiment analysis but on the similarity of user behavior/profiles.

2.3. Different Subtasks in Sentiment Analysis

In this section, I will provide an overview of the different subtasks in sentiment analysis.

2.3.1. Text Classification

The most prominent subtasks in sentiment analysis are the two text classification tasks which I call in this thesis *subjectivity detection* and *polarity classification*. (Note that in the literature other terms may be used for these tasks.) By subjectivity detection, I mean the distinction between objective texts (Sentence (2.9)) and subjective texts (Sentence (2.10)).

(2.9) The car is red.

(2.10) The car looks horrible.

By polarity classification, I define the classification of texts according to different polarity types. The most common types are positive polarity (Sentence (2.11)) and negative polarity (Sentence (2.12)). Further types are neutral polarity (Sentence (2.13)) and indefinite polarity (Sentence (2.14)). The difference between the latter two categories is that while in neutral polarity there is no value judgment conveyed by the statement, in indefinite polarity there is a value judgment conveyed but the polarity is neither definite positive nor definite negative. In many publications, these two categories are omitted. Neutral polarity is omitted as it may not be considered subjective as in (Pang & Lee, 2004). Indefinite polarity is omitted as it is usually less frequently observed than the other categories.

(2.11) The food is delicious.

(2.12) The food tastes awful.

(2.13) I believe that the food is specially imported from Asia.[1]

(2.14) The food is so-so. *(It is neither good nor bad.)*

In this thesis, I will – unlike some previous work on that task, such as (Wilson, Wiebe, & Hoffmann, 2005) – ignore the class of neutral polarity (see Sentence (2.13)) as the text to be classified will contain value judgments.

In recent years, a two-stage classification has been established. One usually decides whether a text is subjective or not (i.e. one applies subjectivity detection) and if the text is subjective one also classifies its polarity (Pang & Lee, 2004). A distinction between these two types of classification is useful since different features are relevant for these two types (Karlgren, Eriksson, Täckström, & Sahlgren, 2010). Another justifying reason is that there are text types where only one type of classification is necessary, e.g. in review classification a subjectivity detection is superfluous since (at least at document level) all reviews are usually subjective.

[1] Note that this type of *polarity* could also be interpreted as *hedging*.

These types of text classification can also be applied on various levels of granularity. The common levels are:

- document level
- sentence level
- word level

Note that the classification at word level can also be referred to as classification at *expression level* or *phrase level*. We will use these three terms interchangeably in this thesis. In this text classification task, expressions are classified in their respective contexts. The classification of expressions in isolation, e.g. the prediction of whether a word is subjective or has a specific polarity type, is another task which (in this thesis) is called *lexicon induction* and will be discussed in Section 2.3.2.

The need for classification on more fine-grained levels than document level can be explained by the fact that sentiment is not uniformly spread throughout a single document. For information extraction systems (like those presented in Section 2.2), which need to identify the sentiment towards a specific entity, it is therefore vital to be able to compute focused sentiment information, i.e. the information from a sentence or a clause with the mentioning of that entity. Another usage for fine-grained sentiment analysis is that it can be used for improving coarse-grained classification (i.e. classification at document level) (Pang & Lee, 2004; McDonald, Hannan, Neylon, Wells, & Reynar, 2007).

Subjectivity Detection

There has been fairly little work at document-level subjectivity detection. Most work on document-level subjectivity detection is usually restricted to blog-posts (Chesley, Vincent, Li Xu, & Srihari, 2005; Ounis et al., 2007; Ounis, Macdonald, & Soboroff, 2009) as these documents are fairly short and tend to be either fully subjective or objective. In contrast to polarity, the overall degree of subjectivity of a document is less relevant for applications in NLP than that of a sentence or a phrase.

Most text classifiers constructed for sentiment analysis are models trained by supervised machine learning classifiers. Various types of features for these classifiers have been explored. Bag of words offer good performance on an in-domain evaluation (Dias, Lambov, & Noncheva, 2009). Improvements can usually be achieved by adding features describing predictive classes of words, such as particular types of adjectives and verbs (Wiebe et al., 2004; Breck, Choi, & Cardie, 2007; Dias et al., 2009) or task-specific lexicons containing subjective expressions or patterns. They can be manually constructed (Wiebe & Riloff, 2005) or automatically generated (Wiebe et al., 2004; Riloff & Wiebe, 2003). Even substituting hypernym synsets from WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) for words helps (Breck et al., 2007). The usage of these predictive classes has also been shown to be an effective means to overcome domain-mismatch problems encountered when bag of words features are used (Dias et al., 2009). Structural features taking syntactic information into account can also improve performance (Wilson et al., 2005; Karlgren et al., 2010). Recently, there have also been research efforts showing that word sense disambiguation improves subjectivity detection (Wiebe & Mihalcea, 2006; Akkaya, Wiebe, & Mihalcea, 2009).

Polarity Classification

For polarity classification the effectiveness of different types of features varies depending on the level of granularity that is considered. On document level (again we consider machine learning classifiers), the majority of research suggests that bag of words perform well (Pang et al., 2002; Salvetti, Reichenbach, & Lewis, 2006), in particular when bigrams and trigrams are added to unigrams. They also outperform more advanced linguistic features using syntactic word dependency information (Ng, Dasgupta, & Arifin, 2006).

In comparison to document-level polarity classification, more linguistic features have been examined on sentence-level and word-level polarity classification. Several works address syntactic structures, mostly compositionality of phrases and clauses (Moilanen & Pulman, 2007; Choi & Cardie, 2008; Thet, Na, Khoo, & Shakthikumar, 2009). Some of these works focus on particular compositional constructions, such as conjunctions (Meena

& Prabhabkar, 2007; Ding & Liu, 2007; Agarwal, T.V., & Chakrabarty, 2008) or conditional clauses (Narayanan, Liu, & Choudhary, 2009). For some languages, such as Chinese, using morphological features, i.e. features modeling the relationship between several morphological units instead of lexical or phrasal units, has also been shown to be effective.

The most predictive cues in polarity classification are *polar expressions*, i.e. words containing a prior polarity, such as *excellent*$^+$ and *awful*$^-$. These expressions can be directly converted to a rule-based classifier (Kennedy & Inkpen, 2005; Klenner, Petrakis, & Fahrni, 2009; Velikovich, Blair-Goldensohn, Hannan, & McDonald, 2010) or be used as features in a machine learning classifier complementing bag-of-words features. This combination is, in particular, effective on sentence and word level (Wilson et al., 2005; Wiegand & Klakow, 2009b; Choi & Cardie, 2009).

Another crucial aspect of polarity classification is *negation modeling*. If a polar expression occurs within the scope of a negation expression, then the polarity of the opinion is reversed:

(2.15) The waiter in that restaurant was [not polite$^+$]$^-$.

There is no consensus on what features perform best on this task. While Karlgren et al. (2010) suggest that only negation features are relevant, Gamon (2004) comes to the conclusion that it is a plethora of different types of linguistic features.

Please note that in the context of polarity classification, we will not consider polar expressions as linguistic features in this thesis. By linguistic features, we understand features derived from general linguistic properties, such as part-of-speech information or syntactic parse trees. Polar expressions are some task-specific lexical features which are regarded as a separate category.

2.3.2. Task-Specific Lexicons

As pointed out in the previous section, text classification tasks in sentiment analysis benefit from task-specific lexicons containing subjective/polar expressions. Though there are several manually created resources (Stone, Dumphy, Smith, Ogilvie, & associates,

1966; Wilson et al., 2005; Bloom, Stein, & Argamon, 2007), there has also been some work on automatically inducing them.

One popular strand of methods makes use of general lexical resources, such as WordNet, and applies some semi-supervised learning scheme relying on some initially labeled seed words in order to generate a lexicon (Esuli & Sebastiani, 2006a, 2006b, 2007; Rao & Ravichandran, 2009). Another strand of methods applies similar techniques to large unlabeled corpora (Turney & Littman, 2003; Velikovich et al., 2010). The lack of structure is compensated by relying on high-precision statistics, such as *point-wise mutual information*, between seed words and candidate words. These restrictive measures only work since the corpora that are used, such as the World Wide Web, are extremely large and contain a considerable amount of redundancy.

Linguistic patterns, such as exploiting the coordination of seed words as a means of finding lexical units with a similar meaning (Hatzivassiloglou & McKeown, 1997) or some language specific heuristics (Zagibalov & Carroll, 2008), have also been employed for lexicon induction.

2.3.3. Entity Extraction

There are two entity extraction tasks in sentiment analysis, being *opinion holder* and *opinion target extraction*:

(2.16) [Koizumi]$_{opinion\ holder}$ maintains [a clear-cut collaborative stance]$_{opinion}$ towards [the U.S.]$_{opinion\ target}$.

The opinion holder is the source from which an opinion emanates whereas the target is the entity towards which the opinion is directed.

Extracting opinion-related entities can be regarded as an information extraction task. It can also considered as a specific subtype of semantic role labeling if one considers an opinion as a predicate or an event whose arguments are opinion holder and opinion target (Bethard, Yu, Thornton, Hatzivassiloglou, & Jurafsky, 2004; Choi, Breck, & Cardie, 2006; S.-M. Kim & Hovy, 2006).

Chapter 7 will discuss this subtask (including a short overview of related work) in more detail focusing on opinion holder extraction.

2.3.4. Other Tasks

Recently, there has been an increasing interest in sentimental text classification using additional types of categories than the two discussed in Section 2.3.1 (Kudo & Matsumoto, 2005; Somasundaran, Wilson, Wiebe, & Stoyanov, 2007; Kobayakawa et al., 2009). The most detailed study is a work on attitude classification (Somasundaran et al., 2007), in which polarity[2] is distinguished from *agreement*, *arguing*, *speculation*, and *intention*. Another trend is sentiment classification on other forms of communication, such as conversation (Wilson, 2008a; Raaijmakers, Troung, & Wilson, 2008; Somasundaran, Namata, Wiebe, & Getoor, 2009). These types require a notably different analysis than the conventional sentiment classification on plain monologues. In dialogues, for example, utterances may not necessarily be composed of complete sentences but just fragments. Unlike monologues, such as news texts, these utterances cannot be properly analyzed in isolation, i.e. without some consideration of their respective contexts. Therefore, a segmentation of the text into dialogue acts is required for a successful opinion analysis (Somasundaran et al., 2009).

There has also been some considerable work on adapting sentiment text classifiers to new domains as there are many domains for which no annotated sentiment corpora exist. The methods that have been applied are structural corresponding learning (Blitzer, Dredze, & Pereira, 2007), variations of semi-supervised learning algorithms (Aue & Gamon, 2005; Tan, Cheng, Wang, & Xu, 2009), and algorithms combining domain-independent rule-based classifiers and domain-specific supervised machine learning classifiers (Andreevskaia & Bergler, 2008; Tan, Wang, & Cheng, 2008; Tan et al., 2009; Qiu, Zhang, Hu, & Zhao, 2009).

Born out of a similar need has been multilingual sentiment analysis, i.e. the task of automatically migrating sentiment resources or tools from one language to another (Hiroshi,

[2]Polarity is referred to as sentiment in this work.

Tetsuya, & Hideo, 2004; Mihalcea, Banea, & Wiebe, 2007; Banea, Mihalcea, Wiebe, & Hassan, 2008; Banea, Mihalcea, & Wiebe, 2008; Brooke, Tofiloski, & Taboada, 2009).

Another major strand in research in sentiment analysis is the joint modeling of sentiment text classification (primarily polarity classification) and target extraction of opinions, or more precisely aspects of the targets (i.e. the properties of the targets that are addressed):

(2.17) I [don't like]$^{-}_{opinion}$ [the design]$_{aspect}$ of [the new iPod]$_{target}$.

A typical scenario in which this task is evaluated is the classification of polarity of product features (Dave et al., 2003; Hu & Liu, 2004; Popescu & Etzioni, 2005; B. Liu, Hu, & Cheng, 2005; Bloom, Garg, & Argamon, 2007). A related task that jointly models the detection of opinions and opinion holders has also been explored (Choi et al., 2006).

Several research efforts have been made addressing the unsupervised (or weakly supervised) learning of specific aspects of targets (Mei, Ling, Wondra, Su, & Zhai, 2007; Snyder & Barzilay, 2007; Du & Tan, 2009; Somasundaran & Wiebe, 2009) since, in many realistic scenarios, the aspects are not known in advance. Attempts to use the relation between target and opinion to (solely) improve polarity classification have also been made (Mullen & Collier, 2004; Brooke & Hurst, 2009; Nowson, 2009).

As far as information retrieval is concerned, there has also been some work on enhancing search engines with sentiment information (Eguchi & Lavrenko, 2006; M. Zhang & Ye, 2008; He, Macdonald, He, & Ounis, 2008; Gerani, Carman, & Crestani, 2009; Santos, He, Macdonald, & Ounis, 2009; J. Kim, Li, & Lee, 2009; F. Liu, Li, & Liu, 2009). This research has been most prominently enforced by the benchmark competitions TREC Blog (Ounis et al., 2007; Ounis, Macdonald, & Soboroff, 2008; Ounis et al., 2009) and TAC Opinion Question Answering (Dang, 2009).

2.4. The Main Challenge in Sentiment Analysis

There is one major challenge in sentiment analysis that concerns (almost) every single subtask in that discipline. I call it the context-dependency of sentiment information. In

virtually all subtasks of sentiment analysis, sentiment information is conveyed by some (textual) cues. The problem of these cues is that they are ambiguous. I will exemplify this on several word-level tasks:

In subjectivity detection, one needs to have a means of distinguishing between contexts in which a potential subjective expression, such as *alarm*, is subjective (Sentence (2.18)) from contexts where it is objective (Sentence (2.19)).

(2.18) His alarm grew.

(2.19) The alarm went off.

In polarity classification, one needs to detect whether a polar expression, such as *like*, undergoes a contextual modification that will change its polarity or at least its polar intensity. Instead of a plain occurrence of a polar expression (Sentence (2.20)), the expression can be negated (Sentence (2.21)), intensified (Sentence (2.22)), or diminished (Sentence (2.23)).

(2.20) I like it.

(2.21) I don't like it.

(2.22) I very much like it.

(2.23) I quite like it.

Moreover, in entity extraction, such as opinion holder extraction, one needs to find out whether a mention of an entity, such as *government*, serves as the opinion holder of a sentiment expression (Sentence (2.24)) or not (Sentences (2.25) and (2.26)).

(2.24) The government approves of the proposal.

(2.25) The government has been dissolved.

(2.26) The public mainly approves of the new government.

To a great extent, these types of ambiguity can be resolved by considering the *textual* context of the words to be classified. Consequently, these issues can be addressed by

methods from NLP. It is precisely these kinds of phenomena that are addressed in this thesis.

There are, however, other types of context-dependencies that address extra-textual issues. For example, Sentence (2.27) cannot be recognized as a negative statement towards a particular novel, since the sentiment information is not lexicalized.

(2.27) I threw the latest Harry Potter novel out of the window.

It requires cultural knowledge to interpret the act of throwing a novel out of a window as indicative of a negative opinion. This type of sentiment information, also known as *pragmatic opinion* (Somasundaran & Wiebe, 2009), is not considered in this thesis due to the complexity of this phenomenon and the brittleness of state-of-the-art NLP methods to model pragmatic knowledge.

3. Feature Design for Sentence-Level Polarity Classification

3.1. Introduction

This chapter presents feature design for sentence-level polarity classification. Though polarity classification has been extensively explored at document level, fewer research efforts have been made at sentence level although the task is an established research problem (Matsumoto, Takamura, & Okumura, 2005; Meena & Prabhabkar, 2007; Agarwal et al., 2008; Narayanan et al., 2009).

Sentiment information is not evenly distributed across a document. Not only do documents usually comprise both subjective and objective sentences but also the polarity of subjective sentences within a document varies. Thus, sentence-level classification can be used to improve document-level classification (McDonald et al., 2007). Moreover, for tasks demanding fine-grained text analyses, such as question answering or text summarization, sentiment classification at sentence level seems more appropriate than document classification.

Even though a sentence is shorter than a document, a sentence itself may contain several polar expressions. We assume that for those cases, there is always one prominent polar expression. For those cases, the *overall polarity* will be the polarity of that polar expression. For examples, there are several polar expressions in Sentence (3.1). The polar expression *successfully* is the prominent expression. In this chapter, we are exclusively interested in the overall polarity of a sentence.

(3.1) [***Although*** he had *difficulties*$^{-}$]$_{other}$, [he *successfully*$^{+}$ managed the job in the

end]$_{main}$..

Due to the small number of words within a sentence, polarity classification at sentence level differs substantially from document-level classification in that resulting feature vectors encoding sentences tend to be much sparser. Therefore, a classifier trained on bag of words performs worse than at document level.

Fortunately, there is a plethora of linguistic features by which a word can be described within a sentence. We consider features, such as *part-of-speech information*, *clause types*, *depth of word constituents*, or *WordNet hypernyms*. At document level, these features have hardly been used. In general, the benefit of these features remains controversial since their extraction is computationally expensive (many of these features require linguistic pre-processing such as part-of-speech tagging or even syntactic parsing) and their contribution in terms of performance is fairly limited since bag-of-words classifiers already pose a robust baseline.

We show that explicit polarity information and a set of simple linguistic features can significantly improve a standard bag-of-words classifier. We also show that a standard classifier can already be significantly improved by linguistic features in the absence of any polarity information.

Using the established division between subjectivity detection and polarity classification (see also Chapter 2), we consider polarity classification as a binary classification task. That is, we assume that each sentence to be classified is subjective. We neglect the distinction between objective and subjective content since this classification is usually solved independently (Pang & Lee, 2004; Ng et al., 2006). Our experiments are carried out on a subset of the MPQA-corpus (Wiebe et al., 2003).

The work presented in this chapter is also described in (Wiegand & Klakow, 2009b).

3.2. Related Work

The most closely related work to this are (Wilson et al., 2005; Choi & Cardie, 2008) which determine the polarity of individual polar expressions using linguistic features. This word-

level task is solved with supervised machine learning methods. The crucial difference to these works is that we attempt to determine the *overall polarity* of a sentence (see Section 3.1) rather than the local contextual meaning of each individual polar expression. Sentence-level polarity classification has the benefit that it can harness features derived from sentence structure displaying some form of prominence that cannot be used for expression-level classification (e.g. we consider different clause types, the main predicate of a sentence, and the depth of word constituents). In expression-level classification, one needs to determine the polarity of *all* polar expressions rather than only the most prominent one. Unlike (Wilson et al., 2005; Choi & Cardie, 2008), we also examine in how far linguistic features improve a bag-of-words feature representation in the absence of any polarity information.

Kudo and Matsumoto (2005) consider polarity and modality classification at sentence level in Japanese. Improvement of a bag-of-words feature set is achieved on both tasks using n-grams based on dependency paths.

Moilanen and Pulman (2007) present a symbolic approach using deep linguistic information. The evaluation is done on headlines and noun phrases but not on complete sentences. The method is not compared with a baseline machine learning approach (e.g. using bag of words) either. A similar compositional approach using more shallow linguistic information is presented in (Klenner et al., 2009). Again, the method is not compared with a baseline machine learning approach.

Some research efforts looking into particular sentence-level constructions for polarity classification have also been attempted. While Meena and Prabhabkar (2007) and Agarwal et al. (2008) deal with conjunctions, Narayanan et al. (2009) examine conditional clauses.

At document level, Gamon (2004) looks at a large set of linguistic features. The performance is increased, but no definite feature subset can be determined to be effective. Karlgren et al. (2010) suggest, on the other hand, that only negation features are relevant. Matsumoto et al. (2005) and Ng et al. (2006) present syntactically motivated features, most of them based on dependency path information. Though some improve-

ment can be achieved with these features, Ng et al. (2006) also show that higher-order n-grams are virtually as effective in terms of performance as these linguistic features.

3.3. Data

As the dataset for our experiments, we decided to use a subset of the MPQA-corpus (Wiebe et al., 2003) since the corpus is known to have a fairly high inter-annotation agreement. Since the polarity annotation within the MPQA-corpus is not at sentence level but expression level, we had to extrapolate the annotation to sentence level. The procedure we apply is similar to the procedure to generate sentence-level subjectivity data presented in (Wiebe & Riloff, 2005). Expressions either labeled as *direct subjective* or *expressive-subjectivity* with attitude-type *positive* or *negative* were identified as polar expressions. The projection to sentence level is straightforward if the annotated polar expressions within one sentence have the same polarity. Sentence (3.2), for example, illustrates the case where there are two expressions with polarity information, which are both negative. Therefore, the overall polarity of the sentence is also negative.

(3.2) Their cause was *an unjust one*$^-$ and therefore had *little support*$^-$.

Of course, there are a lot of sentences in which there are expressions with differing polarity. We manually annotated these sentences (approximately 30% of the final subcorpus we built). Sentence (3.3) illustrates the case where there are two expressions with different polarity. However, the overall polarity is not mixed. There is a clear preponderance of the second expression which is negative. Therefore, the overall polarity of the sentence is negative.

(3.3) "The international community can *support*$^+$ us so far, but it can *never remove the shackles of repression*$^-$", he said.

Moreover, there are also sentences where the overall polarity is mixed as well:

(3.4) African observers *generally approved*$^+$ of his victory while Western governments *denounced*$^-$ it.

The number of sentences with mixed polarity is so small that including it for our classification task was not possible. The final corpus we produced was down-sampled to equal class sizes. It contains 2,934 sentences in total.

3.4. Feature Design

In this work we distinguish between two types of knowledge-based features: *polarity features* and *linguistic features*. The linguistic features have been formulated at two levels: *sentence level* and *word level*. Polarity features have only been formulated at sentence level. Table 3.1 lists all sentence-level features and Table 3.2 all word-level features.

3.4.1. Prior Polarity Features

We use the Subjectivity Lexicon from the MPQA-project (Wilson et al., 2005) as it is fairly large compared to other publicly available lexicons. We consider the polarity values *positive*, *negative*, and *neutral*.[1] Moreover, the lexicon distinguishes between *strong* entries (e.g. *wonderful* or *hideous*) and *weak* entries (e.g. *valid* or *bulky*). We exploit this additional information in separate features.

3.4.2. Linguistic Features

A specific linguistic feature at sentence level refers to the overall amount of polar expressions within a sentence whereas linguistic features at word level describe for each word whether or whether not a certain linguistic property holds for it in the context of a particular sentence. For example, if we consider the linguistic property *verb* (one of the *part-of-speech* types explained below), the corresponding features at sentence level are *number of positive verbs*, *number of negative verbs*, and *number of neutral verbs* (within this sentence), whereas the features at word level are for each word *x*: *is x a verb?* (in this sentence). The benefit of using these two levels is that we have both coarse-grained

[1] We ignored the value *both* since there are only very few entries with that label (approximately 0.25%).

Table 3.1.: List of sentence-level features.

Bare Polarity Features
number of positive/negative/neutral expressions
number of strong positive/negative/neutral expressions
number of weak positive/negative/neutral expressions
Linguistic Features
number of positive/negative/neutral nouns
number of positive/negative/neutral verbs
number of positive/negative/neutral adjectives
number of positive/negative/neutral adverbs
number of positive/negative/neutral other (part-of-speech tags)
is main predicate positive/negative/neutral expression?
number of positive/negative/neutral expressions within main predicate phrase
number of positive/negative/neutral expressions with depth level I
number of positive/negative/neutral expressions with depth level II
number of positive/negative/neutral expressions with depth level III
number of positive/negative/neutral expressions with depth level IV
number of positive/negative/neutral expressions with depth level V
number of positive/negative/neutral expressions in main clause
number of positive/negative/neutral expressions in other clause
number of positive/negative/neutral expressions in weak clause
number of positive/negative/neutral expressions in strong clause
number of positive/negative/neutral expressions modified by intensifier
number of positive/negative/neutral expressions modified by positive expression
number of positive/negative/neutral expressions modified by negative expression
number of positive/negative/neutral expressions modified by neutral expression
number of positive/negative/neutral expressions in modal scope
number of negated positive/negative/neutral expressions

Table 3.2.: List of word-level features.

Linguistic Features
is word a noun/verb/adjective/adverb/other?
add hypernym synsets of word
is word the main predicate?
is word within main predicate phrase?
has word depth level I/II/III/IV/V?
is word within main/other clause?
is word within weak/strong clause?
is word preceded by intensifier?
is word within modal scope?
is word negated?

and fine-grained features. Since all features at word level are independent of polarity information[2], we can also evaluate the impact of structural features which do not take polarity information into account. We consider the following linguistic aspects:

Part-of-Speech Information

The predictability towards polarity varies throughout different parts of speech. Many polarity lexicons, for example the one presented in (Nasukawa & Yi, 2003), contain mostly adjectives. This means that this part-of-speech tag is more important for polarity classification than others (i.e. a polar adjective may be more predictive than a polar noun). Apart from that, part of speech may also be exploited for some basic word sense disambiguation which can be of help in polarity classification since some important polar expressions are ambiguous. For example, the word *like* can either be a polar verb or just a preposition. In the latter case, the word is not relevant for the polarity classification. In order not to add too much sparse information (in particular with regard to features at word level), we only consider the five part-of-speech tags *noun*, *verb*, *adjective*, *adverb*,

[2]Note that, on the other hand, all sentence-level features carry polarity information.

and *other*.

WordNet Hypernyms *(only used at word level)*

The WordNet ontology (Miller et al., 1990) allows words to be generalized to a certain extent. Our features are inspired by Scott and Matwin (1998). For each word in a sentence we add all the hypernyms of its synset.[3] In a sentence-level classification task, the situation that a word is observed in the test set but has not been observed in the training set usually occurs significantly more often than in corresponding document-level classification tasks. The purpose of using WordNet is that words which have not been observed in the training set (but in the test set) hopefully possess hypernyms that have also appeared in the training set. Thus, a sparse distribution of words is compensated for by a less sparse distribution of hypernyms. A similar usage of WordNet has already been shown to work effectively for subjectivity detection (Breck et al., 2007).

Main Predicate & Main Predicate Phrase

We assume that words within a sentence which have a prominent role from a structural perspective are also important words for polarity classification. In this respect, the main predicate of a sentence is of particular importance. We deliberately did not restrict ourselves to verbs since predicative adjectives (*the book is **interesting***) seem to be at least equally important. Sentence (3.5) displays a case where the polarity of the main verb *support*, which is positive, corresponds to the overall polarity of the sentence. The majority of polar expressions, however, is negative. The main predicate feature which is only active on *support* should outweigh the other polar expressions within the sentence with an appropriately learned feature weight.

(3.5) The Pakistani government *supports*$^+$ President Bush and his *war*$^-$ on *terror*$^-$.[4]

[3] In order to avoid word sense disambiguation, we always map a word onto the first synset in the list of its potential synsets. The first synset usually corresponds to the most frequent sense.

[4] It is certainly debatable whether *war* and *terror* should be regarded as polar expressions or as a part of the multi-word expression *war on terror* in which the words *war* and *terror*, though having a prior

Apart from a feature referring exclusively to the main predicate, we also introduce a more general feature for the entire main predicate phrase, i.e. the entire verbal or adjectival phrase. This should allow polar modifiers within the predicate phrase to be included as well:

(3.6) The president of the National $Trust^{+5}$ [acted $unlawfully^{-}]_{predicate\ phrase}$.

We did not consider common grammatical functions (of a predicate) for separate features, such as *subject* or *object*, because we assume that these entities are less likely to carry polar information (e.g. these grammatical functions are usually occupied by opinion holders and opinion targets).

Depth of Word Constituents

In addition to the previous feature which defines prominence on the basis of grammatical functions (which is fairly restrictive), we also introduce a more general feature which is not bound to any grammatical information. We assume that the depth of a word constituent within a syntax tree (i.e. the length of the path from the leaf node to the root node) can be regarded as another indicator as to how prominent the word is within a sentence. The deeper a constituent is embedded, the less prominent it is and, therefore, the less meaningful it should be for polarity classification. In order to avoid too sparse features we restrict ourselves to five depth levels defined in Table 3.3.

Clause Type

We consider syntactic-based and discourse-based clause types. By syntactic-based type, we distinguish between *main clause* and *other clause* (i.e. adverbial clauses, relative clauses etc.). We assume that words within the main clause of a sentence are more predictive to the overall polarity of a sentence than words in other clause types. By

polarity, lose their polar meaning. As we do not have the resources to robustly recognize multi-word expressions, we will consider these words as polar expressions.

[5]We convert each character to its lowercase equivalent. Therefore, the distinction between *Trust* as part of a named entity and *trust* as a common noun or full verb gets lost.

Table 3.3.: Definition of the different depth features.

Feature	**Description**
level I	constituents with depth ≤ 5
level II	constituents with depth ≤ 10
level III	constituents with depth ≤ 15
level IV	constituents with depth ≤ 20
level V	constituents with depth > 20

discourse-based types, we also make use of features inspired by Meena and Prabhabkar (2007) which denote the presence of strengthening discourse connectives (e.g. *but*) and weakening connectives (e.g. *although*).

Both feature types are illustrated by Sentence (3.7). The polarity of the main clause is also the overall polarity. The strength of the polarity of the subordinate clause is decreased by the presence of the weakening discourse connective *although* and by the fact that this is an *other clause*. In Tables 3.1 and 3.2 these clauses are referred to as *weak* and *strong clauses*.

(3.7) [***Although*** he had *difficulties*$^{-}$]$_{other}$, [he *successfully*$^{+}$ managed the job in the end]$_{main}$.

We refrained from defining more specific clause types, e.g. enumerating each subordinate clause, since it would have created extremely sparse features.

Intensifiers

Intensifiers are adjectives and adverbs which strengthen the meaning of words. For example, a word, such as *good*, should obtain a higher weight in a sentence if it is modified by an intensifier, such as *extremely*. We took the intensifiers from (Wilson et al., 2005). Note that we use this feature also as a word-level feature. A classifier trained on word-level features only (i.e. without the knowledge of polar expressions) might still learn that expressions modified by an intensifier are important since the likelihood of

these expressions being polar (in the scope of an intensifier) is quite high.

Modification of Polar Expressions by Other Polar Expressions *(only used at sentence level)*

Polar expressions can modify each other. The consequence of this is that there is a change in the overall meaning. If the polarity of both expressions is the same, there is an intensification (this is similar to the phenomenon described with the previous category type). If the polarity is different, there might be a weakening in strength or even a shift in polarity of the polar expression being modified. The latter phenomenon is illustrated in the following sentence:

(3.8) Korea has *rejected*$^{-}$ the framework *agreement*$^{+}$.

Since the positive expression *agreement* is modified by the negative expression *rejected*, the overall meaning is negative. This sentence also shows that the modifying relation is a long-range relationship that can hardly been captured by higher-order n-grams. This feature only operates at sentence level, since it refers to polar expressions which are not considered at word level.

Modal Scope

If an utterance appears within a modal scope[6], semantically, it is not bound to be true. For polar expressions, we assume that words within modal scope are less important than they usually are. Consider, for example, the positive expression *like* in Sentence (3.9) which is modified by the modal verb *might* and thus semantically weakened.

(3.9) He ***might*** *like*$^{+}$ the book, but I'm not sure.

Negation Scope

Usually, if a word, or more precisely a statement, appears within the semantic scope of a negation, its meaning is reversed. Apart from using standard negation expressions, such

[6]We define the *scope* of constituent x as the set of all constituents which are dominated by the least common ancestor of x.

as *no*, *not*, or *never*, we also add *polarity shifters* (Wilson et al., 2005). Polarity shifters are weaker than negation markers in the sense that they only reverse polarity. They only change one particular polarity type. For instance, the positive shifter *abate* only turns negative polar expressions into positive polar expressions (as in $abate^{+}$ *the* $damage^{-}$). Likewise, the negative shifter *lack* turns positive polar expressions into negative polar expressions (as in $lack^{-}$ *of* $talent^{+}$).

3.5. Experiments

The results of the following experiments are reported on the basis of a 10-fold cross-validation. We evaluate the results using Accuracy, Precision, Recall, and F-Measure (see also Appendix A.1). Feature selection was carried out on the training data of each partitioning during the cross-validation in order to obtain an unbiased set of features. Statistical significance is reported on the basis of a paired t-test with 0.05 as the significance level. We used *SVMLight* (Joachims, 1999a) with its standard configuration (linear kernel) for SVMs. All linguistic features were extracted from the output of Charniak's parser (Charniak, 2000).

3.5.1. Bag-of-Words Feature Set (Baseline)

Following Pang et al. (2002), we encoded all bag-of-words features as binary features indicating the presence (or absence) of a feature in a sentence. In order to define a strict baseline, we need to find out what subset of bag of words performs best. We tested various amounts using χ^2 feature selection (Yang & Pederson, 1997) and found that the best feature set is the one using all words occurring in the training data. This means that a feature selection on this dataset is superfluous.

The average Accuracy using the entire set of words occurring in the training dataset with no further normalization than described above is 67.2%. By using the lemmatizer within *WordNet* we increase the performance by approximately 1.4% to 68.6%. (The size of the unlemmatized feature set with approximately 9,100 tokens is reduced by

approximately 2,000 tokens when lemmatization is used.) Comparing this with results of polarity classification at document level, e.g. Pang et al. (2002) report 82.9% on movie reviews using similar features, suggests that polarity at sentence level is much harder and that there is much more room for improvement given this low-performing baseline.

3.5.2. (Linguistic) Word-Level Features

The first extension of the standard feature set we look into are the linguistic word-level features (see Table 3.2), none of which contains any polarity information. Since polar expressions vary across different domains and common polarity lexicons only capture a unique polarity of polar expressions, the linguistic word-level features should give us a realistic estimate of how good domain-independent features are.

In order to see which features improve the performance of the bag-of-words feature set, we add each feature category (for all words) separately to the standard feature set and measure the increase in performance. We also apply χ^2 feature selection on each separate feature set. Table 3.4 shows the result of this experiment. The table displays the benefit when the optimal feature size is used. We only display the results of the feature types where we could measure a (notable) increase in performance. Clearly *depth of constituents* is the predominant feature with a contribution of 2.1%. *Part of speech*, *clause type*, and *WordNet hypernyms* are very similar in their strength. All features with exception of *main predicate (phrase)* are significantly improving the bag-of-words baseline. We were very surprised that *negation* did not notably increase the baseline performance. However, Pang et al. (2002) also report only negligible improvement.

The upper part of Table 3.5 contrasts the word-level feature set with the other bare bag-of-words feature sets. We applied χ^2 feature selection to the entire linguistic word-level feature set. The classifier using all bag of words and the optimal subset of all linguistic features (i.e. 6,000 additional features) outperforms the simplest baseline classifier by 5.9% which is clearly significant and still 4.5% better than the lemmatized bag-of-words feature set. The linguistic word-level features are the only features in our experiments where a feature selection produced a significantly better performance than using the

Table 3.4.: Benefit of individual word-level feature type categories *(optimal feature size)* when added to bag of words.

Feature Type	**Optimal Size of Feature Set**	**Benefit (Accuracy)**
depth of constituents	2000	+2.1%*
part of speech	2000	+1.3%*
clause type	1000	+1.2%*
WordNet hypernyms	1000	+1.1%*
main predicate (phrase)	1000	+0.8%

*: significantly better than lemmatized bag-of-words baseline on the basis of a paired t-test using $p < 0.05$

entire feature set. The Accuracy of the complete feature set (with approximately 26,000 active features) is more than 2% worse than the optimal feature set.

3.5.3. Sentence Level: Polarity and Linguistic Features

The lower part of Table 3.5 shows the result of the classifiers using different sentence-level feature sets. A classifier only trained on the prior polarity features (see Table 3.1) already achieves 70.4% Accuracy. If we add all linguistic sentence-level features (see also Table 3.1), we obtain an increase in performance by 3.4%. This shows that these remaining sentence-level features encode other important information than the bare prior polarity features.

In order to find out which features are most discriminative and additive at sentence level, we do a best-first forward selection. Unlike χ^2 feature selection, forward selection has the advantage of selecting features encoding disjunct information.[7] The feature selection on the sentence-level features did not significantly improve performance. After all, there are far fewer features in this feature set (less than 100 features) than in the previous word-level feature set (26,000 active features) and, therefore, less noise is expected to be in that feature set. Table 3.6 displays the result of this feature selection. As far

[7]Please note that we could not use this feature selection method for the word-level features since it would have been computationally prohibitive.

Table 3.5.: Performance of different feature sets.

Feature Sets using no Polarity Information					
Features	*Class*	*Rec.*	*Prec.*	*F.*	*Acc.*
bag-of-words (*not lemmatized*)	+	72.9	65.5	69.0	67.2
	−	61.5	69.5	65.2	
bag-of-words	+	63.2	71.0	66.8	68.6
	−	74.1	66.8	70.3	
bag-of-words + linguistic word-level features	+	68.2	75.8	71.7	**73.1**
	−	78.8	71.0	74.4	
Feature Sets using Polarity Information					
Features	*Class*	*Rec.*	*Prec.*	*F.*	*Acc.*
prior-polarity	+	68.0	71.5	69.7	70.4
	−	72.9	69.6	71.1	
prior-polarity + linguistic sentence-level features	+	70.9	75.2	72.9	73.8
	−	76.6	72.6	74.5	
prior-polarity + bag of words	+	74.0	76.1	75.0	75.4
	−	76.8	74.8	75.7	
prior-polarity + bag of words + linguistic word-level features	+	74.6	78.0	76.2	76.7*
	−	78.9	75.7	77.2	
prior-polarity + bag of words + linguistic sentence-level features	+	74.9	77.9	76.3	76.8*
	−	78.7	75.9	77.2	
prior-polarity + bag of words + all linguistic features	+	75.2	78.8	76.9	**77.5***
	−	79.7	76.3	78.0	

*: significantly better than prior-polarity + bag of words on the basis of a paired t-test using $p < 0.05$

as linguistic features are concerned, the results are similar to the feature analysis of the word-level features. The fact that adjectives are the most important part-of-speech tag was to be expected (see discussion above). It is no surprise either that only depth levels I and II occur in the optimal feature set since these two levels usually denote a high level of prominence. With the occurrence of *main predicate*, *main predicate phrase*, and *main clause*, our analysis proves that syntactically prominent constituents within a sentence can be effective features for polarity classification.

Adding lemmatized bag of words instead of the other sentence-level features results in an even higher improvement by 5% to 75.4% showing that bag of words and the prior polarity features are complementary and extremely additive. This number, however, may be optimistic since the polarity lexicon we are using does not have to have such a high coverage on other domains.

Finally, we test in how far we can increase the performance of a feature set comprising prior polarity information and bag of words. Performance is increased by adding either the remaining sentence-level features or word-level features. Adding either set of features results in a statistically significant improvement by 1.3% and 1.4%, respectively. When both levels are added, the gain in performance by 2.1% is even higher. Comparing this number with the simplest feature set we used (i.e. bag of words - *not lemmatized* in Table 3.5) we have an increase by 10.3%.

3.5.4. Other Levels of Representation

We tested two alternative types of feature representations: *bigrams* and *tree-kernels*. However, all these features did not improve the performance of our baseline. Bigrams can be a means of capturing more local structure than unigrams and are known to improve the quality of polarity classification at document level (Ng et al., 2006). We assume that this representation does not work at sentence level due to the greater data-sparseness. The potential of tree-kernels is that structural features are automatically (implicitly) computed and do not have to be explicitly defined. (A detailed introduction will be

Table 3.6.: Best sentence-level features according to best-first forward selection.

Bare Polarity Features
number of positive/negative expressions
number of strong positive/negative expressions
Linguistic Features
number of positive/negative adjectives
number of negative verbs
number of positive/negative expressions with depth level I
number of positive/negative expressions with depth level II
is main predicate a positive expression?
number of negative expressions in predicate phrase
number of positive/negative expressions in main clause
number of positive expressions modified by positive/neutral expressions

given in Chapter 7.) We used SVMLight-TK (Moschitti, 2006b)[8] for our experiments. The reason for the lacking improvement might be due to too much irrelevant information encoded in syntax trees beside the relevant information as the one that is represented by the linguistic features presented in this chapter. In Chapter 7, we will show that for another task, namely opinion holder extraction, tree kernels work quite effectively. One key premise for the application of tree kernels to work is that we only consider subtrees containing little redundant information (such as, in opinion holder extraction, the subtree encoding the relation between a candidate opinion holder and its nearest predicate). The problem for sentence-level text classification is that, unlike in entity extraction, there are no natural subtrees which immediately spring to mind.

The results of these two experiments may be opposed to the findings in (Kudo & Matsumoto, 2005), but we assume that this is due to the different settings of the experi-

[8]We always tested within the hybrid mode which combines the tree-kernel with the standard bag-of-words features.

ments.[9]

3.6. Error Analysis

We found that the golden standard occasionally contains incorrect labels, i.e. positive sentences have been labeled as negative sentences and vice versa. By closer inspection of some of those cases, we found that the reason for that lies in the automatic projection of labels from the phrase level to the sentence level. As mentioned in Section 3.3, we only carried out a fully automatic projection in case the polarity labels of the phrases within one sentence were identical. However, we spotted several sentences in which phrases were missing in the (manual) annotation of the corpus which thus caused an incorrect projection (as the missing phrases possessed a polarity type opposed to the other actually anntoated expressions).

Another source of error lies in the recognition of polar expressions which forms the basis for any sentence-level feature (Section 3.5.3). Not only is the coverage of current polarity lexicons limited but they also fail to provide the necessary information to disambiguate expressions which only possess a polar meaning with some particular sense (Section 2.4). Our lexicon only disambiguates words on the basis of part-of-speech information (Section 3.4.2) but is unable to disambiguate expressions which contain a unique part-of-speech tag.

3.7. Conclusion

In this chapter, I have shown that the baseline performance of polarity classifiers of news text at sentence level using bag of words can be significantly improved by applying both linguistic features and polarity information. Unlike polarity classification at document level, just using bag of words produces a fairly low performance.

[9] Kudo and Matsumoto (2005) report results on Japanese text, they use twice as much data and consider a closed domain (reviews for Personal Handyphone System) presumably comprising more repetitive language than the multi-topic MPQA news corpus.

Though adding prior polarity information to bag of words already gives a significant boost to the baseline performance at sentence level, adding linguistic features can increase this performance even further significantly. In total, our baseline is improved by up to 10.3%. We also showed that in the absence of any polar information, domain-independent structural features can already improve the performance of bag-of-words feature sets by approximately 6%.

4. Detecting Indefinite Polar Utterances

4.1. Introduction

In Chapter 2, I stated that text classification in sentiment analysis is usually a two-stage classification scenario consisting of subjectivity detection and polarity classification. Both scenarios are mostly considered as a binary classification problem. The classification that was presented in the previous chapter fits into that scheme. It is, however, too simplistic. According to that scheme, once a text is considered subjective, it is either positive or negative. Unfortunately, it fails to account for subjective texts which contain an indefinite polar subjectivity.

Sentences (4.1) and (4.2) are definite polar utterances since these sentences can be categorized as either positive or negative:

(4.1) She's always the best of the best!

(4.2) That product is so bad, it should be illegal.

Sentences (4.3) - (4.5) are examples of indefinite polar utterances:

(4.3) That first record was amazing but then they fell off really fast.

(4.4) She has an average voice.

(4.5) I'm not hellishly impressed.

These utterances have in common that they are subjective and express a value judgment. None of these statements can be categorized as definite positive or negative. The indefiniteness is achieved either by stating both positive and negative aspects (Sentence (4.3)),

by using polar expressions not denoting definite polarity (*average* in Sentence (4.4)), or by diminishing/negating definite polar phrases (Sentence (4.5)).

This chapter presents a small set of features to detect indefinite polar sentences. In order to adhere to the common theme of this thesis, I will present domain independent features reflecting the linguistic structure underlying these types of utterances. Since indefinite utterances or even entire indefinite reviews are part of a realistic review collection, those features might be helpful for an accurate text classification.

We give evidence for the effectiveness of these features by incorporating them into an unsupervised rule-based classifier for sentence-level analysis and compare its performance with supervised machine learning classifiers. We restrict ourselves to sentence-level analysis since we are primarily interested in basic utterances (as we want to explore the nature of this type of opinion) for which sentences are a suitable approximation.

The work presented in this chapter is also described in (Wiegand & Klakow, 2010c).

4.2. Related Work

Koppel and Schler (2006) present a machine learning approach to polarity classification where also reviews with indefinite polarity are considered. A binary classifier for positive and negative polarity is learned using bag-of-words features. Reviews being predicted as positive or negative with a low confidence are classified as indefinite polar reviews. The paper does not address features specifically designed for detecting indefinite polar reviews.

Zhao, Liu, and Wang (2008) consider a CRF-based model for sentence-level polarity classification of reviews also taking into consideration indefinite polar sentences as a separate class. Again, there is no discussion about what predictive features are for this class.

Wilson et al. (2005) present polarity classification of news text on phrase level. Apart from positive and negative polar phrases, phrases with both polarities and neutral polarity are considered. However, our task differs greatly from theirs. Wilson et al. (2005) carry out classification of phrases whereas this work deals with sentence-level classification.

Moreover, this chapter addresses another text type being online reviews whereas Wilson et al. (2005) deal with news texts. As all four polar classes are classified within the same classifier, it is not clear which features are predictive for the indefinite polar classes.

Wilson, Wiebe, and Hwa (2004) present features for distinguishing strong from weak opinion clauses. Weak opinion clauses bear some resemblance to the class of indefinite polar expressions. However, the paper does not address polarity. Moreover, the same differences as the one mentioned to (Wilson et al., 2005) (i.e. level of granularity and text type) also apply to (Wilson et al., 2004).

4.3. Data

We extracted a set of reviews from *Rate-It-All*.[1] Since we want to classify sentences, we restricted our choice to reviews which only comprise one sentence. We only chose those domains which given this restriction still contained sufficient reviews. The domains we include in the experiments of this chapter are *Person (person)*, *Sports & Recreation (sports)*, and *Travel, Food, & Culture (travel)*. For definite polar utterances, we extracted reviews rated with 1 or 5 stars and for indefinite reviews, we extracted reviews rated with 3 stars. Of the latter subset, some reviews were manually removed, since they were deemed definite polar utterances. For the sake of simplicity, we generated a balanced dataset via random sampling. This results in a random baseline of 50% in Accuracy.

We chose web reviews for the experiments in this chapter because it is fairly easy to generate annotated data from a set of reviews (as shown above) in comparison to other domains, such as newswire text, where additional manual annotation would have been required. The annotation of the MPQA-corpus could not be used despite the fact that it is at phrase level (and therefore can be projected to sentence-level, as it has been done in Chapter 3) since *indefinite polarity* as such is not contained in the annotation (see also Section 4.2). Some phrases annotated as private states in MPQA may also be found in our dataset as indefinite polar instances. These phrases were then labeled as either *low* positive or negative polar phrases. Unfortunately, we could not make out a systematic

[1] `http://www.rateitall.com`

Table 4.1.: Size of the different datasets.

Domain	Number of Sentences
person	1914
sports	980
travel	1618

correspondence between the annotation in MPQA and the labels in our dataset.

Table 4.1 lists the size of the resulting datasets.

4.4. Feature Design

Table 4.2 lists all the features that we use. The feature set can be divided into the subset indicating indefinite polarity and the subset indicating definite polarity. We will discuss each of these features individually in the forthcoming subsections. Several of the features require the knowledge of polar expressions (e.g. `PosInPast` or `PolarSuper`). For their detection we use, as in the previous chapter, the Subjectivity Lexicon from the MPQA-project (Wilson et al., 2005). This lexicon is well suited for our experiments since it contains a binary intensity feature dividing entries into *weak* polar expressions (e.g. *valid* or *bulky*) and *strong* polar expressions (e.g. *wonderful* or *hideous*). We make use of this distinction in one of our features (`NegStrongPol`). In order to increase the coverage of the polarity lexicon, we add adjectives from the *Macquarie Semantic Orientation Lexicon* (Mohammad, Dunne, & Dorr, 2009).[2] All these entries are categorized as *weak* polar expressions.

4.4.1. Indefinite Polarity Features

The following subsections describe features indicative of indefinite polar opinions.

[2]We found that other entries are too noisy for our application.

Table 4.2.: Description of the feature set.

Feature	**Abbreviation**	**Indefinite Polarity Feature**	**Definite Polarity Feature**	**Example(s)**
concessive conjunctions	`ConcConj`	✓		but, although, however
concessive conjunctions preceded by a polar expression	`ConcAndPolar`	✓		he is **nice but** ...
detensifiers	`Detens`	✓		rather, kind of, slightly, almost
negated strong polar expressions	`NegStrongPol`	✓		not excellent, not bad
negation expressions	`NegExp`	✓		not, never, nothing
middle-of-the-road polar expressions	`MiddleExp`	✓		solid, average, ordinary
positive polar expressions in past tense clause	`PosInPast`	✓		he **used** to be **funny**
polar superlatives	`PolarSuper`		✓	best, funniest, worst
emphatic cues	`EmphCues`		✓	yeah, ah, grrreeeaaat, !

Concessive Conjunctions (ConcConj)

In the introduction to this chapter, we pointed out that one way of expressing indefinite polarity is to state both a positive and a negative opinion in a sentence. An intuitive heuristic to look for utterances in which both positive and negative polar expressions occur is not very effective. We ascribe it to the fact that the detection of polar opinions is very error prone. The relevant polar expressions may not be detected if they are not included in the polarity lexicon, and even if they can be detected, their contextual polarity may be computed incorrectly. Contextual polarity comprises many linguistic phenomena, such as *negation* or *irony*, which are difficult to model computationally.

We found, however, that there is another feature which most often co-occurs with this type of utterance. Concessive conjunctions, such as *but* or *although*, indicate that two clauses represent semantically opposed propositions. In our dataset this is usually a juxtaposition of two polar opinions. Thus, such a conjunction is also indicative of a sentence with an overall indefinite polarity:

(4.6) A nice^{+} wine, *but* definitely [not worth]$^{-}$ the price.

Concessive Conjunctions Preceded by a Polar Expression (ConcAndPolar)

Even though concessive conjunctions may be detected more easily than two contrasting polar opinions, the concessive conjunction may itself be an ambiguous word. For instance, *but* in the following sentence is not a concessive conjunction:

(4.7) They are nothing *but* an untalented stain on the music world ... totally atrocious music.

We found, however, that a co-occurrence of a polar expression preceding the potential concessive conjunction is a fairly reliable way of disambiguating these words.

Detensifiers (Detens)

Another way of expressing indefinite polarity is to diminish polar phrases. Therefore, a further cue may be diminishing expressions, or so-called *detensifiers*, such as *almost*,

slightly, or *less*:

(4.8) Terry is *almost* as good as Robert Jordan, his stories are *slightly less* word encompassing.

For detensifiers, we adhere to the list presented in (Jason, 1988).

Negated Strong Polar Expressions (`NegStrongPol`)

In traditional polarity classification negated polar expressions are interpreted as if the polarity of the polar expressions were reversed (Kennedy & Inkpen, 2005; Klenner et al., 2009). We argue that for the detection of indefinite polarity negated polar expressions should not be equated with unnegated polar expressions with the opposite polarity. Instead, they should be treated as a separate category. In particular, negated strong polar expressions (Sentence (4.9)) may similarly convey indefinite polarity as detensified polar expressions (Sentence (4.10)):

(4.9) They are *not bad*.

(4.10) They are *quite good*.

We did a simple negation detection matching the lexical entries labeled as negations in (Wilson et al., 2005). We did not carry out a disambiguation of negation words. So the performance of this feature can be considered as a lower bound. As we did not employ full parsing for the experiments in this chapter, we define the scope of a negation as the five words following a negation word.

Negation Expressions (`NegExp`)

`NegStrongPol` is a fairly complex feature in which several properties have to co-occur, i.e. the sentence must contain a polar expression which has to be of strong intensity and it has to be within the scope of a negation. The computation of such a feature is error-prone as the negation may not be correctly computed or the strong polar expression may be overlooked as it is not specified in the polarity lexicon. Therefore, we add a feature

just recognizing negations. Admittedly, this feature is not equivalent to the previous feature but its computation should be much more reliable and, often, it should coincide with `NegStrongPol`.

Middle-of-the-Road Polar Expressions (`MiddleExp`)

Indefinite polarity may not only be conveyed by the use of certain linguistic constructions, be it on discourse level (`ConcConj`) or on syntax level (`Detens` or `NegStrongPol`). It can also be lexically realized by so-called *middle-of-the-road polar expressions*, such as *ok*:

(4.11) This beer brand is *ok* ... really far away of the Paulaner Heffeweissen.

We compiled a list of such expressions by starting with a couple of manually defined seed words which were expanded using semantic resources, such as WordNet (Miller et al., 1990). Moreover, we also manually selected a subset of *weak* polar expressions from the polarity lexicon of the MPQA-project. Note that middle-of-the-road polar expressions differ quite substantially from the polar expressions marked as *both* (e.g. think, believe) or *neutral* (e.g. demand, brag) in that lexicon, though the category names may suggest otherwise. `MiddleExp` always implies a value judgment whereas the two categories in the *Subjectivity Lexicon* usually do not have that property. Besides, these two types of expressions did not show any noticeable predictiveness on our datasets.

Positive Polar Expressions in Past Tense Clause (`PosInPast`)

We observed that in many indefinite polar reviews people tend to recall positive aspects concerning the topic they review which they experienced in the past and contrast them with negative aspects they presently perceive. We found that this behavioural pattern can be automatically identified by detecting a positive polar expression uttered in a past tense clause. Reviews are usually written in present tense and we found that if a clause occurs in past tense, then this will most often be accompanied by a switch in tense:

(4.12) [I $used_{Past}$ to $like^{+}$ those chips a lot $better^{+}$ some years ago], now the only way I eat them is with sour cream.

We also experimented with a related feature, i.e. detecting a negative polar expression in a past tense clause, however, we could not measure any correlation between this pattern and the class of indefinite polar utterances.

4.4.2. Definite Polarity Features

The following subsections describe features indicative of definite polar opinions.

Polar Superlatives (PolarSuper)

Definite polar opinions may often be conveyed by a polar superlative:

(4.13) He's the *best* actor.

Intuitively, the polar intensity of a polar superlative (e.g. *best*) is stronger than the intensity of a polar positive (e.g. *good*) or comparative (e.g. *better*). Though polar superlatives are similar to strong polar expressions, such as *excellent*, or intensified polar expressions, such as *very good*, we found in our initial experiments that they are far less predictive for our task than the polar superlative.

Emphatic Cues (EmphCues)

Often, emphatic cues, such as interjections (*yeah*, *ah* etc.), co-occur with definite polar sentences. A feature detecting such cues may help since in our dataset there are many definite polar sentences in which – apart from the emphatic cue – there is no other feature that could be that easily computed. For instance, in the following sentence the polar opinion is pragmatic, i.e. it is not lexicalized. However, there are three exclamation marks whose occurrence is interpreted as an emphatic cue:

(4.14) I can eat this peanut butter on anything*!!!*

For the implementation of this feature, we mainly relied on exclamation marks and the part-of-speech tag indicating interjections, i.e. *UH*. In addition, we formulated regular expressions capturing irregular spelling as in *suuuper* or *grrreeeaaat*.

4.5. Rule-Based Classifier

The features from Section 4.4 can be used as a rule-based classifier. For each test instance, the occurrences of features indicating definite and indefinite polar utterances are counted. We assign the instance the class with the majority of feature occurrences. In case of ties the instance is classified as definite polar since we have fewer features formulated for that class.

4.6. Experiments

We evaluate the results using Accuracy only (see also Appendix A.1). Table 4.3 displays the individual performance of the different features used as a rule-based classifier (as formulated in Section 4.5). We test for each feature whether it is significantly different from a random baseline (i.e. 50% Accuracy). We report statistical significance on the basis of a χ^2 test.

Each of the features is at least significantly better than the baseline when the entire dataset is considered. It is very striking that among the best performing features are `ConcConj` and `NegExp` which are features describing different types of closed-word classes. Their advantage is that they comprise words frequently occurring across all domains.

The features that fail to be significantly better than the baseline on each domain, i.e. `PolarSuper`, `NegStrongPol`, and `PosInPast`, are more complex than most of the other better performing features. They all describe a co-occurrence of separate properties, e.g. `PosInPast` is a polar expression that also happens to be positive and occurs in a past tense clause. We assume that the reason for these features performing less well lies in the sparsity of their occurrence.

Table 4.4 compares the performance of the unsupervised rule-based classifier using all features with supervised classifiers on 10-fold cross-validation. We compare Support Vector Machines (SVMs) using *SVMLight*[3] and a k Nearest Neighbour Classifier (kNN) using *TiMBL*[4]. For *SVMLight* we use the standard configuration and for *TiMBL* we use the 5 nearest neighbours. This setting produces the best overall performance on all domains. All words contained in the training sets are used as features for the supervised classifiers. Following the insights of Pang et al. (2002), features indicate presence within an instance and not its frequency. The inclusion of our novel high-level features (Table 4.2) did not improve performance of these classifiers when they were added to the bag of words. For the rule-based classifier, we also considered subsets of the features, but no significant improvement over the entire feature set could be achieved. SVMs achieve best performance. Both kNN and the rule-based classifier are significantly worse than SVMs. Surprisingly, the rule-based classifier is as robust as kNN. There is no significant difference between the rule-based classifier and kNN.[5]

Figure 4.1 shows the average performance of the different classifiers with varying amounts of labeled training data. For each configuration, we randomly sampled n training instances from the domain corpus and use the remaining instances as test data. We sampled 20 times and report the averaged result. Even for SVMs, it takes more than 400 labeled data instances to achieve a significantly better Accuracy than the unsupervised rule-based classifier. For less robust supervised classifiers, such as kNN, more than 800 labeled data instances are required to achieve the same performance as the rule-based classifier.

4.7. Error Analysis

Our manual inspection of misclassified data instances revealed that several sentences have been incorrectly labeled in the golden standard. The most frequent mistake is that

[3] `http://svmlight.joachims.org`

[4] `http://ilk.uvt.nl/timbl`

[5] Statistical significance is again reported on the basis of a χ^2 test with significance level $p < 0.001$.

Table 4.3.: Accuracy of the different features on the different domains.

Type	**person**	**sports**	**travel**	**all**
ConcConj	72.99***	71.53***	73.24***	72.76***
ConcAndPolar	65.94***	62.76***	66.25***	65.36***
NegExp	58.99***	60.92***	61.37***	60.26***
EmphCues	59.98***	57.86***	60.88***	59.84***
MiddleExp	59.14***	58.06***	59.77***	59.13***
Detens	55.28**	54.90*	55.56**	55.30***
PolarSuper	52.46	57.65***	53.58*	54.56***
NegStrongPol	52.72	54.08	54.39*	53.73***
PosInPast	53.29*	52.65	50.74	52.23*

Statistical significance is reported on the basis of a χ^2 test with significance levels $p < 0.05$ (*), $p < 0.01$ (**) and $p < 0.001$ (***).

Table 4.4.: Comparison of Accuracy of the different classifiers.

Type	**person**	**sports**	**travel**	**average**
rule-based	76.18	78.06	77.32	77.19
kNN	78.00	77.55	75.59	77.05
SVMs	81.19	81.02	80.22	80.81

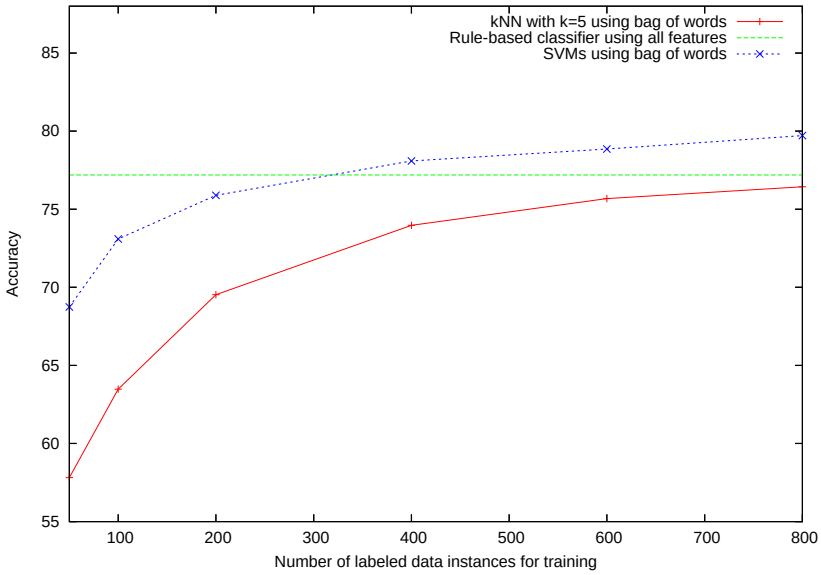


Figure 4.1.: Average Accuracy of the different classifiers using different amounts of labeled training data.

reviews rated with either 1 or 5 reviews, i.e. reviews that we consider as definite polar, are actually indefinite. For some future work on this task, we therefore should manually label sentences in our dataset with regard to polar definiteness from scratch.

We also found that features were frequently not recognized, the reason for that being that words have been misspelt or have been tagged with incorrect part-of-speech tags. By having some cleaner data, Accuracy may increase as the automatic feature extraction would become more reliable. Of course, these two sources of errors (i.e. spelling and part-of-speech tagging) are not the only sources for features being incorrectly extracted. Several of them rely on the recognition of polar expressions but current state-of-the-art polarity lexicons are far from being perfect as they have a limited coverage and cannot sufficiently cope with the ambiguity of polar expressions (see Chapter 3.6).

4.8. Conclusion

In this chapter, we presented a set of discriminative features for the detection of indefinite polar sentences. All features are based on linguistic observations or intuitions. We showed that these features can be used as an unsupervised rule-based classifier which provides as good as performance as supervised machine learning classifiers, such as kNN trained on bag-of-words. When only small amounts of training data are available (i.e. less than 300 sentences), the unsupervised approach even outperforms more robust supervised classifiers, such as SVMs. Since the feature set uses domain-independent features the classifier works equally well throughout different domains.

We leave it to future work to examine the impact of these features in a polarity classifier also accounting for the other common polarity types, i.e. positive and negative. Unfortunately, due to the lack of annotated data for this scenario, this study is beyond the scope of this thesis.

5. Topic-Related Sentence-Level Polarity Classification

5.1. Introduction

In this chapter, I return again to sentence-level polarity classification. While in Chapter 3 the task was to predict the overall polarity of a sentence, in this chapter we are interested in the polarity towards a specific topic, i.e. targets of opinions. The inclusion of targets of opinions may result in a more complex type of classification, however, this task is also more similar to realistic scenarios. People are usually interested in opinions towards certain topics rather than the overall polarity of a sentence. Moreover, even though the task may be more complex than plain polarity classification, the presence of a target mentioning in a sentence may help to overcome the common ambiguity problem that a sentence contains polar expressions with opposing polarity types as will be explained below.

The scenario that is going to be used in this chapter looks as follows: the problem of polarity classification is converted into a retrieval task. A query consisting of a topic and a target polarity, such as *{topic:* ***Mozart****, target polarity:* ***positive****}*, is posed to a topic-related polarity ranker. The ranker should be able to highly rank Sentence (5.1), which contains an opinion about the target whose polarity matches the target polarity, and disprefer Sentence (5.2), which contains an opinion about the target topic but whose polarity is incorrect, and Sentence (5.3), which is merely a factual statement about the target topic.

(5.1) **positive statement:** My argument is that it is *pointless*$^-$ to ordinary mortals like

you and me to discuss why Mozart was a ***genius***$^{+}$.

(5.2) **negative statement:** I have to say that I [don't ***like***$^{+}$]$^{-}$ Mozart.

(5.3) **neutral statement:** Wolfgang Amadeus Mozart's 250th birthday is coming up on the 27th of this month.

In order to highly rank Sentence (5.1), the ranker must be able to decide which of the two polar expressions having opposing polarity types, i.e. *pointless* or *genius*, is related towards the topic. Bag-of-words classifiers, which we will use as a baseline, might therefore mislabel this sentence. A classification which jointly takes the topic term and the polar expressions into account, on the other hand, may result in a correct classification. For example, the closest polar expression, i.e. *genius*, is the expression which actually relates to the topic. This ambiguity can be resolved by both spatial distance and syntactic information. In the current example, there is a direct syntactic relationship, i.e. a *subject-of* relationship, between the topic term and the polar expression relating to it. Usually, syntactic relation features are more precise but also much sparser than proximity features.

Not only is it important to identify the polar expression within a sentence which actually relates to the polar expression but also to interpret a polar expression correctly in its context. In Sentence (5.2), the only polar expression has a positive prior polarity but since it is negated its contextual polarity is negative.

All these observations suggest that there are several sources of information to be considered which is why we examine features incorporating polarity information extracted from a large polarity lexicon, syntactic information from a dependency parse, and surface-based proximity. In particular, we address the issue whether syntactic information is beneficial in this task. Many features that will be tested in this chapter resemble those from previous experiments on plain sentence-level polarity classification in Chapter 3. We also want to examine which of these features maintain their effectiveness on this task.

Modeling topic-related polarity classification as a retrieval task (instead of a traditional classification task) simplifies the task since the ranking does not require that all instances are classified correctly, i.e. lower ranks are virtually neglected by evaluation metrics for

ranking, so incorrect predictions on lower ranks do not mar the overall result. Secondly, neutral statements or opinions with indefinite polarity (as they have been dealt with in Chapter 4) do not have to be specifically modeled, as the target polarity is either positive or negative. Instances that do not match the target polarity should not occur on the higher ranks but a reason, i.e. an explanation why these instances are different (for instance by labeling them as neutral or indefinite polar) is not required.

The work presented in this chapter is also described in (Wiegand & Klakow, 2009c).

5.2. Related Work

The main focus of existing work in sentiment analysis has been on plain polarity classification which is carried out either at document level (Pang et al., 2002), sentence level (Chapter 3), or expression level (Wilson et al., 2005). There has also been quite some work on extracting and summarizing opinions regarding specific features of a particular product, one of the earliest works being (Hu & Liu, 2004). Unlike the work presented in this chapter, the task is usually confined to a very small domain. Moreover, the plethora of positively labeled data instances allows the effective usage of syntactic relation patterns.

Santos et al. (2009) show that a Divergence From Randomness proximity model improves the retrieval of subjective documents. However, neither an evaluation on sentence level and nor an evaluation of polarity classification is conducted.

The works most closely related to the work presented in this chapter are (Kessler & Nicolov, 2009) and (Jakob & Gurevych, 2010a) who examine the detection of targets of opinions by using syntactic information. Whereas they both discuss how to detect whether two entities are in an opinion-target relationship – Kessler and Nicolov (2009) even already know that there is such a relationship in the sentence to be processed – we do not conduct an explicit entity extraction but classify whether or not a sentence contains an opinion-target relationship. Another difference is that we consider this task as a ranking task while Kessler and Nicolov (2009) and Jakob and Gurevych (2010a) consider this as a classification task (Kessler and Nicolov (2009) employ Support Vector

Machines (SVMs) while Jakob and Gurevych (2010a) use Conditional Random Fields (CRFs)). Like Jakob and Gurevych (2010a), we also carry out a cross-domain evaluation, as our queries deal with various different domains. Unlike (Kessler & Nicolov, 2009; Jakob & Gurevych, 2010a), we also restrict the opinion-bearing word to be of a specific polarity. Thus, we can use knowledge about polar expressions in order to predict an opinion-target relationship in a sentence.

The change in focus, i.e. the fact that we deal with a sentence-level ranking task rather than an entity extraction task, raises the question whether a similar amount of syntactic knowledge is necessary or whether sufficient information can be drawn from more surface-based features and lexical knowledge of prior polarity. Moreover, we believe that our results are more significant for realistic scenarios like *opinion question answering*, since our settings are more similar to such a task than the ones presented by Kessler and Nicolov (2009); Jakob and Gurevych (2010a).

5.3. Data

The dataset we use in the experiments of this chapter is a set of labeled sentences retrieved from relevant documents of the TREC Blog06 corpus (Macdonald & Ounis, 2006) for TREC Blog 2007 topics (Macdonald, Ounis, & Soboroff, 2008). The test collection contains 50 topics. For each topic we formulate two separate queries, one asking for positive opinions and another asking for negative opinions. In the final collection we only include queries for which there is at least one correct answer sentence. Thus, we arrive at 86 queries of which 45 ask for positive and 41 ask for negative opinions. The sentences have been retrieved by using a language model-based retrieval (Shen, Leidner, Merkel, & Klakow, 2007). Each sentence from the retrieval output has been manually labeled. One annotator judged whether a sentence expresses an opinion with the target polarity towards a specific topic or not. Difficult cases have been labeled after discussion with another annotator. The additional annotator only annotated those difficult cases. The annotation is strictly done at sentence level, i.e. no information of surrounding context is taken into consideration. This means that each positively labeled sentence

must contain some (human recognizable) form of a polar expression and a topic-related word. Our decision to restrict our experiments to the sentence level is primarily to reduce the level of complexity. We are aware of the fact that we ignore inter-sentential relationships, however, Kessler and Nicolov (2009) state that on their similar dataset 91% of the opinion-target relations are within the same sentence.

The proportion of relevant sentences containing at least one topic term in *our* corpus is 97% which is fairly high. By a topic term, we mean an occurrence of a token being part of the topic. Although 71% of the relevant sentences contain a polar expression of the target polarity according to the polarity lexicon we use, in 50% of the sentences there is also at least one polar expression with opposing polarity. The joint occurrence of a polar expression matching with the target polarity and a topic term is no reliable indicator of a sentence being relevant, either. Only approximately 17% of these cases are correct. The entire dataset contains 25,651 sentences of which only 1,419 (i.e. 5.5%) are relevant[1] indicating a fairly high class imbalance. This statistical analysis suggests that the extraction of correct sentences is fairly difficult.

5.4. Feature Design

In the following, we will describe the different features we use for the task of topic-related polarity classification. Some of the features bear some resemblance to the features used in plain sentence-level polarity classification presented in Chapter 3. The fact that similar features are re-used for this task should be regarded as evidence for the robustness and general applicability of these feature types for sentiment analysis.

5.4.1. Sentence Retrieval, Topic Feature, and Text Classifiers

Our simplest baseline consists of a cascade of a sentence-retrieval engine and two text classifiers, one to distinguish between objective and subjective content, and another to

[1] By relevant, we mean every sentence which expresses a polar opinion (matching the target polarity) towards the topic term, i.e. neither a polar expression nor a topic term need to be present.

distinguish between positive and negative polarity. We employ stemming and only consider unigrams as features. The two text classifiers are run one after another on the ranked output. Rather than combining the scores of the classifiers with the retrieval score in order to re-rank the sentences, we maintain the ranking of the sentence retrieval and delete all sentences being objective and not matching the target polarity. This method produces better results than combining the scores by some form of interpolation and does not require any parameter estimation. This hierarchical two-stage classification (subjectivity detection followed by polarity classification) has already been motivated in Chapter 2.3.1.

We also consider a separate topic feature which counts the number of topic terms within a sentence since this feature scales up better with the other types of features we use for a learning-based ranker than the sentence retrieval score.

5.4.2. Polarity Features

For our polarity features, we mainly rely, as in the previous chapters of this thesis, on the largest publicly available polarity lexicon, the Subjectivity Lexicon (Wilson et al., 2005) from the MPQA-project. We chose this lexicon since, unlike other resources, it does not only have part-of-speech labels attached to polar expressions, thus allowing a crude form of disambiguation[2], but also distinguishes between *weak* and *strong* expressions.

The set of polarity features that we use in this chapter is very similar to the sentence-level *prior polarity* and *linguistic* features used for plain polarity classification presented in Chapter 3.4.

As a basic polarity feature (`PolMatch`), we count the number of polar expressions within a candidate sentence which match the target polarity. Since this basic polarity feature is fairly coarse, we add further polarity features which have specific linguistic properties. We include a feature for strong polar expressions (`StrongPolMatch`) and a feature for polar expressions being modified by an intensifier (`IntensPolMatch`), such as *very*. We suspect that a strong polar expression, such as *excellent*, or an intensified

[2]Thus we can distinguish between the preposition *like* and the polar verb *like*.

Table 5.1.: List of polarity features.

Feature	**Abbreviation**
number of polar expressions within sentence with matching polarity *(basic polarity feature)*	`PolMatch`
number of strong polar expressions within sentence with matching polarity	`StrongPolMatch`
number of intensified polar expressions within sentence with matching polarity	`IntensPolMatch`
number of strong and intensified polar expressions within sentence with matching polarity	`StrongIntensPolMatch`
number of polar nouns/verbs/adjectives within sentence with matching polarity	`Pol`*`POS`*`Match`
number of strong polar nouns/verbs/adjectives within sentence with matching polarity	`StrongPol`*`POS`*`Match`
number of intensified polar nouns/verbs/adjectives within sentence with matching polarity	`IntensPol`*`POS`*`Match`
number of strong and intensified polar nouns/verbs/adjectives within sentence with matching polarity	`StrongIntensPol`*`POS`*`Match`

polar expression, such as *very nice*$^+$, might be more indicative of a specific polarity than the occurrence of any plain polar expression. We use the list of intensifiers from Wilson et al. (2005). Furthermore, we distinguish polar expressions with regard to the most frequent part-of-speech types (`Pol`*`POS`*`Match`), these being *nouns*, *verbs*, and *adjectives*.[3] Some parts of speech, for instance adjectives, are more likely to carry polar information than others (Pang et al., 2002). Table 5.1 lists all polarity features we use. It also includes some combined features of the features mentioned above, i.e. `StrongPol`*`POS`*`Match`, `IntensPol`*`POS`*`Match`, and `StrongIntensPol`*`POS`*`Match`.

We also experimented with features counting the number of polar expressions *not matching* the target polarity but none of these features gave any improvement when they were added to the features counting the number of matches.

[3]We subsume *adverbs* by adjectives as well.

5.4.3. Negation Modeling

A correct contextual disambiguation of polar expressions is important for topic-related sentence-level polarity classification since the instances to be classified are rather sparse in terms of polarity information. Therefore, we conduct negation modeling. Our negation module comprises three steps. In the first step, all potential negation expressions of a sentence are marked. In addition to common negation expressions, such as *not*, we also consider *polarity shifters*. Polarity shifters are weaker than ordinary negation expressions in the sense that they often only reverse a particular polarity type.[4] In the second step, all the potential negation expressions are disambiguated. All those cues which are not within a negation context, e.g. *not* in *not just*, are discarded. In the final step, the polarity of all polar expressions occurring within a window of five words[5] after a negation expression is reversed. We use the list of negation expressions, negation contexts, and polarity shifters from Wilson et al. (2005).

5.4.4. Spatial Distance

Textual proximity provides additional information to the previously mentioned features, as it takes the relation between polar expression and topic term into account. In Sentence (5.4), for example, the positive polar expression *genius* is closest to the topic term *Mozart*, which is an indication that the sentence describes a positive opinion towards the topic.

(5.4) My argument is that it is *pointless*$^-$ to ordinary mortals like you and me to discuss why Mozart was a ***genius***$^+$.

We encoded our distance feature as a binary feature with a threshold value.[6] This gave much better performance than encoding the explicit values in spite of attempts to scale this feature with the remaining ones. Since we do not have any development data, we had to determine the appropriate threshold values on our test data. The threshold

[4] For example, the shifter *abate* only modifies negative polar expressions as in *abate the damage*.
[5] This threshold value is taken from Wilson et al. (2005) which has been determined experimentally.
[6] The feature is active if a polar expression and topic term are sufficiently close.

value is set to 8.[7] Since all feature sets containing this distance feature supported the same threshold value, we have strong reasons to believe that the value chosen is fairly universal. We also experimented with a more straightforward distance feature which checks whether the closest polar expression to the topic term matches the target polarity. However, we did not measure any noticeable performance gain by this feature.

5.4.5. Syntactic Features from a Dependency Path

In addition to polarity and distance features we use a small set of syntactic features. By that we mean all those features that require the presence of a syntactic dependency parse. This set of features supplements both of the other feature types.

Syntactic Prominence Features

Similar to the polarity features are the two *prominence features* we use. Their purpose is to indicate the overall polarity of a sentence. Very similar features have again also been presented in Chapter 3.4 where they have been shown to be effective for sentence-level polarity classification on the news domain. Each polar expression can be characterized by its depth within the syntactic parse tree. Depth is defined as the number of edges from the node representing the polar expression to the root node. Usually, the deeper a node of a polar expression is, the less prominent it is within the sentence. Similar to the distance feature, we define a binary feature (`LowDepth`) which is active if a polar expression has a sufficiently low depth. The threshold value is set to 5.[8] The main predicate (`MainPred`), too, is usually very indicative of the overall polarity of a sentence. Sentence (5.5) is a case where the main predicate coincides with the correct overall polarity.

(5.5) The strings $[\textit{screwed up}]^{-}_{mainPred}$ the concert, in particular, my ***favorite***$^{+}$ scores by Mozart. *(overall polarity: negative, polarity towards Mozart: positive)*

[7] The threshold may appear quite high. However, given the fact that the average sentence length in this collection is at approximately 30 tokens and that there is a tendency of topic terms to be sentence initial or final, this value is fairly plausible.

[8] The large value for the depth feature can be explained by the fact that Minipar uses auxiliary nodes in addition to the nodes representing the actual words.

Table 5.2.: List of syntactic features.

Syntactic Prominence Features	
Feature	**Abbreviation**
number of matching polar expressions with low depth within the syntactic parse tree	`LowDepth`
is the main predicate of the sentence a matching polar expression?	`MainPred`

Syntactic Relation Features	
Feature	**Abbreviation**
number of paths with an immediate dominance relationship between topic term and matching polar expression	`ImmediateDom`
number of paths with a dominating relationship between topic term and matching polar expression	`Dom`
number of paths where topic term dominates matching polar expression	`TopicDomPol`
number of paths where topic term is dominated by matching polar expression	`PolDomTopic`
number of paths between matching polar expression and topic term which are contained within the same event structure	`SameEvent`
number of paths between matching polar expression and topic term which do not cross the root node	`NoCrossRoot`

Syntactic Relation Features

The shortcoming of the prominence features is that they do not consider the relation of a polar expression to a mentioning of a topic but just focus on the overall polarity of a sentence. The overall polarity, however, does not need to coincide with the polarity towards a topic term, as it is shown by Sentence (5.5).

Moreover, textual proximity is sometimes a misleading clue as illustrated by Sentence (5.6) where the polar expression with the shortest distance to the topic term is not the polar expression which relates to it.

(5.6) Mozart, it is $save^{+}$ to say, ***failed***$^{-}$ to bring music one step forward.

That is why we use a set of features describing the dependency relation path between

polar expression and topic term. Unlike previous work (Kessler & Nicolov, 2009), we do not focus on the relation labels on the path due to the heavy data-sparseness we experienced in initial experiments. Instead, we define features on the configuration of the path. The advantage of this is that these features are more general.

We use one feature that counts the number of paths with a direct dominance relationship (`ImmediateDom`), i.e. the paths between polar expressions and topic terms which are directly connected by one edge. All common relationships, such as *subject-verb*, *verb-object*, or *modifier-noun* are subsumed by this feature. We also assume that, in general, any dominance relationship (`Dom`) is more indicative than other paths.[9] Furthermore, we use separate features depending on whether topic term dominates the polar expression (`TopicDomPol`) or it is dominated by such an expression (`PolDomTopic`).

Often a sentence contains more than one clause. A polar expression is less likely to refer to a topic term in case they appear in different statements. We account for this by two additional features. The first counts the number of paths within a sentence between polar expressions and topic terms which are within the same event structure (`SameEvent`). For this feature, we exclusively rely on the event-boundary annotation of a sentence by the dependency parser we use, i.e. Minipar (Lin, 1998). Two nodes are within the same event structure, if the they have the same closest event-boundary node dominating them.[10] Additionally, we define a feature which counts the number of paths which do not cross the root node (`NoCrossRoot`). The root node typically connects different clauses of a sentence.

Table 5.2 summarizes all the different syntactic features we use.

In order to familiarize the reader with the features, Figure 5.1 illustrates a sentence with two candidate paths and the feature updates associated with both paths.

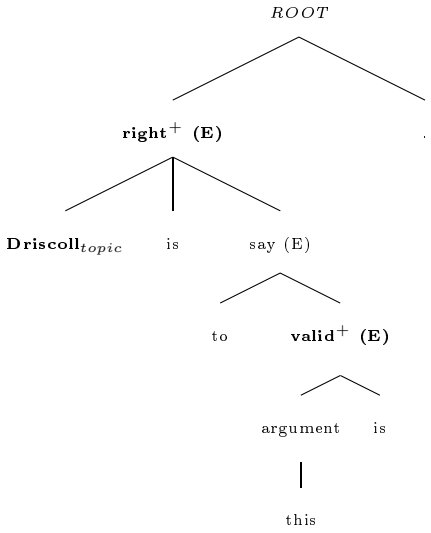
[9] We mean paths which go both up and down a tree.

[10] We assume the dominance relationship to be reflexive.

Sentence: ***Driscoll is right$^+$ to say this argument is valid$^+$.***

Target polarity: ***positive***

Dependency Parse Tree	**Feature Updates for {Driscoll,right}**

	`ImmediateDom++;`
	`Dom++;`
	`PolDomTopic++;`
	`SameEvent++;`
	`NoCrossRoot++;`
	`MainPred:=True;`
	`LowDepth++;`

	Feature Updates for {Driscoll,valid}
	`NoCrossRoot++;` `LowDepth++;`

Figure 5.1.: Illustration of a (simplified) dependency parse tree and corresponding updates for syntactic features. Nodes which present an event boundary are marked with *(E)*. Note that the pair *{Driscoll,right}* expresses a genuine opinion-target relationship. Consequently, much more features fire.

5.5. Experiments

We report statistical significance on the basis of a paired t-test using 0.05 as the significance level on a 10-fold cross-validation. For sentence retrieval, we used the language model-based retrieval engine from Shen et al. (2007). The text classifiers were trained using SVMLight (Joachims, 1999a) in its standard configuration. The subjectivity classifier was trained on the dataset presented by Pang and Lee (2004) which contains movie reviews from www.rottentomatoes.com to represent subjective texts and plot summaries from the Internet Movie Database (www.imdb.com) to represent objective texts. The polarity classifier was trained on a labeled set of sentences we downloaded from *Rate-It-All*[11]. Both datasets are balanced. The former dataset comprises 5,000 sentences and the latter of approximately 6,800 sentences per class. Unlike the standard dataset for polarity classification (Pang et al., 2002), our dataset is not at document level but at sentence level[12] and also comprises reviews from several domains and not exclusively the movie domain. Thus, we believe that this dataset is more suitable for our task since we use it for multi-domain sentence-level classification. We use the entire vocabulary of the data collection as our feature set. Feature selection did not result in a significant improvement on our test data.

For ranking, we use *Yasmet*[13], a Maximum Entropy ranker. Maximum Entropy models are known to be most suitable for ranking tasks (Ravichandran, Hovy, & Och, 2003). We trained the ranker with 1,000 iterations. This gave the best performance on all feature sets. For part-of-speech tagging we employ the *C&C tagger*[14] and for dependency parsing Minipar (Lin, 1998). We evaluate performance by measuring *Mean Reciprocal Rank (MRR)*, *Precision at Rank 10 (Prec@10)*, and *Mean Average Precision (MAP)*. These are common metrics for measuring ranking performance. MRR exclusively focuses on the highest ranked correct instance in a ranking (no matter where it is situated in the ranking). Prec@10 is restricted to the 10 most highly ranked instances. Thus, this

[11] `http://www.rateitall.com`
[12] We only extracted reviews comprising one sentence.
[13] `http://www.fjoch.com/YASMET.html`
[14] `http://svn.ask.it.usyd.edu.au/trac/candc`

Table 5.3.: Performance of factoid sentence retrieval in combination with text classifiers.

Features	**MAP**	**MRR**	**Prec@10**
sentence retrieval	0.140	0.206	0.088
sentence retrieval + subjectivity classifier	0.179	0.247	0.118
sentence retrieval + subjectivity classifier + polarity classifier	0.220	0.267	0.114

metric reflects the (default) presentation of search results of common search engines, such as *Google*. MAP is the most sophisticated metric as it takes into account *all* relevant instances in the entire ranking. A formal definition of these measures is presented in Appendix A.2.

Due to the high coverage of topic terms within the set of positive labeled sentences (97%), we discard all instances not containing at least one topic term. This means that the topic feature counting the number of topic terms (see Section 5.4.1) is no longer an obligatory feature. In fact, we even found in our initial experiments that this gave much better performance than taking all data instances into account and always adding the topic feature.

5.5.1. Impact of Sentence Retrieval Combined with Text Classification

Table 5.3 displays the results of the baselines using sentence retrieval with a subjectivity and a polarity filter. The results show that both text classifiers systematically increase performance of retrieval. Only the increase in Prec@10 is marginal and slightly decreases when polarity classification is added to subjectivity classification.

5.5.2. Comparing Basic Polarity Feature and Text Classifiers

Table 5.4 compares the baseline using sentence retrieval and text classifiers with the basic polarity feature (i.e. `PolMatch`) using polarity information from the polarity lexicon. The polarity feature outperforms the baseline on all evaluation measures, most notably on MRR and Prec@10. We assume that the text classifiers suffer from a domain mis-

match. The polarity lexicon is more likely to encode domain-independent knowledge. Unfortunately, combining the components from the baseline with the polarity feature is unsuccessful. Only the addition of the topic feature (which encodes information similar to the sentence retrieval) to the polarity feature results in a slight (but not significant) increase in MAP. Apparently, the precise amount of word overlap between topic and candidate sentence is less important than in factoid retrieval. Neither do the text classifiers contain any more additional useful information than the polarity feature.

This result also proves our assumption made in Section 5.1 that for this ranking task one does not necessarily have to explicitly model classes other than the target class (i.e. a specific polarity type). Recall from that section that in ordinary classification, one would need to consider a subjectivity classifier to distinguish between factual and subjective statements. The text classifiers which include a subjectivity classifier do not improve the ranking when added to the polarity feature.[15]

Unfortunately, we could not increase the performance of the text classifiers by adding to the bag-of-words features of the text classifiers more expressive linguistic features not relating to polar expressions. While in Chapter 3, an improvement could be achieved by using *linguistic word-level features* (i.e. features combining lexical information with some syntactic properties that those words possess in their particular contexts), on the blog data we did not measure a similar effect. We assume that, like the bag-of-words features, the linguistic word-level features suffer from a domain mismatch. While in Chapter 3 the text is only news-domain (mostly politics), the topics to be found on the blog dataset we are using in this chapter are much more diverse.

5.5.3. Comparing Polarity Features and Syntactic Features

Table 5.5 displays the performance of various feature combinations of polarity and syntactic features. Each feature set is evaluated both without negation modeling (*plain*) and with negation modeling (*negation*). When syntactic features are added to the basic

[15]The same also holds for domain-independent subjectivity features using the polarity lexicon, e.g. the number of subjective expressions in a sentence, with which we also experimented.

Table 5.4.: Performance text classifiers and basic polarity feature.

Features	**MAP**	**MRR**	**Prec@10**
sentence retrieval with text classifiers	0.220	0.267	0.114
basic polarity feature	0.236	**0.420**	**0.212**
basic polarity feature + topic	**0.239**	0.394	0.200
basic polarity feature + text classifiers	0.227	0.380	0.188
basic polarity feature + topic + text classifiers	0.222	0.390	0.179

polarity feature, there is always an increase in performance. With regard to MAP the improvement is always significant. With regard to Prec@10, only the presence of the relation features results in a significant increase. With regard to MRR, for a systematic improvement all polarity features have to be present as well in addition to these features. When the syntactic features are added to all polarity features the increase in performance is similar. The best performing feature set (on average) is the set using all polarity scores and the syntactic relation features. It significantly outperforms the basic polarity feature on all evaluation measures. We, therefore, assume that the syntactic relation features are much more important than the syntactic prominence features.

With the exception of some few feature sets, adding negation modeling increases performance as well. However, the improvement is not systematically significant for any evaluation measure (though for MAP there is only one feature set in which the improvement is not statistically significant).

To a great extent these results are consistent with our results on plain sentence-level polarity classification from Chapter 3. In this chapter, syntactic prominence features always yield an improvement in performance when added to the other polarity features. In Chapter 3, linguistic sentence-level features, which amount to the same type of features as the syntactic prominence features, improved performance when added to prior-polarity features. One additional insight of this chapter is that the syntactic relation features are *more* effective than the syntactic prominence features. Moreover, the impact of negation is different in these two scenarios. While it slightly helps in this chapter it did not

Table 5.5.: Performance of polarity features and syntactic features. Each feature set is evaluated without negation modeling (*plain*) and with negation modeling (*negation*).

Features	**MAP**		**MRR**		**Prec@10**	
	plain	**negation**	**plain**	**negation**	**plain**	**negation**
basic polarity feature	0.236	0.245†	0.420	0.441	0.212	0.215
basic pol. feat. + syntactic prominence feat.	0.258*	0.266*†	0.477*	0.473	0.214	0.216
basic pol. feat. + syntactic relation feat.	0.256*	0.269*†	0.444	0.481†	0.237*	0.249*
basic pol. feat. + all syntactic feat.	0.262*	0.278*†	0.475	0.509*	0.237*	0.244*
all polarity features	0.245	0.257†	0.466	0.489†	0.207	0.215
all pol. feat. + syntactic prominence feat.	0.261*	0.269*	0.477	0.474	0.210	0.222†
all pol. feat. + syntactic relation feat.	**0.273***	0.281*†	**0.509***	0.518*	**0.240***	**0.249***
all pol. feat. + all syntactic feat.	0.272*	**0.284***†	0.502*	**0.526***	0.231*	0.242*†

*: significantly better than basic polarity feature (with/without negation modeling) on the basis of a paired t-test using $p < 0.05$

†: significantly better than the corresponding feature set *without negation modeling* on the basis of a paired t-test using $p < 0.05$

show any improvement in Chapter 3. We strongly assume that this is a side-effect of different feature encoding. While in Chapter 3 the number of negated polar expressions (with a particular polarity type) was taken into consideration with a separate feature, in this chapter it is incorporated into the basic polarity feature.[16] We will see in the next Chapter that the incorporation of negation in the basic polarity feature will also work for rule-based polarity classification on document level.

5.5.4. Impact of Distance Feature

Table 5.6 displays in detail what impact the addition of the distance feature has on the previously presented feature sets. On almost every feature set, there is an increase in performance when this feature is added. However, the degree of improvement varies. It is smallest on those feature sets which include the syntactic relation features. We, therefore, believe that these two feature types encode very much the same thing. Many of the syntactic relation features implicitly demand the topic word and polar expression to be close to each other. Therefore, when a syntactic relation feature fires, so does the distance feature. Unfortunately, our attempts to combine the syntactic relation features with the distance feature in a more effective way by applying feature selection remained unsuccessful. Table 5.6 even suggests that syntactic features are not actually required for this classification task since the best performing feature set only comprises all polarity features and the distance feature. The improvement gained by this feature set when compared to the basic polarity feature is larger than the sum of improvements gained when the two feature subsets are evaluated separately.[17] We assume that in the feature spaces representing the two separate feature sets the decision boundary is highly non-

[16] If we want to count the number of positive polar expressions in a sentence, then we consider negated negative polar expressions as positive polar expressions; in Chapter 3 the number of negated negative polar expressions was regarded as an individual feature and the occurrences of those negated polar expressions did not have any impact on the feature counting the number of positive polar expressions.

[17] The improvement from the basic polarity feature to the optimal feature set is greater than the sum of improvements of the feature set comprising the basic polarity feature and the distance feature and the feature set comprising all polarity features.

Table 5.6.: Impact of distance feature.

Features	**MAP**		**MRR**		**Prec@10**	
		+dist		**+dist**		**+dist**
sentence retrieval with text classifiers	0.220	–	0.267	–	0.114	–
basic polarity feature	0.245	$0.266^{\dagger}$	0.441	$0.491^{\dagger}$	0.215	0.226
basic pol. feat. + syntactic prominence feat.	0.266^{*}	0.276	0.473	0.499	0.216	$0.235^{\dagger}$
basic pol. feat. + syntactic relation feat.	0.269^{*}	0.270	0.481	0.498	0.249^{*}	0.253^{*}
basic pol. feat. + all syntactic feat.	0.278^{*}	0.271	0.509^{*}	0.521	0.244^{*}	0.256^{*}
all polarity features	0.257	$\mathbf{0.302}^{*\dagger}$	0.489	$\mathbf{0.596}^{*\dagger}$	0.215	$\mathbf{0.257}^{*\dagger}$
all pol. feat. + syntactic prominence feat.	0.269^{*}	$0.285^{*\dagger}$	0.474	$0.532^{\dagger}$	0.222	$0.256^{*\dagger}$
all pol. feat. + syntactic relation feat.	0.281^{*}	0.285^{*}	0.518^{*}	$0.569^{*\dagger}$	$\mathbf{0.249}^{*}$	0.256^{*}
all pol. feat. + all syntactic feat.	$\mathbf{0.284}^{*}$	0.281	$\mathbf{0.526}^{*}$	0.555^{*}	0.242^{*}	0.252^{*}

All feature sets – with the exception of *sentence retrieval with text classifiers* – include **negation** modeling.
+dist: distance feature
*: significantly better than basic polarity feature (with/without distance feature) on the basis of a paired t-test using $p < 0.05$
†: significantly better than the corresponding feature set *without distance feature* on the basis of a paired t-test using $p < 0.05$

linear. The *combination* of the two sets provides the feature space with the best possible class separation, even though there are other feature subsets, such as the basic polarity feature and the syntactic features, which are *individually* more discriminative than the feature set comprising all polar expressions or the feature set comprising the basic polarity feature and the distance feature.

Accounting for different types of polar expressions is important and, apparently, this is appropriately reflected by our set of different polarity features. Furthermore, polar expressions within the vicinity of a topic term seem to be crucial for a correct classification, as well. Obviously, defining vicinity by a fixed window size is more robust than relying on syntactic constraints.

Despite its lack of syntactic knowledge, the optimal feature set shows a considerable increase in performance when compared with the baseline ranker relying on text classification with an absolute improvement of 8.2% in MAP, 32.9% in MRR, and 14.3% in Prec@10. There is still an improvement by 6.6% in MAP, 17.6% in MRR, and 4.5% in Prec@10 when the optimal feature set is compared against the simplest ranker comprising one polarity feature (without negation modeling).

5.6. Error Analysis

The result that syntactic relation features are less robust on this task is contrary to our expectations. The poor text quality (i.e. various spelling mistakes, incomplete sentences etc.) may have a notable negative impact on the parsing quality. Moreover, we observed that often *aspects* of topics (Somasundaran & Wiebe, 2009) instead of the topic itself are directly syntactically related to a polar expression. For example, given the query *{topic:* ***Mozart****, target polarity:* ***positive****}*, the relevant Sentence (5.7) contains the polar expression with matching polarity, i.e. *nice*, and the aspect of the topic, i.e. *tunes*, (and not the topic) in a modifier relationship.

(5.7) Mozart wrote $\boldsymbol{nice}^{+}$ tunes_{aspect}.

Unfortunately, the task of extracting (potential) aspects of topics in an unrestricted domain is extremely difficult which is why we ignored it for this task.

Another issue that might have degraded the performance of the syntactic relation features could be the fact that we did not carry out any pronoun resolution since the noisy blog data heavily degrade the quality of resolution. As a result of that given the query *{topic:* ***Driscoll****, target polarity:* ***negative****}*, the polar expression with matching polarity in Sentence (5.8), i.e. *embarrassed*, cannot be related to the topic *Driscoll*, since the two words are in two different clauses. However, the referring expression *he* is the subject of the polar expression.

(5.8) I'm a very $tolerant^{+}$ person but if that is what Driscoll_i said, he_i should be $\boldsymbol{embarrassed}^{-}$ of himself.

Pronoun resolution has been shown to improve performance on related tasks, such as topic-related entity extraction of opinions (Jakob & Gurevych, 2010b). However, the effectiveness on our data may be limited as (based on our comparison with several publicly available corpora used for sentiment analysis) our blog data will be much noisier than the dataset on which the pronoun resolution has been applied (Zhuang, Jing, & Zhu, 2006).

5.7. Conclusion

In this chapter, we have evaluated different methods for topic-related polarity classification at sentence level. We have shown that a polarity classifier based on simple bag-of-words text classification produces fairly poor results. Better performance can be achieved by classifiers using features derived from a polarity lexicon. Obviously, the polarity information encoded in polarity lexicons is more domain independent. Optimal performance of this type of classifier can be achieved when a small set of lightweight linguistic polarity features is used in combination with a distance feature. A distance feature thus helps to disambiguate polarity information in a sentence. Therefore, to some extent a joint modeling of polarity information and topic information is beneficial. Syntactic features derived from a dependency parse are not necessary for this classification task when a distance feature is considered.

6. Bootstrapping Algorithms for Polarity Classification

6.1. Introduction

Supervised polarity classification, in particular classifiers using bag of words, are heavily domain-dependent, i.e. they usually generalize fairly badly across different domains. (One such example has been described in the previous chapter, i.e. in Chapter 5.5.2). Yet the costs to label data for any possible domain are prohibitively expensive.

In this chapter, I will present experiments and results for bootstrapping algorithms for polarity classification on document level. I will focus on two types of methods:

- semi-supervised learning
- supervised classifiers bootstrapped with the help of rule-based classifiers

In both methods a (large) unlabeled corpus is annotated with some prior knowledge about the task. While in the first method this is achieved by using small amounts of labeled data, it is a rule-based classifier in the second method. The extended annotation i.e. the annotation on the previously unlabeled corpus should ideally present a labeled training set that allows more robust classifiers to be built than the classifiers exclusively using the prior knowledge source.

The purpose of this chapter is to show under what settings these bootstrapping methods work for polarity classification on document level and also compare the two types with each other. As in the previous chapters, I will in particular focus on the impact of linguistic knowledge on this classification task.

In this chapter we exclusively consider document classification since most research on polarity classification is done at the document level. We have, however, strong reasons to believe that the majority of insights gained by the experiments presented in this chapter also hold for sentence-level polarity classification since there are many similarities between these two tasks (as shown, for example, by the effective re-usage of features from Chapters 3 and 5 in the experiments of Section 6.6.1).

The work presented in this chapter is also described in (Wiegand & Klakow, 2009a, 2010a).

6.2. Related Work

There are only a few publications dealing with semi-supervised learning on document-level polarity classification. Beineke, Hastie, and Vaithyanathan (2004) combine an unsupervised web-mining approach using point-wise mutual information (Turney, 2002) with labeled training data. Dasgupta and Ng (2009) suggest applying unsupervised learning (i.e. clustering) to classify unambiguous data instances and restrict manual annotation to hard data instances. Aue and Gamon (2005) present experiments using semi-supervised learning focusing on domain adaptation. Neither different algorithms nor feature sets are compared in these works.

In this chapter, we look into adjectives and adverbs as features in detail. Pang et al. (2002) use feature sets exclusively comprising adjectives for supervised document-level polarity classification but report performance to be worse than that of a standard bag-of-words feature set. However, Ng et al. (2006) increase performance significantly by adding to a standard feature set higher-order n-grams in which adjectives are replaced by their in-domain polarity which has been established via manual annotation.

Bootstrapping supervised machine learning classifiers with the help of rule-based classification has been effective in the detection of subjective sentences (Wiebe & Riloff, 2005). The method has also been applied to polarity classification, but so far only on Chinese data (Qiu et al., 2009; Tan et al., 2008). While the performance of bootstrapped classifiers has been compared with out-of-domain classifiers in (Tan et al., 2008), this method

is embedded into a complex bootstrapping system which also extends the vocabulary (or feature set) of the rule-based classifier in (Qiu et al., 2009). Neither of these works examines the relationship to semi-supervised learning, nor discusses various settings of the self-training algorithm, in particular, different feature sets for the supervised classifier.

6.3. Bootstrapping Algorithms

6.3.1. Semi-Supervised Learning Algorithms

We will now briefly describe the different semi-supervised learning algorithms we use in this chapter. Throughout the next sections, we adhere to the following notation:

A document is denoted by x_i (or $\vec{x}_i$ in a vectorial context). Words which are part of some predefined feature set are denoted by w_k. In total, there are N documents encompassing L labeled and U unlabeled documents. A labeled data instance is denoted by x_i^l whereas an unlabeled data instance is labeled as x_i^u. The label c_j of an individual document i is $y_i \in \{-1, 1\}$.

Expectation Maximization Algorithm

The Expectation Maximization Algorithm (EM) for a Naïve Bayes classifier first estimates an expected posterior probability distribution of class label c_j given a document x_i (which can be either labeled or unlabeled), defined as $h(x_i, c_j)$, in the *expectation step*:

$$h(x_i, c_j) = \frac{P(x_i|c_j)}{\sum_k P(x_i|c_k)} \tag{6.1}$$

The *maximization step* uses this expected probability estimate in order to re-estimate class-dependent probabilities of the individual words:

$$P(w_k|c_j) = \frac{\sum_{i=1}^{N} \sum_{\{x_i:w_k \in x_i\}} h(x_i, c_j)}{Z_j} \tag{6.2}$$

where Z_j is a normalization. The new estimates $P(w_k|c_j)$ are used to update the document probabilities $P(x_i|c_j)$ in the expectation step. Equations 6.1 and 6.2 are iterated

until the overall likelihood converges:

$$L = \sum_{i=1}^{L} logP(x_i^l|y_i) + \sum_{j=1}^{U} log \sum_{c} P(x_j^u|c) \tag{6.3}$$

Initially, the probabilities $P(x_i|c_j)$ are directly estimated from the labeled training collection. Since the distribution of the classes is uniform in all the experiments which we use this classifier, we omit the estimation of the class prior.

Transductive Support Vector Machines

Transductive Support Vector Machines (TSVMs) (Joachims, 1999b) use an extended objective function of SVMs:

$$OF_{tsvm} = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=0}^{L}\xi_i + C^*\sum_{j=0}^{U}\xi_j^* \tag{6.4}$$

which includes in addition to a weight vector $\vec{w}$, a regularizer C, and a set of *slack* variables ξ_i for all labeled instances, an extra regularizer C^* and an extra set of slack variables ξ_j^* for unlabeled instances.

The algorithm first learns a base model M_{svm} using the original objective function of SVMs. All unlabeled instances are labeled with that model. A new model M_{tsvm}^i is created by minimizing the extended objective function OF_{tsvm} and using the predicted labels of the unlabeled instances of M_{svm} as a proxy. A small C^* is chosen. Then, the algorithm iteratively computes improved models M_{tsvm}^{i+1} by swapping two opposing labels of some originally unlabeled documents which have been misclassified according to M_{tsvm}^i. C^* is increased with each iteration step. If there are no more misclassifications, the final model has been found.

Spectral Graph Transduction

In Spectral Graph Transduction (SGT) (Joachims, 2003), all data x_i of a collection (i.e. labeled and unlabeled) are represented as a symmetrized and similarity-weighted k

nearest-neighbour (knn) graph G. Its adjacency matrix is defined as $A = A' + A'^T$ where

$$A'_{ij} = \begin{cases} \frac{sim(\vec{x_i},\vec{x_j})}{\sum_{\vec{x}_k \in knn(\vec{x_i})} sim(\vec{x}_i,\vec{x}_k)} & \text{if } \vec{x}_j \in knn(\vec{x_i}) \\ 0 & \text{else} \end{cases} \tag{6.5}$$

and $sim(\cdot,\cdot)$ is any common similarity function. The graph G is decomposed into its spectrum. For this, the smallest 2 to $d+1$ eigenvalues and eigenvectors of the normalized Laplacian $L = B^{-1}(B - A)$ where B is the diagonal degree matrix with $B_{ii} = \sum_j A_{ij}$ are computed. The spectrum is used for minimizing the normalized graph cut:

$$\min_{\forall y_i} \frac{cut(G^+, G^-)}{|\{i : y_i = 1\}||\{i : y_i = -1\}|} \tag{6.6}$$

where G^+ and G^- denote the set of positive and negative classified vertices in the graph. The cut-value $cut(G^+, G^-) = \sum_{i \in G^+} \sum_{j \in G^-} A_{ij}$ is the sum of the edge-weights of a cut partitioning the graph into two clusters.

6.3.2. Self-Training a Polarity Classifier using the Output of a Rule-Based Classifier

The idea of this bootstrapping method is that a domain-independent rule-based classifier is used to label an unlabeled dataset. Unlike semi-supervised learning (see Section 6.3.1), no labeled training data are used. The only knowledge available is encoded in the rule-based classifier. In polarity classification, the rule-based classifier typically counts the number of positive and negative polar expressions within a data instance (i.e. a document) and assigns it the polarity type having the majority of polar expressions. The data instances labeled by the rule-based classifier with a high confidence serve as labeled training data for a supervised machine learning classifier. The supervised classifier is typically trained with bag-of-words features.

Ideally, the resulting supervised classifier is more robust on the domain on which it was trained than the rule-based classifier. The improvement can be explained by the fact that the rule-based classifier only comprises domain-independent knowledge, i.e. in polarity classification this corresponds to the knowledge of domain-independent polar expressions. The supervised classifier, however, makes use of domain-specific features,

i.e. words such as *crunchy*$^+$ (food domain) or *buggy*$^-$ (computer domain), which are not part of the rule-based classifier. It may also learn to correct polar expressions that are specified in the polarity lexicon but have a wrong polarity type on the target domain. A reason for a type mismatch may be that a polar expression is ambiguous and contains different polarity types throughout the different domains (and common polarity lexicons usually only specify one polarity type per entry). For instance, in the movie domain the polar expression *cheap* is predominantly negative, as it can be found in expressions, such as *cheap films*, *cheap special-effects* etc. In the computer domain, however, it is predominantly positive as it appears in expressions, such as in *cheap price*. If such a polar expression occurs in sufficient documents which the rule-based classifier has labeled correctly, then the supervised learner may learn the correct polarity type for this ambiguous expression on that domain, despite the fact that the opposite type is specified in the polarity lexicon.

We argue that using a rule-based classifier instead of few labeled (in-domain) data instances – as is the case in semi-supervised learning – is more worthwhile since we exploit two different types of features being domain-independent polar expressions and domain-specific bag of words which are known to be complementary (Andreevskaia & Bergler, 2008). Semi-supervised learning usually just makes use of one homogeneous feature set.

Figure 6.1 illustrates both semi-supervised learning and self-training using a rule-based classifier for bootstrapping.

For reasons of simplicity, we will often refer to the specific version of self-training we consider in this chapter (i.e. self-training using a rule-based classifier) as plain *self-training* in the following sections.

6.4. Data

In this chapter, we use both the dataset of *IMDb* movie reviews (Pang et al., 2002) and a set of reviews extracted from *Rate-It-All*[1]. We evaluate on the former because it is

[1] `http://www.rateitall.com`

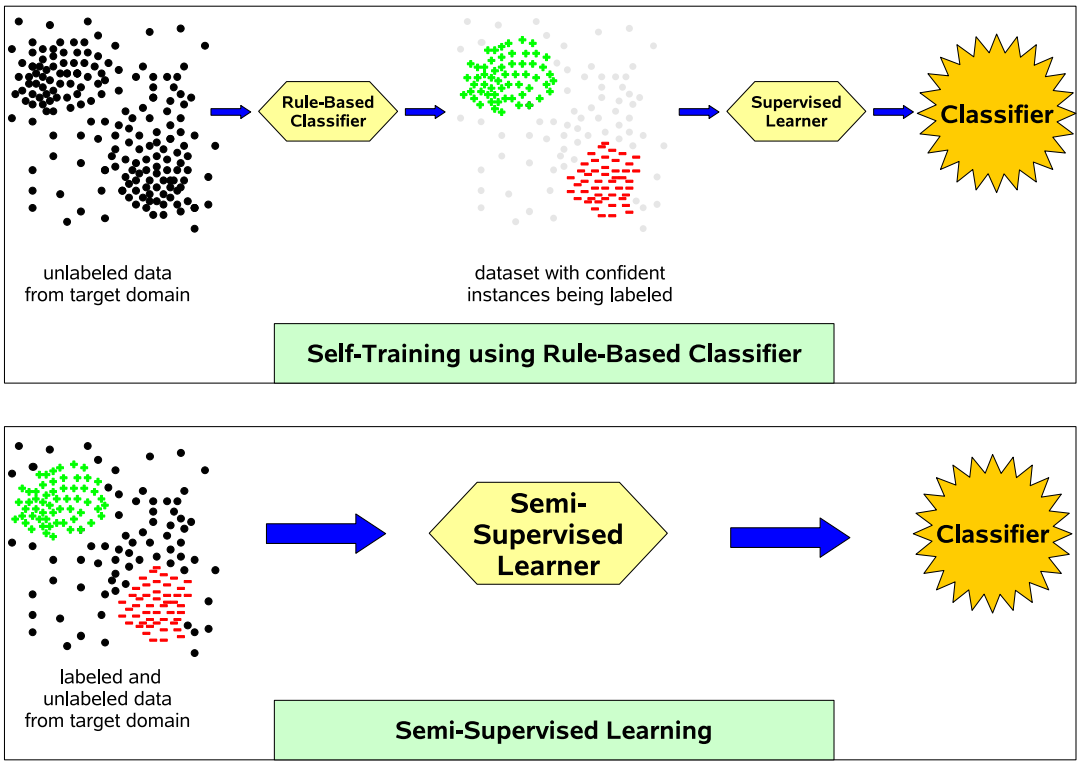


Figure 6.1.: Comparison of semi-supervised learning and self-training using a rule-based classifier for bootstrapping.

considered a benchmark dataset for polarity classification. The additional data are used to show that our findings are valid throughout different domains. We chose four domains from the list of *Topic Categories* of the website[2] which we thought are very different from the movie domain[3] and for which we could extract sufficient training data. We took *Computer & Internet (computer)*, *Products (products)*, *Sports & Recreation (sports)*, and *Travel, Food, & Culture (travel)*. Table 6.1 lists the properties of the corpora from the different domains. We follow the method from previous work (Blitzer et al., 2007) to infer the polarity of the reviews from Rate-It-All. Ratings with less than 3 stars are considered negative reviews whereas ratings with more than 3 stars are positive reviews. 3 star reviews are labeled *mixed*. The actual class of these reviews is unknown. Usually a 3 star review should be neutral in the sense that it equally enumerates both positive and negative aspects about a certain topic, so that a definite verdict in favor or against it is not possible. That is also why we cannot assign these instances either of the other two groups previously mentioned, i.e. *positive* and *negative*. During a manual inspection of some randomly chosen instances, however, we also found definite positive and negative reviews among 3 star reviews. For this work, we leave these instances in the category of mixed reviews. We only used reviews in our experiments having at least 3 sentences in order to rule out too fragmentary instances.

6.5. Semi-Supervised Polarity Classification

I assume that discriminative feature sets are far more important in semi-supervised learning than in supervised learning since there is less reliable information contained in small labeled datasets. This is why I put emphasis on the discussion of feature sets or feature selection methods in this section. Since we exclusively consider polarity classification at document level, we restrict the type of features to bag of words since it is known to be very effective for document-level classification (Ng et al., 2006).

[2]The data were downloaded in 2008, so the appearance and content of the website may have changed.

[3]This is why we did not use the *person* domain from Chapter 4 as it mostly concerns celebrities also being discussed in the movie domain.

Table 6.1.: Properties of the different domain corpora.

Domain	Source	4 & 5 Stars† Positive	3 Stars† Mixed	1 & 2 Stars† Negative	Vocabulary Size
computer	*Rate-It-All*	952	428	1253	15083
products	*Rate-It-All*	2292	554	1342	21975
sports	*Rate-It-All*	4975	725	1348	24811
travel	*Rate-It-All*	9397	1772	3289	38819
movies	*IMDb*	1000	0	1000	50920

(†only relates to the *Rate-It-All* data)

6.5.1. The Different Feature Sets

In the context of semi-supervised document-level text classification the purpose of feature selection is to remove features that are irrelevant or noisy for a particular classification task. The elimination of these features does not only result in an increase in efficiency but may also improve the Accuracy of a classifier.

Term Frequency Cut-off

The simplest feature selection method is using a term-frequency cut-off. The rationale behind this is that rarely observed terms do not contribute to a good classifier. Usually, this selection method is combined with stop-word removal.[4] Very frequently occurring terms, in particular function words, are not considered to be predictive for a particular class label, since they are uniformly distributed throughout all classes.

Polarity Lexicons

In our experiments, we use Appraisal Groups (AG) (Whitelaw, Garg, & Argamon, 2005), General Inquirer (GI) (Stone et al., 1966), the Subjectivity Lexicon from the MPQA-project (MPQA) (Wilson et al., 2005), and SentiWordNet (SWN) (Esuli & Sebastiani,

[4]We use a publicly available list of stopwords: `http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words`

2006b). From GI we use all polar expressions and from AG we only consider *orientation words* that are not neutral (Whitelaw et al., 2005). From MPQA, we use – like in previous chapters – both *weak* and *strong* subjective words (Wilson et al., 2005) with either positive or negative prior polarity.[5] These polarity lexicons have been successfully applied to polarity classification (Kennedy & Inkpen, 2005; Wilson et al., 2005; Whitelaw et al., 2005).

SentiWordNet (SWN) does not specify the polarity of individual words but synsets (i.e. senses of words). The database provides a non-negative polarity score $senseScore(s, p)$ for each synset s and polarity $p \in \{+, -\}$. Neutral polarity strength is denoted by 0. Usually, words have different senses associated with them. There are even words which have both senses with positive and negative polarity. Therefore, most words have various polarity scores associated with them. Our goal is to derive a unique polarity for each word with a corresponding score denoting its strength. We use the unique scores in order to find a subset of SWN with highly polar expressions. We estimate the strength of a word w and a polarity p, i.e. $wordScore(w, p)$, by:

$$wordScore(w, p) = \max_{s} [senseScore(s, p)] \tag{6.7}$$

where $s \in synsets(w)$. The final polarity of the word, i.e. $pol(w)$, is the polarity with the maximum polarity score:

$$pol(w) = \arg\max_{p} [wordScore(w, p)] \tag{6.8}$$

The unique score denoting the polarity strength is defined as:

$$strength(w) = \max_{p} [wordScore(w, p)] \tag{6.9}$$

By using only the subset of SWN instead of the entire set (we chose all words with $strength(w) \geq 0.5$), we increased the Accuracy of the semi-supervised classifiers by approximately 1.5% on average. We reduced the size of the initial version by 70% which substantially increased the efficiency of model learning. A subset of SWN based on taking the average rather than taking the maximum produced slightly worse results.

[5] Note that just focusing on the strong entries resulted in a decrease in performance.

Adjectives and Adverbs

Adjectives, such as *superb* or *poor*, are usually regarded as very predictive words for polarity classification. Their impact on semi-supervised learning has not yet been examined. Even if this feature set is too small for supervised learning (Pang et al., 2002), it might still be effective in semi-supervised learning. In contrast to supervised learning, large feature sets which are noisy cannot be compensated by the information contained in many labeled documents. Smaller but more predictive feature sets are preferable. We use feature sets of frequently occurring adjectives and adverbs in our document collection. The feature sets are extracted using the C&C part-of-speech tagger.[6] After manually inspecting the 600 most frequent stemmed adjectives and adverbs from the movie domain dataset (Pang et al., 2002), we estimate that more than 20% of the expressions are ambiguous with regard to part of speech.[7] Thus, our selection method if combined with stemming also captures some polar verbs and nouns. By looking at the list of extracted adjectives and adverbs from other domains, we observed that unlike current polarity lexicons this method allows both some colloquial expressions, such as *crappy*, and highly domain-dependent polar expressions, such as *creamy* or *crunchy* from the food domain, to be detected.

Optimal Feature Size

Table 6.2 lists the optimal size[8] of the different feature sets we used in our experiments.[9] By far, the smallest feature set are adjectives and adverbs; the largest feature set is SWN.

6.5.2. Experiments

The results of *all* our experiments below are reported on the basis of 20 randomized partitionings. Each partitioning comprises a labeled dataset of varying length for train-

[6] `http://svn.ask.it.usyd.edu.au/trac/candc`

[7] For example, *Interesting* (adj) and *interests* (noun) are both reduced to *interest*.

[8] The optimal size was determined by testing all semi-supervised algorithms trained on various amounts of labeled documents and $1,000$ unlabeled documents.

[9] Due to the stemming we applied some of the entries in the original polarity lexicons were conflated.

Table 6.2.: Optimal size of the different feature sets.

Feature Set	**Type**	**#Words**
top n words	statistical selection	3000
top n non-stopwords	statistical selection	2000
top n adjectives & adverbs	statistical & linguistic selection	**600**
Appraisal Groups (AG)	manual polarity lexicon	2014
General Inquirer (GI)	manual polarity lexicon	2882
Subjectivity Lexicon (MPQA)	manual polarity lexicon	4615
SentiWordNet (SWN)	semi-automatic polarity lexicon	11366

ing and another dataset comprising 1,000 documents used as unlabeled training data and test data. We adhere to this configuration since it is required by the toolkit we use. However, it is not uncommon to use test data as unlabeled training data in semi-supervised learning (Aue & Gamon, 2005; Joachims, 1999b, 2003). We also experimented with larger amounts of unlabeled data but did not measure any improvement in performance. The labeled training data and the test data are always mutually exclusive. We report the results of experiments carried out on the movie review database (Pang et al., 2002) (benchmark dataset) and the results of cross-domain experiments using reviews from *Rate-It-All*. Since the movie dataset is the standard dataset we will discuss our experiments on this domain in more detail. The movie dataset comprises 2,000 reviews whereas for the other domains we could only acquire 1,800 documents per domain. For the sake of simplicity, all datasets are balanced. We report statistical significance on the basis of a paired t-test using 0.05 as the significance level. We only state the results of the optimally sized feature sets (see Section 6.5.1). Since there is no difference in performance between the optimally sized feature set with the most frequent words and the optimally sized feature set with most frequent non-stopwords, we only evaluated the latter feature set. We used *SVMLight*[10] for SVMs and TSVMs and *SGTLight*[11] for

[10] `http://svmlight.joachims.org`
[11] `http://sgt.joachims.org`

SGT. We evaluate the results using Accuracy (see also Appendix A.1). Feature vectors consist of tf-idf weighted words appearing in the pre-defined feature set normalized by document length. This produced the best results throughout our experiments. Further modifications of the standard configuration of *SVMLight* (e.g. by changing regularization parameters) did not improve performance. We also confirm the results from (Aue & Gamon, 2005) who report that further modifications on EM, i.e. by weighting the unlabeled data[12], do not improve performance. For *SGTLight* we mainly adhered to the standard configuration (as discussed in (Joachims, 2003)). Since we had no development data for optimizing the only task-sensitive parameter k, i.e. the number of nearest neighbours, we simply took the optimized value for the only text classification corpus tested in previous work (Joachims, 2003) (i.e. *Reuters collection*). The current choice (i.e. $k = 800$) should thus guarantee a fairly unbiased setting. EM is smoothed by absolute discounting (Zhai & Lafferty, 2001). All classifiers are run with a reasonable parameter setting but we did not attempt to tune the parameters to the current task. We also stem the entire text since some polarity lexicons we use also include lemmas of inflectional words, such as nouns and verbs. Moreover, stemming has considerable advantages for the feature set comprising adjectives and adverbs (see discussion above). In-domain feature sets (i.e. frequent non-stopwords and frequent adjectives and adverbs) are obtained by considering the entire dataset of a particular domain.

Unsupervised Algorithm using Different Polarity Lexicons (*Movie Domain*)

Before comparing the different polarity lexicons in the context of semi-supervised learning, we shortly display their performance using a completely unsupervised algorithm. A test document is assigned the polarity of the majority of polar expressions in that document. This experiment should give an idea of the intrinsic predictiveness of the polarity lexicons. Note that we refrain from using any further linguistic modeling, e.g. negation modeling, in order to improve this baseline since we also run the semi-supervised classifiers with plain bag-of-words features (i.e. we carry out feature selection but beyond that we do not

[12]Note that this is similar to regularization in TSVMs.

Table 6.3.: Accuracy of unsupervised algorithm using different polarity lexicons (movie domain).

SWN	AG	GI	MPQA	GI+Turney
54.20	54.45	59.90	61.75	63.30

incorporate any expressive high-level features). Table 6.3 lists the results (on the movie domain). Though all lexicons perform significantly better than the random baseline (i.e. 50%), the best performance of MPQA with 61.75 is still very low.

We also evaluated an extension GI+Turney which weights the polar expressions in GI according to the association scores to a very small number of manually selected highly polar seed words, such as *excellent* or *poor* (Turney & Littman, 2003).[13] The scores for entries in GI are calculated in the same way as the scores for words in the web-based lexicon induction method using *Pointwise Mutual Information* (Turney, 2002). The improvement (towards GI) is significant, even though the scores have been gained by domain-independent web-data.

In the following, we show that very small amounts of labeled in-domain documents can produce significantly better results using semi-supervised learning.

Comparison of the Different Polarity Lexicons with Other Feature Sets (*Movie Domain*)

Table 6.4 displays the performance of different (semi-supervised) classifiers on different feature sets (again on the movie domain). On average, polarity lexicons perform significantly better than the top 2000 non-stopwords. The same holds for an inexpensive small feature set of in-domain adjectives and adverbs. On EM, we even achieved the best performance with the latter feature set. The best performing feature set for the movie dataset is AG. On several configurations, it is even significantly better than any other feature set using semi-supervised learning.

[13] Unfortunately, currently only the weights for entries of GI are available to us.

Table 6.4.: Accuracy of different classifiers on different feature sets using different amounts of labeled documents (movie domain).

(a) 20 labeled documents

	Top 2000	**SWN**	**MPQA**	**GI**	**AG**	**Adj**
SVMs	59.81	61.24*	*63.07**	61.48*	62.22*	61.44*
EM	67.50	67.31	68.73	66.63	69.44*	*69.54**
TSVMs	64.57	67.04*	66.58*	65.53	*68.87**	68.37*
SGT	62.60	67.39*	67.10*	66.14*	***70.28****†	66.58*

(b) 200 labeled documents

	Top 2000	**SWN**	**MPQA**	**GI**	**AG**	**Adj**
SVMs	72.05	74.93*	74.35*	72.72	*75.88**†	73.14*
EM	73.44	76.46*	75.02*	73.80	75.46*	*77.32**
TSVMs	73.48	76.80*	75.73*	74.72*	*77.89**†	75.12*
SGT	70.91	77.55*	77.78*	75.12*	***80.21****†	76.90*

*: significantly better than Top 2000 on the basis of a paired t-test using $p < 0.05$

†: significantly better than any other feature set on the basis of a paired t-test using $p < 0.05$

Semi-Supervised Classifiers (*Movie Domain*)

We compared all different learning algorithms using their respective best feature sets. Figure 6.2 displays the results. (Again, these experiments have been run on the movie domain.) All semi-supervised algorithms are better than the strict supervised baseline (i.e. SVMs trained on AG) on small amounts of labeled data. EM gets worse than SVMs trained on AG when more than 400 labeled documents are used, but still outperforms SVMs trained on top 2000 non-stopwords when less than 700 labeled documents are used. TSVMs and SGT, on the other hand, constantly perform better than SVMs.

Clearly, the best classifier is SGT which, with the exception of $1,000$ labeled data, is always significantly better than any other classifier tested. At approximately 200 labeled documents, SGT already performs as well as SVMs trained on a standard feature set (i.e. top 2000 non-stopwords)

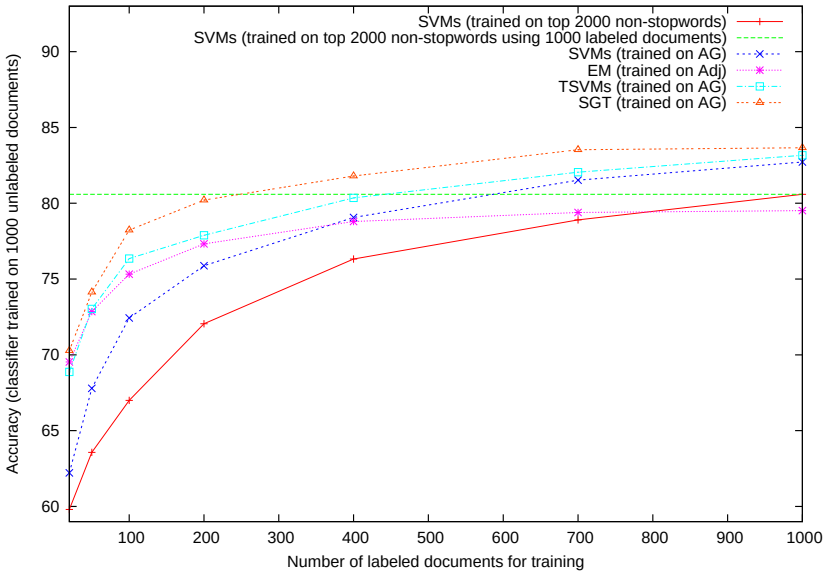


Figure 6.2.: Performance of different learning algorithms on the best respective feature set (movie domain).

Just using 20 labeled documents offers an increase by 7% in performance in comparison to the best unsupervised classifier (i.e. GI+Turney displayed in Table 6.3).

Complex Feature Sets that Do not Improve Performance

Contrary to our expectations, adding explicit polarity information to the feature set by including the number of positive and negative polar expressions according to the pertaining polarity lexicon did not improve performance.

We also experimented with more expressive features by adding bigrams with one token being a polar expression, an adjective, or an adverb. On semi-supervised learning we did not measure any increase in performance. We assume that this is due to data-sparseness. Similar to (Ng et al., 2006), we observed an increase in performance by approximately 2% on supervised classifiers (when more than 400 labeled documents are used).

Cross-Domain Experiments

In order to validate our findings from the movie domain, we repeat some of the previous experiments on other domain corpora using the reviews from *Rate-It-All*. In particular, we want to know whether semi-supervised learning works there as well, whether SGT outperforms other classifiers, whether polarity lexicons improve performance, and whether adjectives and adverbs produce classifiers competitive to average polarity lexicons. We do not attempt to carry out detailed domain studies which would be beyond the scope of this section.

Table 6.5 lists the average performance of all classifiers on different feature sets using 20 labeled documents. For the sake of completeness we also include the results from the movie domain. There is no significant difference among the feature sets using SVMs, but there is a difference between top 2000 non-stopwords and the remaining feature sets on semi-supervised classification (with the exception of EM). All polarity lexicons and adjectives and adverbs perform significantly better than top 2000 non-stopwords using TSVMs and SGT. On average, the performance of EM is worse than any of the other semi-supervised classifiers. The results of TSVMs and SGT are similar to our previous observations on the benchmark dataset. SGT is the best performing classifier (in particular in combination with adjectives).

Table 6.5.: Average Accuracy of different semi-supervised classifiers across all domains using different feature sets (trained on 20 labeled documents & 1,000 unlabeled documents).

	Top 2000	**SWN**	**MPQA**	**GI**	**AG**	**Adj**
SVMs	**61.17**	61.13	60.81	**61.17**	60.77	60.68
EM	64.41	65.09*	64.08*	63.88*	65.10*	**65.22***
TSVMs	63.87	66.79*	66.51*	66.26*	65.98*	**67.20***
SGT	64.60*	66.92*	67.69*	67.83*	67.22*	**68.30***

*: significantly better than SVMs on the basis of a paired t-test using $p < 0.05$

Table 6.6 shows the performance on the individual domains and feature sets using 20 labeled documents on SGT. On average, semi-supervised learning improves performance significantly over supervised learning. On some domains (e.g. *computer*) using a standard feature set (i.e. using top 2000 non-stopwords in the collection) produces good results. However, on some other domains, such as *travel*, there is no improvement whatsoever. Polarity lexicons can perform significantly better than top 2000 non-stopwords (e.g. GI on *travel* or, most notably, AG on *movie*) but there are also domains where they are actually worse than the standard feature set (e.g. the *sports* domain). There is no polarity lexicon which consistently outperforms all other polarity lexicons on all domains. A feature set comprising in-domain adjectives and adverbs, however, is more robust: Firstly, it never performs worse than the standard feature set. Secondly, it is never significantly worse than the average performance of polarity lexicons and, thirdly, there might be some domain, such as *sports*, where it outperforms any other feature set. Considering the small effort required to generate such a feature set should make it particularly attractive.

Table 6.6.: Accuracy of SGT on different domains using different feature sets (trained on 20 labeled documents & 1,000 unlabeled documents).

	SVMs	**SGT**					
Domain	**Top 2000**	**Top 2000**	**SWN**	**MPQA**	**GI**	**AG**	**Adj**
computer	67.75	73.88*	*75.77**†	74.77*	73.95*	73.74*	74.51*
products	62.38	67.20*	68.45*†	68.40*†	*69.84**†	68.44*†	68.79*†
sports	57.96	61.83*	57.57	59.80*	60.62*	58.53	*63.55**
travel	57.95	57.48	65.44*†	68.37*†	*68.62**†	65.09*†	68.05*†
movies	59.81	62.60*	67.39*†	67.10*†	66.14*†	*70.28**†	66.58*†
average	61.17	64.60*	66.92*	67.69*	67.83*	67.22*	**68.30***

*: significantly better than SVMs using Top 2000 on the basis of a paired t-test using $p < 0.05$

†: significantly better than SGT using Top 2000 on the basis of a paired t-test using $p < 0.05$

Figure 6.3 displays the performance of SGT on various feature sets averaged over all domains using various amounts of labeled training data. SGT only significantly outper-

forms SVMs when less than 200 labeled documents are used. Therefore, we restricted the figure to the range ending at that size. The lower performance of the averaged results must be due to some properties of the *Rate-It-All* data (either noise or the dataset is more difficult) since the individual performance of the semi-supervised classifiers on the movie domain was significantly better. Despite the lower performance, we can still use the averaged results to characterize the relation between the different feature sets in semi-supervised learning. Both polarity lexicons and adjectives and adverbs are significantly better than top 2000 non-stopwords and there is no significant difference between polarity lexicons and adjectives and adverbs.

All these results support both the competitiveness of adjective and adverbs and the robustness of SGT. Given the best feature set in a particular domain, the average gain in improvement compared to SVMs only trained on 20 labeled documents using top 2000 non-stopwords is approximately 8.5% when SGT is used. This is a clear indication that semi-supervised learning for polarity classification works across all domains when only tiny amounts of labeled data are used.

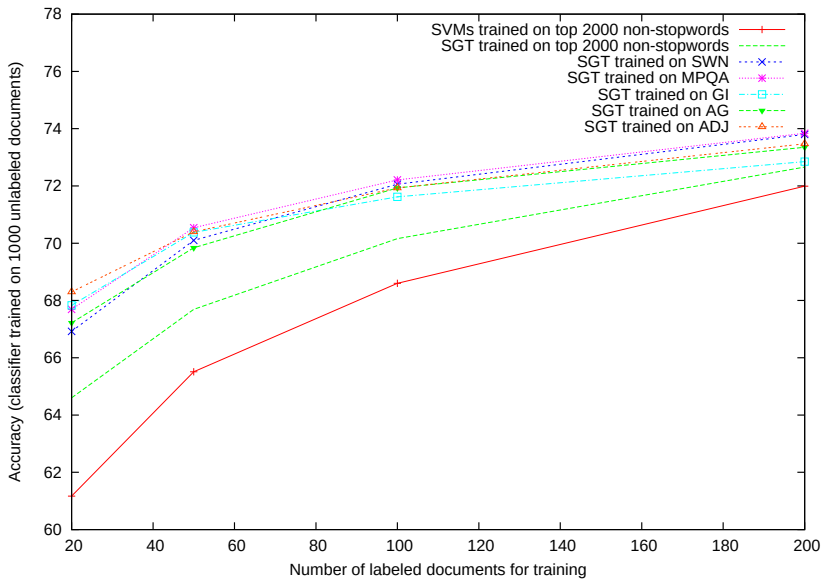


Figure 6.3.: SGT trained on different amounts of labeled data and different feature sets averaged over all domains (1,000 unlabeled documents).

6.5.3. Conclusion of Experiments on Semi-Supervised Learning

In this section we have shown that semi-supervised learning can be successfully applied to document-level polarity classification. Significant improvement over supervised classification can be achieved across all domains when less than 200 labeled documents are available. On the movie domain we even achieved improved performance across all amounts of labeled training data. SGT is the classifier which produces better results than all other semi-supervised classifiers used in our experiments. On average, polarity lexicons and adjectives and adverbs perform better than just using frequent in-domain non-stopwords. Adjectives and adverbs are less expensive to obtain and more robust throughout different domains. Thus, these experiments show that the consideration of linguistic knowledge, be it the knowledge of polar expressions or the knowledge of adjectives and adverbs, is helpful for semi-supervised learning.

6.6. Bootstrapping Supervised Polarity Classifiers using Rule-Based Classification

I assume that the performance of supervised polarity classifiers bootstrapped with the help of rule-based classifiers depends on two components:

- the type of rule-based classifier that is chosen
- the feature set on which the supervised classifier is trained

This is why I will focus on these two aspects in the discussion of this method.

6.6.1. Rule-Based Classifier

In the following, we describe how a polarity lexicon is converted to a rule-based polarity classifier. The polarity lexicon, the list of other important word classes being intensifiers, negation expressions (including the rules to disambiguate them), and polarity shifters are, as in the experiments from the previous chapters, taken from the *MPQA* project (Wilson

et al., 2005). We chose this resource since due to its feature diversity it allows the construction of the most complex polarity classifier.

Feature Extraction

Any word in a review that is not included in a polarity lexicon is discarded. Positive words (e.g. *excellent*) are assigned the value +1, negative words (e.g. *awful*) −1, respectively.

Basic Word Sense Disambiguation with Part-of-Speech Tags

The polarity lexicon we use has part-of-speech tags attached to polar expressions in order to disambiguate them, e.g. the word *like* is either a polar verb or a preposition (in which case it is meaningless for polarity classification). We identify words as polar expressions only if their part-of-speech tags[14] also match the specification in the lexicon. This can be considered as some basic form of word sense disambiguation.

Negation Modeling

If a polar expression occurs within the scope of a negation, its polarity is reversed (e.g. $[not\ \text{nice}^{+}]^{-}$). The negation modeling we use in this chapter, which includes both the disambiguation of potential negation expressions and the usage of polarity shifters, is identical to the method described in Chapter 5.4.3.

Heuristic Weighting

So far, all polar expressions contained in the polarity lexicon are assigned the same absolute weight, i.e. (±)1. This does not reflect reality. Polar expressions differ in their individual polar intensity or, in case of ambiguous words, in their likelihood to convey polarity. Therefore, they should not obtain a uniform weight. We propose a heuristic weighting scheme based on particular properties of polar expressions. We focus on properties that have been effectively incorporated into features in Chapters 3 and 5 on sentence-level polarity classification. The properties considered for heuristic weighting

[14] For part-of-speech tagging, we again use the *C&C* tagger.

have already been motivated and proven effective in previous work (Kennedy & Inkpen, 2005; Pang et al., 2002).

Intuitively, strong polar expressions, such as *chaotic*, should obtain a higher weight than weak polar expressions, such as *bulky*. The same holds for intensified polar expressions, i.e. an ordinary (weak) polar expression has a similar polar intensity when it is modified by an intensifier as a strong polar expression, e.g. *extremely disordered* and *chaotic*.

The part of speech of a polar expression usually sheds light on the level of ambiguity of the word. If a polar expression is an *adjective*, its prior probability of being polar is much higher than the one of polar expressions with other parts of speech, such as verbs or nouns (Pang et al., 2002). Therefore, polar adjectives should obtain a larger weight than polar expressions with other parts of speech.

Since there are no development data in order to adjust the weights for the previously mentioned properties, we propose to simply *double* the value of a polar expression if either of these properties applies. If n of these properties apply for a polar expression, then its value is doubled n times. For instance, an intensified adjective is assigned the value of 4, i.e. $2 \cdot 2$.

Classification

For each data instance the *contextual* scores assigned to the individual polar expressions are summed. If the sum is positive, then the instance is classified as positive. It is classified as negative, if the sum is negative. We assign to all cases in which the sum is 0 the polarity type which gives best performance on that individual dataset (which is usually negative polarity). Thus, we have a stronger baseline that is to be beaten by self-training.

Note that the prediction score of a data instance, i.e. the sum of contextual scores of the polar expressions, can also be interpreted as a confidence score. This property is vital for effectively using this rule-based classifier in self-training. Thus, previously mentioned instances with a score of 0, for example, are unlikely to occur in the labeled training

set since it only includes instances labeled with a high confidence score. The sum of contextual scores is normalized by the overall number of tokens in a test instance. This normalization reflects the density of polar expressions within the instance. The greater the density of polar expressions of a particular type is in a text, the more likely the text conveys that polarity.

Figure 6.4 summarizes all steps of the rule-based classifier.

1. Lexicon loading, i.e. polar expressions, negation words, and intensifiers
2. Preprocessing:
   (i) Stem test instance.
   (ii) Apply part-of-speech tagging to test instance.
3. Polar expression marking:
   (i) Check whether part-of-speech tag of potential polar expression matches lexical entry (*basic word sense disambiguation*).
   (ii) Mark strong polar expressions.
4. Negation modeling:
   (i) Identify potential negation words (including polarity shifters).
   (ii) Disambiguate negation words.
   (iii) Reverse polarity of polar expression in scope of (genuine) negation.
5. Intensifier marking
6. Heuristic weighting: double weight in case polar expression is:
   (i) a strong polar expression
   (ii) an intensified polar expression
   (iii) a polar adjective.
7. Classification: assign test instance the polarity type with the largest (normalized) sum of scores.

Figure 6.4.: Rule-based classifier.

Table 6.7.: Properties of the different rule-based classifiers.

Properties	$\mathbf{RB}_{Plain}$	$\mathbf{RB}_{bWSD}$	$\mathbf{RB}_{Neg}$	$\mathbf{RB}_{Weight}$
basic word sense disambiguation		✓	✓	✓
negation modeling			✓	✓
heuristic weighting				✓

Table 6.8.: Description of the different feature sets.

Feature Set	**Abbreviation**
the 2000 most frequent non-stopwords in the domain corpus	Top2000
the 600 most frequent adjectives and adverbs in the domain corpus	Adj600
all polar expressions within the polarity lexicon	MPQA
all unigrams in the domain corpus	Uni
all unigrams and bigrams in the domain corpus	Uni+Bi

Different Versions of Classifiers

We define four different types of rule-based classifiers. They differ in complexity. The simplest classifier, i.e. RB_{Plain}, does not contain word sense disambiguation, negation modeling, or heuristic weighting. RB_{bWSD} is like RB_{Plain} but also contains basic word sense disambiguation. RB_{Neg} is like RB_{bWSD} but also contains negation modeling. The most complex classifier, i.e. RB_{Weight}, is precisely the algorithm presented in the previous sections. Table 6.7 summarizes the different classifiers with their respective properties.

6.6.2. Feature Sets

Table 6.8 lists the different feature sets we examine for the supervised classifier (within self-training) and the semi-supervised classifiers. We list the feature sets along their abbreviation with which they will henceforth be addressed. The first three features (i.e. Top2000, Adj600, and MPQA) have been used in the previous experiments on semi-supervised learning (Section 6.5). They all remove noise contained in the overall

vocabulary of a domain corpus. The last two features (i.e. Uni and Bi) are known to be effective for supervised polarity classification (Ng et al., 2006). Bigrams can be helpful in addition to unigrams since they take into account some context of polar expressions. Thus, crucial constructions, such as negation (*[not nice]*$^{-}$) or intensification (*[extremely nice]*$^{++}$), can be captured. Moreover, multiword polar expressions, such as *[low tax]*$^{+}$ or *[low grades]*$^{-}$, can be represented as individual features. Unfortunately, bigram features are also fairly sparse and contain a considerable amount of noise.

6.6.3. Experiments

For the following experiments we mainly adhere to the settings of our experiments on semi-supervised learning (see Section 6.5). We deliberately chose these settings in favor of semi-supervised learning in order to have a strong baseline for the proposed self-training method. We again use a balanced subset (randomly generated) for each domain. The *Rate-It-All* dataset consists of $1,800$ data instances per domain, whereas the *IMDb* dataset consists of $2,000$ data instances. We just consider (definite) positive and (definite) negative reviews. The rule-based classifiers and the self-trained classifiers (bootstrapped with the help of rule-based classification) are evaluated on the entire domain dataset. The $1,000$ most highly-ranked data instances (i.e. 500 positive and 500 negative instances) are chosen as training data for the supervised classifier. This setting provided good performance in our initial experiments. For the supervised classifier, we chose SVMs. All words are stemmed. We report statistical significance on the basis of a paired t-test using 0.05 as the significance level unless we explicitly state otherwise. We evaluate the results using Accuracy and F-Measure (see also Appendix A.1).

Comparison of Different Rule-Based Classifiers

Table 6.9 shows the results of the different rule-based classifiers across the different domains. On average, the more complex the rule-based classifier gets, the better it performs. The only notable exceptions are the *products* domain (from RB_{Neg} to RB_{Weight}) and the *sports* domain (from RB_{Plain} to RB_{bWSD}). We assume, however, that in particular those

results in the sports domain are heavily affected by the high degree of spelling errors. On average (i.e. considering all domains), however, the improvements are statistically significant.

Table 6.9.: Comparison of different rule-based classifiers (RB) (for each domain, performance is evaluated on a balanced corpus).

Domain	$\mathbf{RB}_{Plain}$	$\mathbf{RB}_{bWSD}$	$\mathbf{RB}_{Neg}$	$\mathbf{RB}_{Weight}$
computer	64.11	70.61	73.56*	74.28
products	60.78	66.06*	71.06*	70.94
sports	64.33	64.39	67.50	68.89
travel	64.61	67.39	70.72*	72.61
movies	61.75	64.80*	67.85*	71.30*
average	63.12	66.65*	70.14*	71.60*

*: significantly better than *all* less complex rule-based classifiers on the basis of a χ^2 test using $p < 0.05$

Self-Training with Different Rule-Based Classifiers and Different Feature Sets

Table 6.10 compares self-training (SelfTr) using different rule-based classifiers and different feature sets for the embedded supervised classifier. In addition to Accuracy, we also listed the F-Measure of the two different classes. The results are averaged over all domains. With the exception of RB_{Neg} in combination with Top2000 and MPQA, there is always a significant improvement from a rule-based classifier to the corresponding self-trained version. If Top2000 or MPQA is used, there is a drop in performance from RB_{Neg} to SelfTr in the *sports* domain. Improving a rule-based classifier also results in an improvement of the self-trained classifier. With exception of SelfTr(RB_{Plain}) to SelfTr(RB_{bWSD}) this is even significant.

The feature set producing the best results is Uni+Bi. Uni+Bi is statistically significantly better than Uni. This means that, as far as feature design is concerned, the supervised classifier within self-training behaves similar to ordinary supervised classification (Ng et al., 2006). Unlike in semi-supervised learning, a noiseless feature set is not

necessary. Best performance of SelfTr using a large set of polar expressions is reported in (Qiu et al., 2009). The feature set comprises an open-domain polarity lexicon and is automatically extended by domain-specific expressions. Our results suggest a less complex alternative. Using SelfTr with unigrams and bigrams (i.e. SelfTr$_{Uni+Bi}$) already provides better classifiers than SelfTr with a polarity lexicon (i.e. SelfTr$_{MPQA}$). The increase is approximately 3%.

It is also worth pointing out that the gain in performance that is achieved by improving a basic rule-based classifier (i.e. RB$_{Plain}$) by modeling constructions (i.e. RB$_{Weight}$) is the same as is gained by just self-training it with the best feature set (i.e. SelfTr$_{Uni+Bi}$).

The relation between the F-Measures of the two different classes differs between RB and SelfTr. In RB, the score of the positive class is always significantly better than the score of the negative class. This is consistent with previous findings (Andreevskaia & Bergler, 2008). The gap between the two classes, however, varies depending on the complexity of the classifier. In RB$_{Plain}$, the gap is 17.45%, whereas it is less than 6% in RB$_{Neg}$ and RB$_{Weight}$. In SelfTr, the F-Measure of the negative class is usually better than the score of the positive class.[15] This relation between the two classes is typical of learning-based polarity classifiers (Andreevskaia & Bergler, 2008). However, it should also be pointed out that the size of the gap is much smaller (usually not greater than 2%). Moreover, the size of the gap does not bear any relation to the gap in the original RB, i.e. though there is a considerable difference in size between the gaps of RB$_{Plain}$ and RB$_{Neg}$ (17.45% to 5.02%), the size of the gaps in the self-trained versions is fairly similar (e.g. for SelfTr$_{Uni+Bi}$ 3.55% and 2.19%).

We also experimented with a combination of bag of words and the knowledge encoded in the rule-based classifier, i.e. the two features: the number of positive and negative polar expressions within a data instance. The performance of this combination is worse than a classifier trained on bag of words. The correlation between the two class labels and the two polarity features is disproportionately high since the polarity features essentially

[15] The only exception where the reverse is always true is SelfTr$_{MPQA}$. This does not come as a surprise since this feature set resembles RB most.

Table 6.10.: Performance of self-trained classifiers with different feature sets (experiments are carried out on a balanced corpus and results are averaged over all domains).

	$\mathbf{RB}_{Plain}$			$\mathbf{RB}_{bWSD}$			$\mathbf{RB}_{Neg}$			$\mathbf{RB}_{Weight}$		
Type	**F+**	**F−**	**Acc.**	**F+**	**F−**	**Acc.**	**F+**	**F−**	**Acc.**	**F+**	**F−**	**Acc.**
RB (*Baseline*)	69.81	52.36	63.12	70.39	61.79	66.65	72.42	67.40	70.14	74.26	68.30	71.60
$\mathbf{SelfTr}_{Top2000}$	70.15	70.88	70.53*	70.26	71.55	70.92*	72.78	73.88	73.40	74.79	74.18	75.73*
$\mathbf{SelfTr}_{Adj600}$	68.94	69.92	69.44*	70.08	71.41	70.76*	72.46	73.90	73.20*	74.34	75.82	75.10*
$\mathbf{SelfTr}_{MPQA}$	69.18	67.85	68.55*	70.03	69.46	69.75*	72.50	72.19	72.15	74.57	75.47	75.04*
$\mathbf{SelfTr}_{Uni}$	69.82	71.16	70.51*	70.53	72.41	71.50*	73.17	74.87	74.05*	75.73	77.67	76.74*
$\mathbf{SelfTr}_{Uni+Bi}$	71.14	74.69	71.94*†	71.41	73.64	72.57*†	74.39	76.12	75.29*†	76.43	78.62	77.58*†

*: Accuracy significantly better than RB on the basis of a paired t-test using $p < 0.05$

†: Accuracy significantly better than SelfTr_{Uni} on the basis of a paired t-test using $p < 0.05$

encode the prediction of the rule-based classifier. Consequently, the supervised classifiers develop a strong bias towards these two features and inappropriately downweight the bag-of-words features.

Table 6.11 compares rule-based classification and self-training on individual domains. In some domains self-training does not work. This is most evident in the *sports* domain using self-training on RB_{bWSD}. Apparently, the better the rule-based classifier is, the more likely a notable improvement by self-training can be obtained. Note that in the *sports* domain the self-trained classifier using the most complex rule-based classifier, i.e. $\text{SelfTr}(\text{RB}_{Weight})$, achieves the largest improvement compared to the rule-based classifier. These observations are also representative for the remaining feature sets examined but not displayed in Table 6.11.

Self-Training using Rule-Based Classifiers Compared to Semi-Supervised Learning

In the following experiments, we use Spectral Graph Transduction (SGT) (Joachims, 2003) as a semi-supervised learning classifier, since it provided best performance in previ-

Table 6.11.: Comparison of Accuracy between different rule-based classifiers (RB) and self-trained classifiers (SelfTr) trained on best feature set (Uni+Bi) on different domains (for each domain, performance is evaluated on a balanced corpus).

	$\mathbf{RB}_{Plain}$		$\mathbf{RB}_{bWSD}$		$\mathbf{RB}_{Neg}$		$\mathbf{RB}_{Weight}$	
Domain	**RB**	**SelfTr**	**RB**	**SelfTr**	**RB**	**SelfTr**	**RB**	**SelfTr**
computer	64.11	80.22	70.61	81.72	73.56	83.67*	74.28	83.50*
products	60.78	70.78	66.06	73.89*	71.06	77.00*†	70.94	77.00*†
sports	64.33	66.44	64.39	64.94	67.50	68.89†	68.89	72.78*†‡
travel	64.61	69.56	67.39	69.83	70.72	73.33*†	72.61	76.89*†‡
movies	61.75	72.70	64.80	72.45	67.85	73.55	71.30	77.75*†‡
average	63.12	71.94	66.65	72.57	70.14	75.29*†	71.60	77.58*†‡

*: significantly better than SelfTr bootstrapped on RB_{Plain}, †: significantly better than SelfTr bootstrapped on RB_{bWSD}, ‡: significantly better than SelfTr bootstrapped on RB_{Neg}; statistical significance is based on a χ^2 test using $p < 0.05$

ous experiments on semi-supervised learning (see Section 6.5). For each configuration (i.e. training and test partition) we randomly sample 20 partitions from the corpus. Labeled training and test data are always mutually exclusive but the test data (500 positive and 500 negative instances) can be identical to the unlabeled training data.

Figure 6.5 compares self-training bootstrapped on the output of rule-based classification (SelfTr) to supervised learning (SL) and semi-supervised learning (SSL). We compare two variations of SelfTr. SelfTr-A, like SSL, uses 1,000 randomly sampled data instances for both training and testing. (Again, we report the averaged result over 20 samples.) SelfTr-B (like in previous sections) selects 1,000 training instances by confidence from the entire dataset. The test data are, however, the same as in SelfTr-A. Unlike our previous experiments on SSL in which Top2000 was predominantly used for SL, we chose Uni+Bi as a feature set. It produces better results than Top2000 on classifiers trained

on larger training sets (i.e. ≥ 400).[16] For SSL, we consider Uni+Bi and Adj600, which is the feature set with the overall best performance using that learning method. For SelfTr, we consider the best classifier, i.e. SelfTr_{Uni+Bi}.

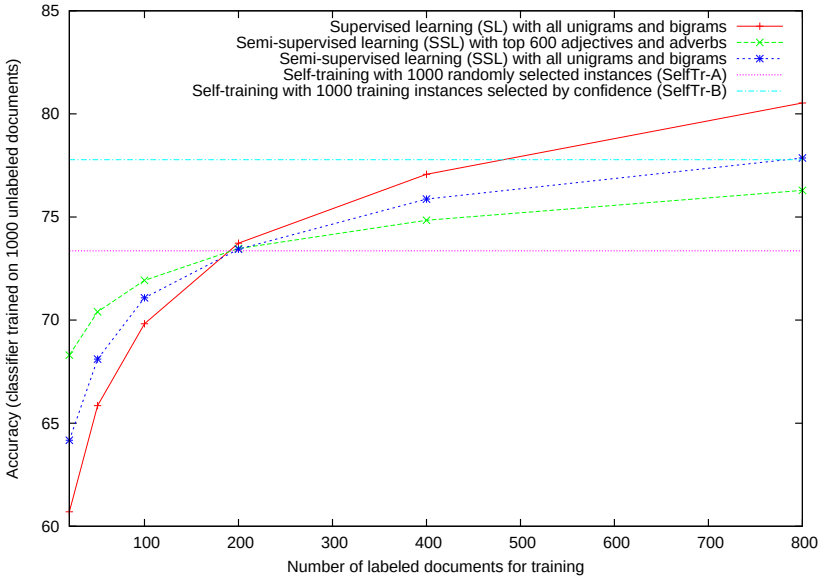


Figure 6.5.: Comparison of self-training and semi-supervised learning (performance is evaluated on balanced corpus and results are averaged over all domains).

Though SSL gives a notable improvement on small labeled training sets (i.e. ≤ 100), it produces much worse performance than SL on large training sets (i.e. ≥ 200). Adjectives and adverbs are a very reliable predictor. However, the size of the feature set is fairly small. Too little structure can be learned on large labeled training sets using such a small feature set. Using larger (but also noisier) feature sets for SSL, such as Uni+Bi, improves performance on larger labeled training sets. However, even with Uni+Bi SSL does not reach a performance comparable to SL on large training sets and it is significantly worse than Adj600 on small training sets.

Whenever SSL outperforms SL, every variation of SelfTr also outperforms SSL. SelfTr-B is significantly better than SelfTr-A which means that the quality of labeled instances

[16] Note that our previous experiments in SSL focused on small training sets.

matters and SelfTr is able to select more meaningful data instances than are provided by random sampling. Unfortunately, SSL-methods, such as SGT, do not incorporate such a selection procedure for the unlabeled data. Further exploratory experiments using the *entire* dataset as unlabeled data for SSL produced, on average, results similar to those using $1,000$ instances. This proves that SSL cannot internally identify as meaningful data as SelfTr-B does. Whereas SSL significantly outperforms SL on training sets using less than 200 training instances, the best variation of SelfTr, i.e. SelfTr-B, significantly outperforms SL on training sets using less than 400 instances. This difference is, in particular, remarkable since SelfTr does not use any manually labeled training data at all whereas SSL does.

Natural Class Imbalance and Mixed Reviews

In this section, we want to investigate what impact natural class imbalance has on self-training. While in both SL and SSL class imbalance should be a minor problem[17] since a class distribution can be estimated from the labeled training set (and, hopefully, the estimate is similar to the distribution on the test set), there is no prior information regarding the class distribution in self-training. This aspect has only been marginally covered in previous work (Qiu et al., 2009; Tan et al., 2008). In those works, different class ratios on the test set are evaluated. However, the same amount of positive and negative reviews is always selected for training. We assume that the optimal performance of self-training can be achieved when the class distribution of training and test set is identical and we will provide evidence for that. Moreover, we want to explore what impact different distributions between the two sets have on the Accuracy of the classifier and how different class-ratio estimation methods perform.

Previous work dealing with bootstrapping polarity classifiers using unlabeled data also focuses on datasets exclusively consisting of definite positive and negative reviews (Dasgupta & Ng, 2009; Qiu et al., 2009; Tan et al., 2008). In this section, the unlabeled dataset will also include mixed reviews, i.e. 3 star reviews (see Section 6.4). This review

[17] This is why, as far as text classification is concerned, we address class imbalance only in this section.

category is part of every realistic review collection and therefore should be taken into consideration for self-training. Unfortunately, the way that we formulate SSL for polarity classification does not allow us to also include these unlabeled 3 star reviews. Due to the unavailability of such data the experiments have only been carried out on the *Rate-It-All* data. We also add the constraint that the test data must be disjoint from the unlabeled training data.[18]

Test data are exclusively (definite) positive reviews (i.e. 4 & 5 star reviews) and (definite) negative reviews (i.e. 1 & 2 star reviews). From each domain, we randomly sample 200 data instances 10 times. We state the results averaged over these different test sets. The class ratio on each test set corresponds to the distribution of definite polar reviews, i.e. 3 star reviews are ignored. The distribution has been presented on Table 6.1 on page 94.

The unlabeled training dataset is the dataset of a domain excluding the test data. As labeled training data for the embedded supervised classifier within self-training, we use 70% of data instances labeled by the rule-based classifier ranked by confidence of prediction (across all domains and configurations, this size provided best results). Hopefully, most mixed reviews should be among the remaining 30%.

In the first experiment, we just focus on class imbalance (i.e. 3 star reviews are excluded). We will examine a self-trained classifier using the class-ratio estimate of a rule-based classifier as it is the most obvious estimate since the rule-based classifier is also used for generating the labeled training data. In particular, we want to explore whether there is a systematic relationship between the class distribution, the class-ratio estimate of the rule-based classifier and the resulting self-trained classifier. Table 6.12 lists the actual distribution of classes on the test set, the deviation between the distribution as it is predicted by the rule-based classifier and the actual distribution along the information towards which class the rule-based classifier is biased. Finally, we also list the absolute improvement/deterioration of the self-trained classifier in comparison to the

[18] We can include this restriction in this section since we will not consider the semi-supervised learning algorithm SGT in this section.

rule-based classifier. We will only consider the best rule-based classifier, i.e. RB_{Weight}, and for self-training, we will exclusively consider the best configuration from the previous experiments, i.e. SelfTr_{Uni+Bi}. The table shows that the quality of class-ratio estimates of rule-based classifiers varies among the different domains. The deviation is greatest on the *computer* domain. This is also the only domain in which the majority class are the negative reviews. With exception of the *sports* domain, the rule-based classifier always overestimates the amount of positive reviews. This overestimation is surprising considering that the polarity lexicon we use contains almost twice as many negative as positive polar expressions. This finding, however, is consistent with our earlier observation that rule-based classifiers have a bias towards positive reviews, i.e. they achieve a better F-Measure for positive reviews than for negative reviews.[19] Table 6.12 also clearly shows that the deviation negatively correlates with the improvement of the self-trained classifier towards the rule-based classifier. The improvement is greatest on the *sports* domain where the deviation is smallest and the greatest deterioration is obtained on the *computer* domain where the deviation is largest. In summary of this experiment, the class distribution of the data has a significant impact on the final self-trained classifier. In case there is a heavy mismatch between actual and predicted class ratio, the self-training approach will not improve the rule-based classifier.

In the following experiment we will compare how alternative class-ratio estimates relate to each other when applied to self-training. We compare the actual (oracle) distribution (Ratio-Or) with the balanced class ratio (Ratio-Bal), the class ratio as predicted by the rule-based classifier over the entire dataset (Ratio-RB) and estimates gained by a small amount of randomly sampled data instances from the dataset. We randomly sample 20 (Ratio-20), 50 (Ratio-50), and 100 (Ratio-100) instances. For each configuration (i.e. 20, 50, and 100), we sample 10 times, run SelfTr for each sample and report the averaged result. We compare the self-trained classifier with a classifier always assigning a test instance to the majority class (Majority-Cl) and the rule-based classifier (RB_{Weight}).

[19] We also observed that this bias is significantly larger on the simplest classifiers, i.e. RB_{Plain}, which is plausible since on this classifier the gap between F-Measures of positive and negative reviews is also largest (see Table 6.10).

Table 6.12.: Class imbalance and its impact on self-training.

Domain	Class Distribution (+ : −)	Deviation of Predicted Distribution from Actual Distribution	Class Towards which Predicted Distribution is Biased	Difference in Accuracy between RB and SelfTr(RB)
computer	43.17 : 56.83	16.30	+	−3.60
products	63.07 : 36.93	6.65	+	−0.25
sports	78.68 : 21.32	2.10	−	+3.15
travel	74.07 : 25.93	3.71	+	+1.30

Table 6.13.: Accuracy of different classifiers tested on naturally imbalanced data: for self-trained classifiers the unlabeled data also contain 3 star reviews; numbers in brackets state the results on a dataset which excludes 3 star reviews.

Classifier		computer	products	sports	travel	average
Majority-Cl		56.83	63.07	78.68	74.07	68.17
RB$_{Weight}$		73.80	76.00	77.35	79.50	76.66
SelfTr	**Ratio-Or**	82.80 (83.35)	80.90 (81.70)	81.25 (81.10)	81.70 (81.60)	81.66 (81.94)
	Ratio-Bal	**83.25 (82.95)**	75.40 (76.05)	62.55 (60.30)	66.95 (66.10)	72.04 (71.35)
	Ratio-RB	75.95 (70.20)	77.50 (75.75	**80.75 (80.50)**	**81.15 (80.80)**	78.84 (76.81)
	Ratio-20	77.36 (77.95)	77.61 (78.10)	79.10 (79.01)	78.94 (79.44)	78.01 (77.91)
	Ratio-50	80.43 (80.91)	80.45 (80.86)	79.94 (79.94)	80.64 (80.52)	80.37 (80.56)
	Ratio-100	80.96 (81.47)	**80.69 (81.27)**	80.62 (80.50)	80.76 (80.58)	**80.76 (80.96)**

This time, we also include the 3 star reviews in the unlabeled dataset.

Note that since Ratio-20, Ratio-50, and Ratio-100 are averaged results over 10 samples whereas the remaining classifiers are single results, we refrain from doing a statistical significance test as there is no commonly accepted way of comparing those different types of data (i.e. averaged results vs. single results).

Table 6.13 displays the results. We also display results of the datasets without using 3 star reviews in brackets. SelfTr using Ratio-Bal produces the worst results among the self-training classifiers. This is the only method used in previous work (in Chinese) (Qiu et al., 2009; Tan et al., 2008). Apparently, English data are more difficult than Chinese and, in English, SelfTr is more susceptible to deviating class-ratio estimates since in (Qiu et al., 2009; Tan et al., 2008) SelfTr with Ratio-Bal scores rather well. Ratio-Or produces best results which comes as no surprise since the class distribution in training and test set is the same. On average, Ratio-100 produces the second best result as it also gives fairly reliable class-ratio estimates (the deviation is 3.3% on average, whereas the deviation of Ratio-Bal is 18.16%). Both Ratio-50 and Ratio-100 produce results which are better than Majority-Cl and RB_{Weight}. As Ratio-Or, Ratio-Bal, Ratio-20, Ratio-50, and Ratio-100 suggest, the presence of mixed polar reviews does not produce different results. It is very striking, however, that the results of Ratio-RB are better using the 3 star reviews which seems counter-intuitive. We found that this is a corpus artifact. As already stated in Section 6.4, 3 star reviews do not only contain indefinite polar reviews but also positive and negative reviews. We also noted that Ratio-RB has a bias towards predicting too many positive instances. The bias is stronger if 3 star reviews are not included in the ratio-prediction (deviation of 8.5% instead of 6%). We, therefore, assume that among the 3 star reviews the proportion of negative-like reviews is greater than among the remaining part of the dataset and RB within SelfTr detects them as such. Thus, the bias towards positive polarity is slightly neutralized.

In summary of this experiment, using small samples of labeled data instances is the most effective way for class ratio estimation enabling SelfTr to consistently outperform Majority-CL and $Ratio_{Weight}$. Mixed reviews only have a marginal impact on the final

overall result of SelfTr.

6.6.4. Conclusion of Experiments on Bootstrapping Supervised Classifiers with Rule-Based Classification

In this section, we examined the effectiveness of bootstrapping a supervised polarity classifier with the output of an open-domain rule-based classifier. The resulting self-trained classifier is usually significantly better than the open-domain rule-based classifier since the supervised classifier exploits in-domain features. As far as the choice of the feature set is concerned, the supervised classifier within self-training behaves very much like an ordinary supervised classifier. The set of all unigrams and bigrams performs best.

The type of rule-based classifier has an impact on the performance of the final classifier. Usually, the more accurate the rule-based classifier is, the better the resulting self-trained classifier is. Therefore, modeling open-domain constructions relevant for polarity classification, such as negations or intensification, is important for this type of self-training. Thus, I have shown another aspect in sentiment analysis in which linguistic information is important to be considered.

In cases in which semi-supervised learning outperforms supervised learning, self-training at least also performs as well as the semi-supervised classifier. A great advantage of self-training is that it can choose instances to be added to the labeled training set by using confidence scores whereas in semi-supervised learning one has to resort to random sampling. The resulting data from self-training are usually much better.

Self-training also outperforms a rule-based classifier and a majority-class classifier in more difficult settings in which mixed reviews are part of the dataset and the class distribution is imbalanced, provided that the class-ratio estimate does not deviate too much from the actual ratio on the test set. A class-ratio estimate can be obtained by the output of the rule-based classifier but, on average, using small amounts of labeled samples from the data collection (i.e. approximately 50 instances) produces more reliable results.

Since this self-training method works under realistic settings, it is more robust than

semi-supervised learning, and its embedded supervised classifier only requires simple feature sets in order to produce reasonable results, it can be considered an effective method to overcome the need for large amounts of labeled in-domain training data for polarity classification.

6.7. Error Analysis

The improvements achieved by applying semi-supervised learning presented in this chapter are significantly smaller than they have been reported on other text classification tasks, such as conventional topic classification (Nigam, McCallum, Thrun, & Mitchell, 2000). Moreover, the performance gain on the movie domain (Section 6.5.2) is much larger than the average improvement on all domains (Section 6.5.2). We assume that the noticeable improvement obtained by semi-supervised learning on the movie domain is an exception. This improvement could only be achieved in combination with one particular polarity lexicon (i.e. AG). Unfortunately, we know only little as to how this manual lexicon has been built. Given our cross-domain evaluation, however, we have strong reason to believe that this lexicon was tuned for the movie domain. Therefore, we only need to answer why the general impact of semi-supervised learning on polarity classification is so low.

Similar to the dataset used for the detection of indefinite polarity (Chapter 4), the gold standard used for the experiments in this chapter may suffer from the fact that the labels for the data instances have been automatically generated, i.e. the ratings that have been assigned by the individual reviewers may not always be correct. However, we do not think that this is a general obstacle and the sole reason for the limited performance of semi-supervised learning. If our golden standard severely suffered from noise, then supervised learning and self-training should have been similarly affected. However, for both we have provided evidence in this chapter that is not the case. Therefore, we must assume that there is an inherent reason for the low performance of semi-supervised learning. One reason may be that topic information contained in the documents interferes with polarity information (as every document does not only possess some polarity but

addresses some specific topic). The fact that semi-supervised learning only provides a notable improvement over supervised learning when a feature set with a high proportion of polar expressions is used may support this assumption (as in those feature sets topic information is removed to a great extent). We do not think that it is possible to improve the performance of semi-supervised learning on polarity classification with a reasonable effort. If one confines the feature set to polar expressions, then some improvement towards supervised learning can be achieved, but only if very few labeled training data are considered. If there is a reasonable amount of labeled documents, e.g. 200 and more, then such a feature set provides too little expressiveness (usually at this point, supervised classifiers significantly outperform the semi-supervised classifier). If, however, a larger but less restricted feature set were considered, then the semi-supervised learner confuses topic information with polarity information.

Conceptually speaking, self-training offers a better alternative, since it incorporates both a predictive but also restrictive feature set (i.e. a polarity lexicon) and a more expressive but also noisier feature set (i.e. all unigrams and bigrams). Moreover, self-training encapsulates those different feature sets in two different classifiers (i.e. the former in a rule-based classifier and the latter in a supervised learner). The rule-based classifier has the advantage to restrict labels to data instances for which it makes a confident prediction. As a consequence, the unrestricted and more expressive feature set is used on labeled training data which have a higher quality than randomly selected labeled instances used in semi-supervised learning (see also Section 6.6.3). Semi-supervised learning cannot reach the level of performance of self-training as it does not possess this flexibility.

Self-training performs much better than semi-supervised learning but there is even room for improvement for this classifier. The rule-based classifier used for the experiments on self-training relies (as many other components/features of the classifiers presented in the previous chapters) on a robust recognition of polar expressions. Therefore, similar problems are encountered caused by limitations of currently available polarity lexicons. Yet these limitations are fairly difficult to overcome (see Chapter 3.6 for more details).

6.8. Conclusion

Polarity classification is a difficult text classification task and this becomes apparent if bootstrapping algorithms for this task are considered. In order for bootstrapping to become effective, one needs to make use of a fairly predictive source of information. For instance, semi-supervised learning depends on a predictive feature set, otherwise no improvement will be achieved. Surprisingly, adjectives and adverbs have the same effectiveness as polarity lexicons. In comparison to semi-supervised learning, a bootstrapping method using a rule-based classifier seems to be more promising, since in all settings we examined the latter either outperformed the former or was at least equally robust. There are three major advantages that we discovered. Firstly, self-training does not require any manually labeled training data at all. Secondly, the rule-based classifier can choose training samples by itself (using confidence scores) and thus can choose those instances which are most useful. Thirdly, our experiments suggest that improving the quality of rule-based classifiers also improves the quality of the bootstrapped classifier. Thus, this method leaves plenty of room for improvement as the most complex rule-based classifier we used in this chapter is still very crude compared to other compositional approaches, such as (Moilanen & Pulman, 2007) or (Klenner et al., 2009). The effectiveness of semi-supervised classifiers, however, is restricted to small labeled training sets and we could not find a potential direction for future work to improve them.

7. Convolution Kernels for Opinion Holder Extraction

7.1. Introduction

In this chapter, we leave the realm of text classification in sentiment analysis and turn to opinion holder extraction. Together with opinion target extraction, opinion holder extraction is one of the common entity extraction tasks in sentiment analysis. It is considered a critical component of several NLP applications, such as opinion question-answering (i.e. systems which automatically answer opinion questions, such as "What does [X] like about [Y]?"). Such systems need to be able to distinguish which entities in a candidate answer sentence are the sources of opinions (= opinion holder) and which are the targets.

In other NLP tasks, in particular, in relation extraction, there has been much work on *convolution kernels*, i.e. kernel functions exploiting huge amounts of features without an explicit feature representation. Previous research on that task has shown that convolution kernels, such as sequence or tree kernels, are quite competitive when compared to manual feature engineering (Moschitti, 2008; Bunescu & Mooney, 2005; Nguyen, Moschitti, & Riccardi, 2009). In order to effectively use convolution kernels, it is often necessary to choose appropriate substructures of a sentence rather than representing the sentence as a whole structure (Bunescu & Mooney, 2005; M. Zhang, Zhang, & Su, 2006). As for tree kernels, for example, one typically chooses the syntactic subtree immediately enclosing two entities potentially expressing a specific relation in a given sentence. The opinion holder detection task is different from this scenario. There can be *several* cues within a

sentence to indicate the presence of a genuine opinion holder and these cues need not be member of a particular word group, e.g. they can be opinion words (see Sentences (7.1)-(7.3)), communication words, such as *maintained* in Sentence (7.2), or other lexical cues, such as *according to* in Sentence (7.3).

(7.1) The U.S. commanders consider$_{opinion}$ the prisoners to be unlawful_combatants$_{opinion}$ as opposed to prisoners of war.

(7.2) During the summit, Koizumi maintained$_{communication}$ a clear-cut_collaborative-_stance$_{opinion}$ towards the U.S. and emphasized that the President was objective$_{opinion}$ and circumspect.

(7.3) According_to$_{cue}$ Fernandez, it was the worst_mistake$_{opinion}$ in the history of the Argentine economy.

Thus, the definition of boundaries of the structures for the convolution kernels is less straightforward in opinion holder extraction.

The aim of this chapter is to explore in how far convolution kernels can be beneficial for effective opinion holder detection. We are not only interested in how far different kernel types contribute to this extraction task but we also contrast the performance of these kernels with a manually designed feature set used as a standard vector kernel.

Moreover, we will show that in order to obtain a good performance the consideration of linguistic knowledge is essential for several aspects of a classifier based on convolution kernels being:

- the level of representation
- the scope for each convolution kernel
- the semantic categories that are used to generalize convolution kernels

The work presented in this chapter is also described in (Wiegand & Klakow, 2010b).

7.2. Related Work

Choi, Cardie, Riloff, and Patwardhan (2005) examine opinion holder extraction using CRFs with various manually defined linguistic features and patterns automatically learned by the AutoSlog system (Riloff, 1996). The linguistic features focus on named-entity information and syntactic relations to opinion words. In this chapter, we use very similar settings. The features presented in (S.-M. Kim & Hovy, 2005; Bloom, Stein, & Argamon, 2007) resemble very much (Choi et al., 2005). Bloom, Stein, and Argamon (2007) also consider communication words to be predictive cues for opinion holders.

S.-M. Kim and Hovy (2006) and Bethard et al. (2004) explore the usefulness of semantic roles provided by FrameNet (Fillmore, Johnson, & Petruck, 2003) for both opinion holder and opinion target extraction. Due to data-sparseness, S.-M. Kim and Hovy (2006) expand FrameNet data by using an unsupervised clustering algorithm.

(Choi et al., 2006) is an extension of (Choi et al., 2005) in that opinion holder extraction is learned jointly with opinion detection. This requires that opinion expressions and their relations to opinion holders are annotated in the training data. Semantic roles are also taken as a potential source of information. In our work, we deliberately work with minimal annotation and, thus, do not consider any labeled opinion expressions and relations to opinion holders in the training data. We exclusively rely on entities marked as opinion holders. In many practical situations, the annotation beyond opinion holder labeling is too expensive.

Complex convolution kernels have been successfully applied to various NLP tasks, such as relation extraction (Bunescu & Mooney, 2005; M. Zhang et al., 2006; Nguyen et al., 2009), question answering (D. Zhang & Lee, 2003; Moschitti, 2008), and semantic role labeling (Moschitti, Pighin, & Basili, 2008). In all these tasks, they offer competitive performance to manually designed feature sets. Bunescu and Mooney (2005) combine different sequence kernels encoding different contexts of candidate entities in a sentence. They argue that several kernels encoding different contexts are more effective than just using one kernel with one specific context. We build on that idea and compare various scopes eligible for opinion holder extraction. Moschitti (2008) and Nguyen et al. (2009)

suggest that different kinds of information, such as word sequences, part-of-speech tags, syntactic and semantic information should be contained in separate convolution kernels. We also adhere to this notion.

7.3. Data

As labeled data, we use the sentiment annotation of the *MPQA 2.0-corpus*[1]. Opinion holders are not explicitly labeled as such. However sources of *private states* and *subjective speech events* (Wiebe et al., 2003) are a fairly good approximation of the task. Previous works (Choi et al., 2005; S.-M. Kim & Hovy, 2005; Choi et al., 2006) use similar approximations. Please note, however, since we use a different version of the MPQA-corpus and a more restrictive but also more accurate definition[2], the numbers presented in this chapter cannot be directly compared with these publications. However, we tried to account for comparability by using similar features in our manual feature set (i.e. our baseline) as part of our manually designed feature set (see also Section 7.4.5).

Also note that in this work, we deliberately omit any opinion information from the annotation in the golden standard, since it is not only very difficult for human annotators to annotate but it is also difficult to recognize automatically.

7.4. Method

In this work, we consider all noun phrases (NPs) as possible candidate opinion holders. Therefore, the set of all data instances is the set of the NPs within the MPQA 2.0-corpus. Each NP is labeled as to whether it is a genuine opinion holder or not. Throughout this section, we will use Sentence (7.4) as an example.

(7.4) During the summit, Koizumi maintained$_{communication}$ a clear-cut_collaborative-

[1] www.cs.pitt.edu/mpqa/databaserelease

[2] For instance, e-mail correspondence with the first author of (Choi et al., 2005) confirmed that sources of private states and *all* speech events (rather than only subjective speech events) had been considered opinion holders.

Table 7.1.: The different levels of representation.

Type	**Description**	**Example**
WRD	sequence of words	During the summit , Koizumi$_{CAND}$ maintained a clear-cut collaborative stance ...
WRD_{GN}	sequence of generalized words	During the summit , CAND$_{PERSON}$ COMM OPINION ...
POS	part-of-speech sequence	IN DET NN PUNC CAND VBD DET JJ JJ NN ...
POS_{GN}	generalized part-of-speech sequence	IN DET NN PUNC CAND$_{PERSON}$ COMM OPINION ...
$CONST$	constituency tree	*see Figure 7.1(a)*
$CONST_{AUG}$	augmented constituency tree	*see Figure 7.1(b)*
$GRAM_{WRD}$	grammatical relation path labels with words	Koizumi$_{CAND}$ NSUBJ↑ maintained DOBJ↓ stance
$GRAM_{POS}$	grammatical relation path labels with part-of-speech tags	CAND NSUBJ↑ VBD DOBJ↓ NN
PAS	predicate argument structures	*see Figure 7.2(a)*
PAS_{AUG}	augmented predicate argument structures	*see Figure 7.2(b)*

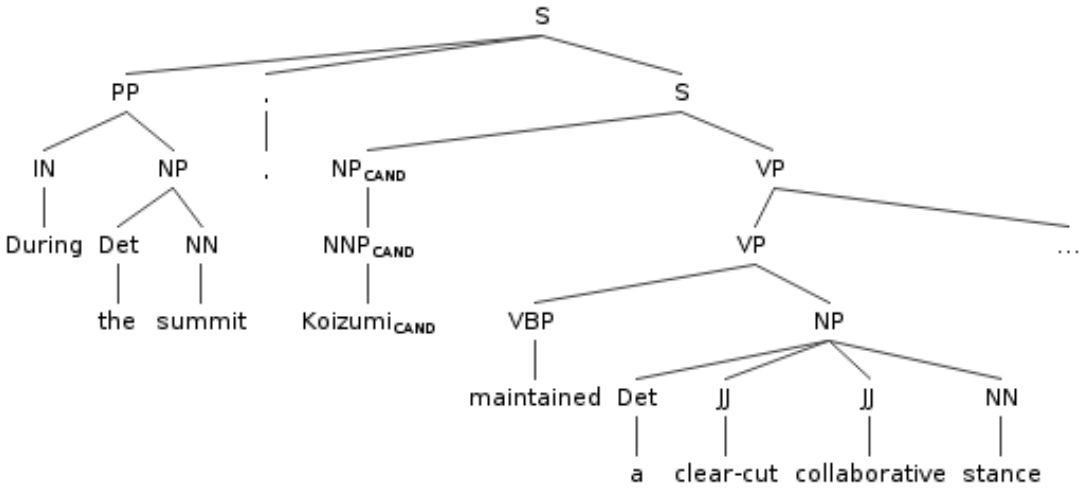
_stance$_{opinion}$ towards the U.S. and emphasized that the President was objective$_{opinion}$ and circumspect.

7.4.1. The Different Levels of Representation

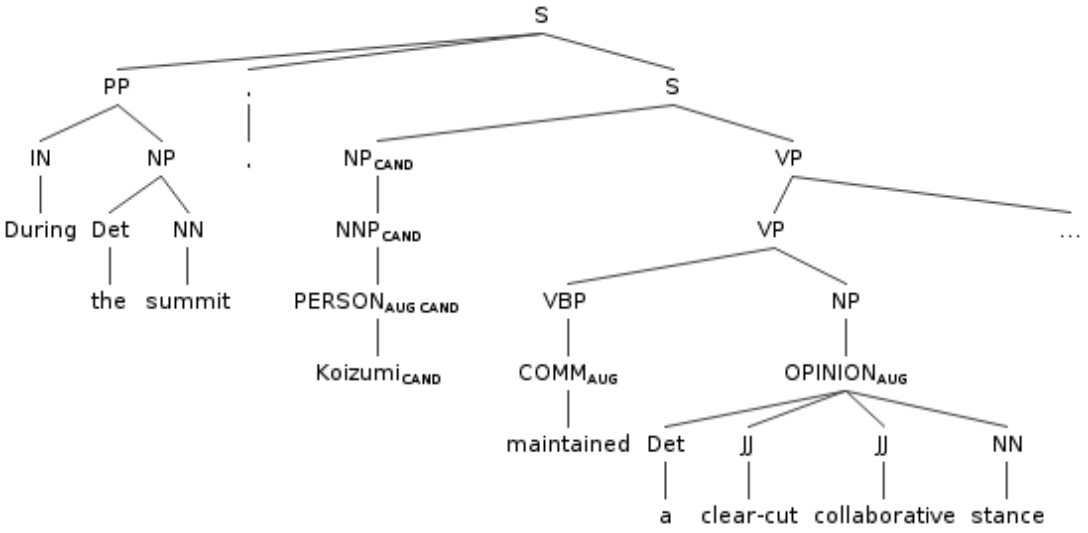
Several levels of representation are important for opinion holder extraction. We will briefly address every individual level that is going to be considered in this chapter. Table 7.1 lists all the different levels that are used in this work.

Words

As already pointed out in the introduction of this chapter, there are certain words which are indicative of a genuine opinion holder when occurring in the vicinity of the candidate.

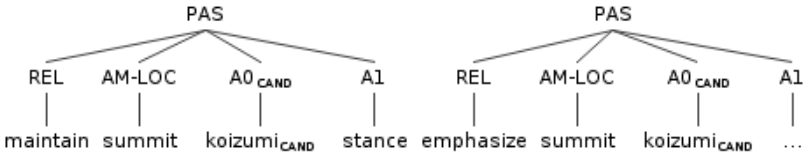


(a) plain

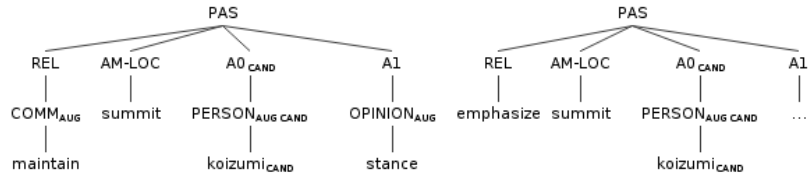


(b) augmented

Figure 7.1.: Constituency parse trees ($CONST$).



(a) plain



(b) augmented

Figure 7.2.: Predicate-argument structures (PAS).

Therefore, word sequences (WRD) are considered as a level of information. In addition to the plain word level, we also introduce another level in which generalization is employed (WRD_{GN}) where certain words or phrases are replaced by their corresponding semantic categories which are known to be predictive for opinion holder extraction (Choi et al., 2005; S.-M. Kim & Hovy, 2005; Choi et al., 2006; S.-M. Kim & Hovy, 2006; Bloom, Stein, & Argamon, 2007). The semantic categories that we consider are *named-entity tags*, an OPINION tag for *opinion words*, and a COMM tag for *communication words*. Additionally, all candidate tokens are reduced to one generic CAND token. By applying generalization we hope to account for data-sparseness.

Parts of Speech

The usage of part-of-speech sequences provides a more abstract level of representation. That is why we assume that it might be possible to recognize some predictive sequential patterns that are more general than the patterns on word level. Similar to the word level, we also add another level with generalized part-of-speech information (POS_{GN}) in which tags representing words or phrases belonging to semantic categories are replaced by semantic categories. We use the same categories as on word level.

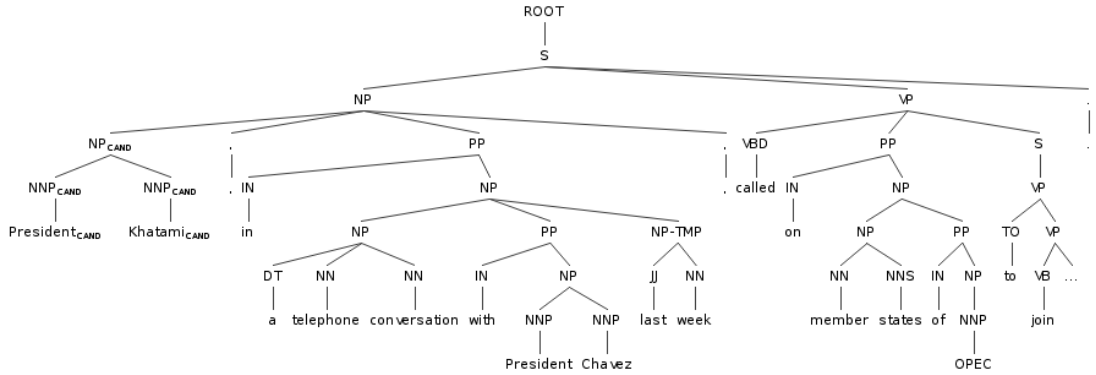


Figure 7.3.: Illustration of long-distance relationship between candidate opinion holder *President Khatami* and related cue *called*.

Constituency Parse Trees

Constituency parse trees ($CONST$) allow to capture some long-range relationships that cannot be captured by the previous levels of representation. For example in Figure 7.3, the opinion holder, i.e. *President Khatami*, is fairly wide apart from the cue that relates to it, i.e. *called* (communication word), as there are 11 intervening tokens.[3] However, the relation path from NP_{CAND} to the word consists of just 5 edges.

We also add another level of representation in which we augment constituency parse trees by the semantic categories ($CONST_{AUG}$) we also considered for WRD_{GN} and POS_{GN}. The additional nodes with these semantic categories are added in such a way that they directly dominate the pertaining words or phrases representing them.

Grammatical Relations from a Dependency Parse Tree

Like constituency parse trees, grammatical relations ($GRAM$) also allow the consideration of long-range dependencies, however, they abstract even more from surface structures. For instance, a grammatical relation, such as *subject-of*, abstracts from active and passive voice constructions, such as Sentences (7.5) and (7.6).

[3]Please note that the cue *conversation* (communication word) is nearer to the candidate but its presence is coincidental. It is not related to the candidate, as it is part of a parenthetical insertion.

(7.5) [The European Commission]$_{subject}$ has *critisized*$_{opinion}$ the Bush administration.

(7.6) The Bush admistration has been *critisized*$_{opinion}$ by [the European Commission]$_{subject}$.

In addition to plain grammatical relations we also have a further level, $GRAM_{POS}$, in which words are replaced by part-of-speech tags in order to capture some more general path sequences.

Note that the grammatical relation paths, i.e. $GRAM_{WRD}$ and $GRAM_{POS}$, can only be applied in case there is another expression in the focus in addition to the candidate opinion holder of the data instance itself, e.g. the nearest opinion expression to the candidate. Section 7.4.4 explains in detail how this is done.

Predicate Argument Structures

The most abstract level of representation are predicate argument structures (PAS). For this level, we use the PropBank annotation scheme (Kingsbury & Palmer, 2002). Unlike $CONST$, PAS just focuses on entities being arguments of a predicate. So, the resulting structures in PAS are flatter than those structures provided by dependency parse trees (which ideally encode relations among all words in a sentence).

In addition to that, the labels assigned to arguments also abstract from overt syntactic variation as $GRAM$ does. However, the labels generalize even across different parts-of-speech. For instance, in Sentence (7.7) the opinion holder is the subject of the verbal predicate *agreed* and is assigned the semantic role of an agent. The agent in the PropBank taxonomy corresponds to *A0*. In Sentence (7.8), the opinion holder is not the subject of the nominalization but its modifier. It is, however, still the agent. Grammatical relations are ambiguous in contrast to semantic roles as Sentence (7.9) shows. In that sentence there is no opinion holder but the grammatical relations are identical to Sentence (7.8). The semantic difference is only reflected by the semantic role assigned to *Kyoto* which is not an agent.

(7.7) The U.S.$^{subject}_{A0}$ *has agreed*$_{PRED(V)}$ to the resolution.

(7.8) The $\underline{\text{U.S.}}_{A0}^{modifier}$ $agreement_{PRED(N)}$ to take missiles out of Turkey [...] *(The U.S. agreed to do something).*

(7.9) The $\text{Kyoto}_{AM-LOC}^{modifier}$ $agreement_{PRED(N)}$ is an international agreement linked to the United Nations. (*Kyoto is the place where the agreement was made.*)

Similar to constituency parse trees, we also add another level of representation in which augmentation is employed (PAS_{AUG}).

7.4.2. Support Vector Machines and Kernel Methods

Support Vector Machines (SVMs) are one of the most robust supervised machine learning techniques in which training data instances $\vec{x}$ are separated by a hyperplane $H(\vec{x}) = \vec{w} \cdot \vec{x} + b = 0$ where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. One advantage of SVMs is that kernel methods can be applied which map the data to other feature spaces in which they can be separated more easily. Given a feature function $\phi : \mathbb{O} \rightarrow \mathbb{R}$, where $\mathbb{O}$ is the set of the objects, the kernel trick allows the decision hyperplane to be rewritten as: $H(\vec{x}) = \left(\sum_{i=1...l} y_i \alpha_i \vec{x}_i \right) \cdot \vec{x} + b = \sum_{i=1...l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b = \sum_{i=1...l} y_i \alpha_i \phi(o_i) \cdot \phi(o) + b$

where y_i is equal to 1 for positive and -1 for negative examples, $\alpha_i \in \mathbb{R}$ with $\alpha_i \geq 0, o_i \forall_i \in \{1, \ldots, l\}$ are the training instances and the product $K(o_i, o) = \langle \phi(o_i) \cdot \phi(o) \rangle$ is the kernel function associated with the mapping ϕ.

7.4.3. Sequence and Tree Kernels

A sequence kernel (SK) measures the similarity of two sequences by counting the number of common subsequences. We use the kernel by Taylor and Christianini (2004) which has the advantage that it also considers subsequences of the original sequence with some elements missing. The extent of these *gaps* in a sequence is suitably reflected by a weighting function incorporated into the kernel.

Tree kernels (TKs) represent trees by their substructures. The feature space of these substructures, or fragments, is mapped onto a vector space. The kernel function computes the similarity of pairs of trees by counting the number of common fragments. In this

work, we evaluate two tree kernels: Subset Tree Kernel (STK) (Collins & Duffy, 2002) and Partial Tree Kernel (PTK_{basic}) (Moschitti, 2006a).

In STK, a tree fragment can be any set of nodes and edges of the original tree provided that every node has either all or none of its children. This constraint makes that kind of kernel well-suited for constituency trees which have been generated by context free grammars since the constraint corresponds to the restriction that no grammatical rule must be broken. For example, STK enforces that a subtree, such as *[VP [VBZ, NP]]*, cannot be matched with *[VP [VBZ]]* since the latter *VP* node only possesses one of the children of the former.

PTK_{basic} is more flexible since the constraint of STK on nodes is relaxed. This makes this type of tree kernel less suitable for constituency trees. We, therefore, apply it only to trees representing predicate-argument structures (PAS) (see Figure 7.2). Note that a data instance is represented by a set of those structures (i.e. all predicate-argument structures of a sentence in which the head of the candidate opinion holder occurs) rather than a single structure. Thus, the actual partial tree kernel function we use for this task, PTK, sums over all possible pairs PAS_l and PAS_m of two data instances x_i and x_j: $PTK(x_i, x_j) = \sum_{PAS_l \in x_i} \sum_{PAS_m \in x_j} PTK_{basic}(PAS_l, PAS_m)$.

To summarize, Table 7.2 lists the different kernel types we use coupled with the appropriate levels of representation. This choice of pairing has already been motivated and empirically proven suitable on other tasks (Moschitti, 2008; Nguyen et al., 2009).

Table 7.2.: The different types of kernels.

Type	**Description**	**Levels of Representation**
SK	Sequential Kernel	$WRD_{(GN)}$, $POS_{(GN)}$, $GRAM_{WRD}$, $GRAM_{POS}$
STK	Subset Tree Kernel	$CONST_{(AUG)}$
PTK	Partial Tree Kernel	PAS
VK	Vector Kernel	*not restricted*

7.4.4. The Different Scopes

We argue that using the entire word sequence or syntax tree of the sentence in which a candidate opinion holder is situated to represent a data instance produces too large structures for a convolution kernel. Since a classifier based on convolution kernels has to derive meaningful features by itself, the larger these structures are the more likely noise is included in the model. Previous work in relation extraction has also shown that the usage of more focused substructures, e.g. the smallest subtree containing the two candidate entities of a relation, is more effective (M. Zhang et al., 2006). Unfortunately, in our task there is only one explicit entity we know of for each data instance which is the candidate opinion holder. However, there are several indicative cues within the context of the candidate which might be considered important. We identify three different cues being the nearest *predicate*, i.e. full verb or nominalization, *opinion word*, and *communication word*.[4] For each of these expressions, we define a scope where the boundaries are the candidate opinion holder and the pertaining cue. Given these scopes, we can define resulting subsequences/subtrees and combine them.

We further add two *background scopes*, one being the semantic scope of the candidate opinion holder and the entire sentence. As semantic scope we consider the subclause in which a candidate opinion holder is situated. The subclause should contain most relevant relationships between candidate opinion holder and other linguistic entities while being considerably smaller than the entire sentence at the same time. Typically, the subtree representing a subclause has the closest S node dominating the candidate opinion holder as the root node and it contains only those nodes from the original sentence parse which are also dominated by that S node and whose path to that node does not contain another S node.

Figure 7.4 illustrates the different scopes. Abbreviations are explained in Table 7.3. As already mentioned in Section 7.4.1 for grammatical relation paths, a second expression in addition to the candidate opinion holder is required. These expressions can be derived from the different scopes, i.e. for $PRED$ it is the nearest predicate to the candidate, for

[4] These three expressions may coincide but do not have to.

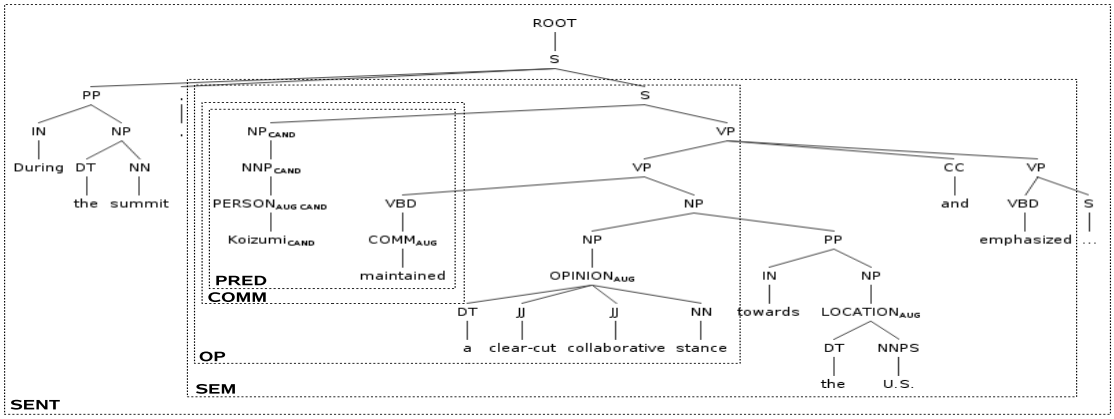


Figure 7.4.: Illustration of the different scopes on a $CONST_{AUG}$; nodes belonging to the candidate opinion holder are marked with $_{CAND}$.

OP it is the nearest opinion word, and for $COMM$ it is the nearest communication word. For the background scopes SEM and $SENT$, however, there is no second expression in focus. Therefore, grammatical relation paths cannot be defined for these scopes.

Table 7.3.: The different types of scope.

Type	**Description**
$PRED$	scope with the boundaries being the candidate opinion holder and the nearest predicate
OP	scope with the boundaries being the candidate opinion holder and nearest opinion word
$COMM$	scope with the boundaries being the candidate opinion holder and the nearest communication word
SEM	semantic scope of the candidate opinion holder, i.e. subclause containing the candidate
$SENT$	entire sentence in which in the opinion holder occurs

7.4.5. Manually Designed Feature Set for a Standard Vector Kernel

In addition to the different types of convolution kernels, we also define an explicit feature set for a vector kernel (VK). Many of these features mainly describe properties of the

relation between the candidate and the nearest predicate[5] since in our initial experiments the nearest predicate has always been the strongest cue. Adding these types of features for other cues, e.g. the nearest opinion or communication word, only resulted in a decrease in performance. Table 7.4 lists all the features we use. Note that this manual feature set employs all those sources of information which are also exploited by the convolution kernels. Some of the information contained in the convolution kernels can, however, only be represented in a more simplified fashion when using a manual feature set. For example, the first PAS in Figure 7.2(a) is converted to just the pair of predicate and argument representing the candidate (i.e. *REL:maintain_ A0:Koizumi*). The entire PAS is not used since it would create too sparse features. Convolution kernels, on the other hand, can cope with those complex structures as input since they internally match substructures. Manual features are less flexible since they do not account for partial matches.

Table 7.4.: Manually designed feature set.

headword/governing category of CAND
is CAND capitalized/a person?
is CAND *subj/dobj/iobj/pobj* of OPINION/COMM?
is CAND preceded by *according to*? (Choi et al., 2005)
does CAND contain possessive and is followed by OPINION/COMM? (Choi et al., 2005)
is CAND preceded by *by* which is attached to OPINION/COMM? (Choi et al., 2005)
predicate-argument pairs in which CAND occurs
lemma/part-of-speech tag/subcategorization frame/voice of nearest predicate
is nearest predicate OPINION/COMM?
does CAND precede/follow nearest predicate?
words between nearest predicate and CAND (bag of words)
part-of-speech sequence between nearest predicate and CAND
constituency path/grammatical relation path from predicate to CAND

[5]We select the nearest predicate by using the syntactic parse tree. Thus, we hope to select the predicate which syntactically relates to the candidate opinion holder.

7.5. Experiments

We used 400 documents of the MPQA-corpus for five-fold cross-validation and 133 documents as a development set. We report statistical significance on the basis of a paired t-test using 0.05 as the significance level. All experiments were done with the *SVM-Light-TK* toolkit[6]. The results are reported using Accuracy, Precision, Recall, and F-Measure as evaluation measures (see also Appendix A.1). We evaluated on the basis of exact phrase matching. We set the trade-off parameter $j = 5$ for all feature sets. For the manual feature set we used a polynomial kernel of third degree which resulted in better performance than a linear kernel. These two critical parameters were tuned on the development set. As far as the sequence and tree kernels are concerned, we used the parameter settings from (Moschitti, 2008), i.e. $\lambda = 0.4$ and $\mu = 0.4$. Kernels were combined using plain summation. The documents were parsed using the Stanford Parser (Klein & Manning, 2003). Named-entity information was obtained by the Stanford tagger (Finkel, Grenager, & Manning, 2005). Semantic roles were obtained by using the parser by Y. Zhang, Wang, and Uszkoreit (2008). Opinion expressions were identified using the Subjectivity Lexicon from the MPQA-project (Wilson et al., 2005). Communication words were obtained by using the Appraisal Lexicon (Bloom, Stein, & Argamon, 2007). Nominalizations were recognized by looking up nouns in NOMLEX (Macleod, Grishman, Meyers, Barrett, & Reeves, 1998).

7.5.1. Notation

Each kernel is represented as a triple:

$\langle$**levelOfRepresentation** (Table 7.1), **scope** (Table 7.3), **typeOfKernel** (Table 7.2)$\rangle$

For example, $\langle CONST, SENT, STK \rangle$ is a Subset Tree Kernel of a constituency parse having the scope of the entire sentence. Note that not all combinations of these three parameters are meaningful.

[6] available at `disi.unitn.it/moschitti`

Table 7.5.: Result of the vector kernel (VK).

Acc.	**Prec.**	**Rec.**	**F.**
93.63	53.28	59.37	56.16

In the following, we will just focus on important and effective combinations. The kernel composed of manually designed features is denoted by just VK. The kernel composed of predicate-argument structures is denoted by $\langle PAS, SENT, PTK \rangle$.

7.5.2. Vector Kernel (VK)

Table 7.5 displays the result of the vector kernel using a manually designed feature set. It should be interpreted as a baseline. Due to the high class imbalance we will focus on the comparison of F-Measure throughout this chapter rather than Accuracy which is fairly biased on this dataset. The F-Measure of this classifier is at 56.16%.

7.5.3. Sequence Kernels (SKs)

For both sequence and tree kernels we need to find out what the best scope is, whether it is worthwhile to combine different scopes, and what different layers of representation can be usefully combined.

The upper part of Table 7.6 lists the results of simple word kernels using the different scopes. The performance of the kernels using individual scopes varies greatly. The best scope is $PRED$ (1), the second best is SEM (2). The good performance of $PRED$ does not come as a surprise since the sequence is the smallest among the different scopes, so this scope is least affected by data sparseness. Moreover, this result is consistent with our initial experiments on the manual feature set (see Section 7.4.5).

Using different combinations of the word sequence kernels shows that $PRED$ and SEM (6) are a good combination, whereas OP, $COMM$, and $SENT$ (7;8;9) do not positively contribute to the overall performance which is consistent which the individual scope evaluation. Apparently, these scopes capture less linguistically relevant structure.

Table 7.6.: Results of the different sequence kernels.

ID	Kernel	Acc.	Prec.	Rec.	F.
1	$\langle WRD, PRED, SK\rangle$	93.25	51.08	42.29	46.26
2	$\langle WRD, OP, SK\rangle$	92.77	46.38	32.52	38.21
3	$\langle WRD, COMM, SK\rangle$	92.42	43.70	35.99	39.46
4	$\langle WRD, SEM, SK\rangle$	93.16	50.32	34.65	41.04
5	$\langle WRD, SENT, SK\rangle$	90.60	29.90	27.29	28.53
6	$\langle WRD, PRED, SK\rangle + \langle WRD, SEM, SK\rangle$	93.78	56.55	41.36	47.77
7	$\sum_{j\in\{PRED,OP,COMM\}} \langle WRD, j, SK\rangle$	93.55	54.26	39.50	45.71
8	$\sum_{j\in Scopes\setminus SENT} \langle WRD, j, SK\rangle$	93.82	57.21	40.28	47.26
9	$\sum_{j\in Scopes} \langle WRD, j, SK\rangle$	93.63	55.15	39.52	46.03
10	$\langle WRD, PRED, SK\rangle + \langle POS, PRED, SK\rangle$	93.03	49.39	53.53	51.37
11	$\sum_{i\in\{PRED,SEM\}} (\langle WRD, i, SK\rangle + \langle POS, i, SK\rangle)$	93.86	55.60	53.22	54.38
12	$\sum_{i\in\{PRED,SEM\}} \langle WRD, i, SK\rangle + \langle GRAM_{WRD}, PRED, SK\rangle$	94.01	58.19	45.88	51.29
13	$\sum_{i\in\{PRED,SEM\}} \langle WRD, i, SK\rangle + \sum_{j\in\{PRED,OP,COMM\}} \langle GRAM_{WRD}, j, SK\rangle$	93.83	56.28	45.64	50.40
14	$\sum_{i\in\{PRED,SEM\}} \langle WRD, i, SK\rangle + \langle GRAM_{WRD}, PRED, SK\rangle + \langle GRAM_{POS}, PRED, SK\rangle$	93.98	56.59	53.92	55.21
15	$\sum_{i\in\{PRED,SEM\}} (\langle WRD, i, SK\rangle + \langle WRD_{GN}, i, SK\rangle)$	93.97	57.08	49.46	53.00
16	$\sum_{i\in\{PRED,SEM\}} (\langle WRD, i, SK\rangle + \langle POS_{GN}, i, SK\rangle)$	93.97	56.60	52.42	54.42
17	$\sum_{i\in\{PRED,SEM\}} (\langle WRD, i, SK\rangle + \langle WRD_{GN}, i, SK\rangle + \langle POS, i, SK\rangle + \langle POS_{GN}, i, SK\rangle)$	93.85	55.16	57.00	56.06
18	$\sum_{i\in\{PRED,SEM\}} (\langle WRD, i, SK\rangle + \langle WRD_{GN}, i, SK\rangle + \langle POS, i, SK\rangle + \langle POS_{GN}, i, SK\rangle) + \langle GRAM_{WRD}, PRED, SK\rangle + \langle GRAM_{POS}, PRED, SK\rangle$	**94.21**	**57.64**	**59.81**	**58.70**

The next part of Table 7.6 shows the contribution of POS kernels when added to WRD kernels. Adding the corresponding POS kernel to the WRD kernel with $PRED$ scope (10) results in an improvement by more than 5% in F-Measure. We get another improvement by approximately 3% when the corresponding SEM kernels (11) are added. This suggests that POS is an effective generalization and that the two scopes $PRED$ and SEM are complementary.

For the $GRAM_{WRD}$ kernel, the $PRED$ scope (12) is again most effective. We assume that this kernel most likely expresses meaningful syntactic relationships for our task. Adding the $GRAM_{POS}$ kernel (14) gives another boost by almost 4%.

Generalized sequence kernels are important. Adding the corresponding WRD_{GN} kernels to the WRD kernel with $PRED$ and SEM scope results in an improvement from 47.77% (1) to 53.00% (15) which is a bit less than the combination of WRD and $POS_{(GN)}$ kernels (16). However, these types of kernels seem to be complementary since their combination provides an F-Measure of 56.06% (17). This kernel combination already performs on a par with the manually designed vector kernel though less information is taken into consideration.

Finally, the best combination of sequence kernels (18) comprises WRD, WRD_{GN}, POS, and POS_{GN} kernels with $PRED$ and SEM scope combined with a $GRAM_{WRD}$ and a $GRAM_{POS}$ kernel with $PRED$ scope. The performance of 58.70% significantly outperforms the vector kernel.

7.5.4. Tree Kernels (TKs)

Table 7.7 shows the results of the different tree kernels. The table is divided into two halves. The left half (A) are plain tree kernels, whereas the right half (B) are the augmented tree kernels. As far as $CONST$ kernels are concerned, there is a systematic improvement by approximately 2% using tree augmentation. This proves that further non-syntactic knowledge added to the tree itself results in an improved F-Measure. However, tree augmentation does not have any impact on the PAS kernels.

The overall performance of the tree kernels shows that they are much more expres-

Table 7.7.: Results of the different tree kernels.

		A $i = CONST, j = PAS$				**B** $i = CONST_{AUG}, j = PAS_{AUG}$			
ID	**Kernel**	**Acc.**	**Prec.**	**Rec.**	**F.**	**Acc.**	**Prec.**	**Rec.**	**F.**
19	$\langle i, PRED, STK \rangle$	92.89	48.68	62.34	54.67	93.12	49.99	65.04	56.52
20	$\langle i, OP, STK \rangle$	93.04	49.49	54.71	51.96	93.27	50.93	59.06	54.68
21	$\langle i, COMM, STK \rangle$	92.76	47.79	55.89	51.50	92.96	49.03	58.85	53.47
22	$\langle i, SEM, STK \rangle$	93.70	54.40	52.13	53.23	93.90	55.47	56.59	56.03
23	$\langle i, SENT, STK \rangle$	92.42	44.34	39.92	41.99	92.50	45.20	42.40	43.74
24	$\sum_{k \in \{PRED, OP, COMM\}} \langle i, k, STK \rangle$	93.62	53.26	60.05	56.44	93.77	54.06	63.21	58.26
25	$\sum_{k \in \{PRED, SEM\}} \langle i, k, STK \rangle$	93.90	55.26	59.50	57.30	94.13	56.57	63.12	59.67
26	$\sum_{k \in Scopes \setminus SENT} \langle i, k, STK \rangle$	94.09	56.65	59.68	58.11	94.21	57.21	62.61	59.80
27	$\sum_{k \in Scopes} \langle i, k, STK \rangle$	94.14	57.41	57.88	57.63	94.29	**58.11**	61.10	59.56
28	$\langle j, SENT, PTK \rangle$	92.11	45.02	**69.96**	53.51	91.92	44.27	67.39	53.43
29	$\sum_{k \in \{PRED, SEM\}} \langle i, k, STK \rangle + \langle PAS, SENT, PTK \rangle$	94.05	55.68	66.01	**60.40**	94.16	56.18	**68.36**	**61.67**
30	$\sum_{k \in Scopes \setminus SENT} \langle i, k, STK \rangle + \langle PAS, SENT, PTK \rangle$	**94.30**	**57.95**	62.62	60.19	**94.36**	58.07	64.94	61.31

sive than sequence kernels. For instance, in order to obtain the same performance as of $\langle CONST_{AUG}, PRED, STK \rangle$ (19B), i.e. a single kernel with an F-Measure 56.52, it requires several sequence kernels, hence much more effort. The performance of the different $CONST$ kernels relative to each other resembles the results of the WRD kernels. The best scope is $PRED$ (19). By far the worst performance is obtained by the $SENT$ scope (23). The combination of $PRED$ and SEM scope achieves an F-Measure of 59.67% (25B), which is already slightly better than the best configuration of sequence kernels (18).

The performance of the PAS kernel (28A) with an F-Measure of 53.51% is slightly worse than the best single plain $CONST$ kernel (19A). The PAS kernel and the $CONST$ kernels are complementary, since their best combination (29B) achieves an F-Measure of 61.67% which is significantly better than the best combination of $CONST$ kernels (26B) or sequence kernels (18).

7.5.5. Combination of Kernel Types

Table 7.8 lists the results of the different kernel type combinations. The convolution kernels outperform VK. However, if VK is added to the best TKs, the best SKs, or both, a slight increase in F-Measure is achieved. The best performance with an F-Measure of 62.61% is obtained by combining all kernels though the best SKs only have a marginal impact.

7.6. Error Analysis

It is difficult to state precisely what the shortcomings of the proposed approach presented in this chapter are. We found that the most predictive scope for the different kernels is the predicate scope. However, we found that our automatic recognition of the nearest predicate is not always correct. For instance, we assume that the nearest predicate (according to the syntactic relation path) is also the predicate which relates to the candidate opinion holder. There are several cases, in which this is, unfortunately, not the case.

Table 7.8.: Results of kernel combinations.

Combination	**Acc.**	**Prec.**	**Rec.**	**F.**
VK	93.63	53.28	59.37	56.16
best SKs	94.21	57.64	59.81	58.70
best TKs	94.16	56.18	**68.36**	61.67*
VK + best SKs	94.34	58.44	61.27	59.82*
VK + best TKs	94.33	57.41	68.03	62.27*
best SKs + best TKs	94.49	**59.22**	63.96	61.49*
VK + best SKs + best TKs	**94.53**	59.10	66.57	**62.61***†

*: significantly better than best SKs; †: significantly better than best TKs; all convolution kernels are significantly better than VK; statistical significance is based on a paired t-test using $p < 0.05$

Moreover, the recognition of nominalizations depends on a lexicon of those predicates. However, this lexicon has only a limited coverage and several entries are ambiguous. For instance, *opposition* may be a predicate but it can also refer to the political parties opposing a government. Our procedure cannot make such a distinction. It is fairly difficult to estimate the impact of these shortcomings as we believe that by using a combination of different kernels with different scopes, the incorrect processing of individual structures may be compensated by the correct processing of other structures. For instance, the predicate scope may be computed incorrectly but the semantic scope may still comprise the actual predicate relating to the candidate opinion holder.

We encountered similar problems for the semantic role labeling. For instance, the assignment of roles for arguments of nominalizations is often incorrect (either incorrect constituents are chosen or an argument is not assigned to a constituent at all). Since, however, the relation between nominalizations and their arguments is usually restricted to short-range dependencies, these relations may often be implicitly encoded in the constituency parse subtrees that we use.

7.7. Conclusion

In this chapter, we compared convolution kernels for opinion holder extraction. Similar to the insights gained by the text classification tasks in sentiment analysis presented in previous chapters, opinion holder extraction, too, requires the consideration on various linguistic aspects. In terms of convolution kernels we obtained following results:

We showed that, in general, a combination of two scopes, namely the scope immediately encompassing the candidate opinion holder and its nearest predicate and the subclause containing the candidate opinion holder, provide best performance. The usage of the entire sentence for convolution kernels, i.e. the scope which requires no linguistically motivated processing, results in a very poor performance.

The fact that the scopes having the nearest opinion word or communication word as a boundary do not perform best does not mean that the knowledge of these semantic categories is not relevant for this type of classification. Indeed, we found that generalizing sequences or augmenting trees with these categories (rather than using them for scope boundaries) results in a consistent improvement.

Tree kernels containing constituency parse information and semantic roles achieve better performance than sequence kernels or vector kernels using a manually designed feature set. A combination of different kernel types is effective. Best performance is achieved if all kernels are combined. These results suggest that various levels of representation in various types of kernels are a promising solution for opinion holder extraction.

8. Conclusion & Future Work

8.1. Conclusion

In this thesis, we presented various subtasks in sentiment analysis in which the consideration of linguistic knowledge is useful. Linguistic knowledge can be incorporated in several ways as will be presented below:

In *sentence-level polarity classification*, we added linguistic features and features counting polar expressions to bag of words. The addition of features counting polar expressions to bag of words results in a great performance gain. However, to some extent general linguistic features not containing knowledge about polarity, such as depth of a word leaf node in the syntactic parse tree or WordNet hypernyms, can also increase performance in the absence of polar expressions. In addition, the combination of the two feature types (on top of bag of words) is also slightly better than the best individual result (i.e. the combination of bag of words and polar expressions). Therefore, in order to obtain the best overall result, the inclusion of linguistic features is necessary.

In order to *distinguish between definite and indefinite polar sentences*, we devised a rule-based classifier based on features derived from linguistic insights, such as polar expressions indicating middle-of-the-road polarity and various groups of function words (e.g. detensifiers or concessive conjunctions). The resulting classifier performs on a par with a k-Nearest Neighbour Classifier and also outperforms Support Vector Machines when less than labeled 300 training instances are considered. The rule-based classifier may be outperformed by a supervised classifier, such as Support Vector Machines, but unlike the supervised classifier it does not require labeled in-domain data but exclusively relies on linguistic insights which should be generally applicable.

In *topic-related polarity classification*, a ranker using polar expressions, some lightweight linguistic features (based on part-of-speech information, strength of polarity, intensification, and negation), and a feature accounting for the spatial distance between polar expression and topic word clearly outperforms a cascade of sentence-retrieval used in conjunction with two text classifiers using simple bag-of-words features to select subjective sentences and sentences whose polarity matches the given target polarity.

In a detailed study on the effectiveness of *bootstrapping algorithms for document-level polarity classification*, we found that the incorporation of linguistic knowledge (that is relevant for the task) is actually a requirement for the pertaining bootstrapping algorithm to work well. Semi-supervised learning depends on a very predictive feature set. On a cross-domain evaluation the usage of in-domain adjectives and adverbs, i.e. the restriction of the feature set towards a particular linguistic part of speech, is considerably more effective than a plain bag-of-words feature set in which frequent non-stopwords are used. Unfortunately, the incorporation of further linguistic knowledge in that class of classifiers is not effective. The situation is different, however, if one considers another bootstrapping method in which a supervised classifier self-trained by a rule-based classifier is considered. In contrast to machine learning classifiers where some considerable performance is usually already achieved by employing bag of words, be it unrestricted or restricted – as in the case of semi-supervised learning[1] – which can be difficult to beat in certain tasks, a rule-based classifier is usually more sensitive to the incorporation of linguistic knowledge. We found that the more linguistic knowledge about contextual polarity is encoded in a rule-based classifier (i.e. basic word sense disambiguation, negation modeling, and emphasizing certain constructions/expressions which convey a higher polar intensity), the better the self-trained classifier becomes. Not only can this insight be considered a general justification for linguistic modeling of polarity but it can also be regarded as an incentive for further linguistic modeling beyond the modeling that has been presented in this thesis (see Section 8.2 for ideas of more sophisticated rule-based classification).

Finally, modeling *opinion holder extraction with convolution kernels* also requires the

[1] The usage of in-domain adjectives and adverbs should still be considered a bag-of-words feature set.

consideration of linguistic insights. For a good performance various levels of representation (beyond plain sequential word information), in particular, deeper linguistic information, as provided by parse trees or semantic-role labeling, are required and work effectively when used in tree kernels. Moreover, a combination of two scopes, a scope with the candidate opinion holder and its nearest predicate being the boundaries and a scope with the subclause in which the candidate opinion holder is embedded, outperform other scopes, in particular, the simplest scope requiring no linguistically motivated processing, i.e. the entire sentence.

Unfortunately, the answer to the question of what gain in general knowledge has been achieved in this thesis is less straightforward than pinpointing certain effective ways of incorporating linguistic knowledge in specific subtasks in sentiment analysis. This thesis did not propose a new theory accounting for sentiment analysis as a whole and I have doubts whether such a theory can ever be devised. Moreover, it might not even be necessary. In this thesis, I instead tried to determine appropriate methods from natural language processing (NLP) for specific subtasks in sentiment analysis (and this usually involved linguistic feature engineering). I assume that each subtask can be characterized by specific task-independent properties or parameters settings which suggest the applicability of certain NLP methods.

For instance, in this thesis it could be established that for supervised text classification in sentiment analysis the level of granularity is a property which decides on which features are likely to be effective. In supervised document-level classification, bag of words (including higher order ngrams) perform well while in sentence-level classification, more advanced linguistic features and generalizing features relying on the knowledge of subjective expressions are effective. Not only the level of granularity but also the type of classifier has an impact on the effectiveness of linguistic knowledge. For example, in document-level rule-based classification the incorporation of linguistic knowledge is far more effective than in supervised machine learning. I also considered the task of opinion holder extraction which bears some significant similarity to common NLP tasks, such as relation extraction and semantic role labeling. It is, therefore, no surprise that sequen-

tial information and structural information in the form of convolution kernels are helpful which have also been successfully applied to those common NLP tasks mentioned above.

These examples support the view that the effectiveness of certain NLP methods on specific subtasks in sentiment analysis can be explained with the help of specific properties of those subtasks. I argue that establishing the dependencies between settings and effectiveness of NLP methods requires general knowledge about NLP methods rather than an immense task-specific knowledge. The task-specific knowledge is, however, useful for fine-tuning the feature set and thus obtain state-of-the-art performance. Furthermore, these regularities should also enable the prediction of appropriate NLP methods if a new subtask in sentiment analysis were considered.

8.2. Future Work

This section briefly outlines possible extensions of methods presented in this thesis and other possible scenarios related to these tasks or methods which may be worthwhile examining in future work:

- **Bootstrapping Supervised Classifiers with more Complex Rule-Based Classification:** Our experiments on bootstrapping supervised classifiers with rule-based classification (Chapter 6) suggest that the more complex the rule-based classifier is, the better the supervised classifier performs. Therefore, more complex rule-based polarity classifiers than the ones presented in this thesis might be worthwhile examining.

  One way of extending the rule-based classifier could be by assigning more fine-grained weights to polar expressions. In this thesis, we proposed the weight of 1 to plain polar expressions and double the weight if the polar expression happens to be in an intensifying context. Brooke et al. (2009) annotate all polar expressions in a polarity lexicon with polar scores on a scale between -5 and $+5$. Such additional annotation should enable a more accurate distinction between different polar expressions.

Another way of extending the current rule-based classification could be by enhancing the negation model. Currently, we use fixed window size for the scope of a negation. However, recently Jia, Yu, and Meng (2009) showed that polarity classification improves the more linguistically accurate the scope model becomes. The best performance is obtained by a scope model using syntactic information.

Furthermore, some kind of compositional semantics for sentiment analysis, such as (Moilanen & Pulman, 2007), could be employed in order to combine the scores of polar expressions from different clauses in a sentence[2] in order to compute the score of the overall sentence. Currently, the scores of disambiguated polar expressions are just summed.

- **Bootstrapping Methods using Rule-Based Classification Applied to Other Tasks:** The bootstrapping method using rule-based classification as presented in Chapter 6 may also be effective for the other subtasks in sentiment analysis which have also been discussed in this thesis.

  The task of distinguishing between indefinite polarity and definite polarity as discussed in Chapter 4 might be a suitable candidate for this method. In this task, two different types of features (i.e. bag of words and a set of linguistically motivated high-level features) similar to the two feature sets used for bootstrapping (traditional) polarity classifiers had been presented. Due to these similarities, the application of this bootstrapping method should be fairly straightforward.

  The application of this method to opinion holder extraction, however, might be more difficult as a sufficiently robust domain-independent rule-based classifier is required for this task. Given that even fully supervised classifiers with a rich feature set using various levels of information still produce comparably low performance, the construction of such a rule-based classifier appears challenging.

- **Convolution Kernels for Target Extraction of Opinions:** As convolution

[2] Thus, one could differentiate between polar expressions from the main clause and polar expressions from subordinate clauses.

kernels applied to opinion holder extraction produced promising results, one might also wonder whether similar results can be obtained for targets of opinions. The major problem in this scenario is that there is a significantly greater diversity of linguistic units representing a target. While on the MPQA-corpus opinion holders tend to be realized as noun phrases, targets can assume virtually any shape of constituent. This is quite intuitive since opinions may be directed towards a certain person, thing, behaviour, attitude, or event. To make it worse, we found that there is a considerable amount of targets which cannot be matched onto any linguistic constituent. We observed that this is often the case when the target is an entire proposition. Apparently, manually annotating the scope of such complex structures is more difficult than that of simple concrete objects, such as persons or things. Even if those cases were neglected, the heterogeneity of targets would increase the instance space dramatically which would have a severe impact on the running time of the convolution kernel algorithm.

Alternatively, these experiments could also be carried out on corpora providing similar annotation. The JDPA Sentiment Corpus (Kessler et al., 2010) or the Darmstadt Service Review Corpus (Toprak et al., 2010) may be more suitable, since they focus on product/web-serves. Thus, the entities labeled as opinion targets are more restricted to specific linguistic entities, such as noun phrases.

- **Unsupervised Generalization for Sentiment Analysis:** Throughout many experiments in this thesis, generalizing from lexical units often resulted in an improvement of performance, e.g. the knowledge of polar expressions or WordNet hypernyms on sentence-level polarity classification helped when corresponding features were added to bag of words. Generalization is always useful when there is sparse lexical information. This is usually the case when fine-grained text classification, such as sentence level or expression level, or entity extraction is considered. Unfortunately, all types of generalization we used in this work have been knowledge-driven. In future work, one might examine various unsupervised generalization techniques (e.g. clustering) for their effectiveness in sentiment analysis.

- **Subjectivity Word Sense Disambigation:** As it has been suggested in this thesis several times, one major downside of the polarity lexicons used is that they do not properly distinguish between the different senses of polar expressions. An expression may be subjective only if it conveys a particular sense. In several experiments, we carried out some basic disambiguation using part-of-speech information, however, there are many ambiguous polar expressions which have a unique part of speech. For those cases, we have been unable to provide a suitable disambiguation. Though some more sophisticated form of subjectivity word sense disambiguation (Akkaya et al., 2009) might be worthwhile to pursue in future work, the necessary resources (i.e. lexicons and labeled corpora) are currently not available.

A. Evaluation Measures

A.1. Measures for Classification and Extraction

The most common evaluation measure for classification is ***Accuracy***:

$$Accuracy = \frac{\#\text{correct instances}}{\#\text{correct instances} + \#\text{incorrect instances}} \tag{A.1}$$

For classification tasks in which the performance of individual classes is to be evaluated, measures other than Accuracy are usually considered. This is in particular true of extraction tasks, in which only the positive class, i.e. the instances to be extracted, is of interest. For these cases, *Precision*, *Recall*, and *F-Measure* are considered. They are defined by *true positives* which are the instances which belong to the class to be evaluated and are correctly classified, *false positives* which are not instances of the class to be evaluated but are misclassified as such, and *false negatives* which are instances of the class to be evaluated but are misclassified as instances of another class.

The measure that evaluates the proportion of correctly classified instances (of the class that is to be evaluated) within the set of instances predicted to be of that class is ***Precision*** which is formally defined by Formula A.2:

$$Precision = \frac{\#true\ positives}{\#true\ positives + \#false\ positives} \tag{A.2}$$

Precision does not take into consideration the instances of a class that have been erroneously assigned to another class. This is, however, done by ***Recall*** whose formal definition is given in Formula A.3:

$$Recall = \frac{\#true\ positives}{\#true\ positives + \#false\ negatives} \tag{A.3}$$

Finally, ***F-Measure*** is an evaluation measure combining the complementary measures *Precision* and *Recall*. In this thesis, the most common form, the so-called *harmonic mean*, is used. The formal definition of this measure is given in Formula A.4:

$$F\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{A.4}$$

A.2. Measures for Ranking

A fairly simple ranking measure evaluating the rankings for a set of queries Q is ***Mean Reciprocal Rank (MRR)*** in which for each query the correct instance with the highest rank is considered. Its formal definition is given in Formula A.5:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{A.5}$$

While MRR is fairly restricted since only one correct instance is considered, ***Precision at Rank n (Prec@(n))*** considers all correct instances at the top n ranks:

$$Prec@(n) = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{\#\text{correct instances for query } i \text{ within top } n \text{ ranks}}{n} \tag{A.6}$$

Note that this definition is also sometimes referred to as *Average Precision at Rank n* since one actually calculates the average of the precision of individual rankings for a set of queries.

Finally, ***Mean Average Precision (MAP)*** is a measure which considers *all* correct instances within a ranking and not just the highest ranked instance or all instances to a certain cut-off level. It completely traverses each ranking and sums at each rank n at which a correct instance is found Prec@(n). This is additionally normalized by the number correct instances for that query in the entire collection:

$$MAP = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{\sum_{n=1}^{N} (Prec@(n) \cdot \delta(r))}{\#\text{ correct instances for } i \text{ within the entire collection}} \tag{A.7}$$

where

$$\delta(n) = \begin{cases} 1 & \text{if instance at rank } n \text{ is correct} \\ 0 & \text{else} \end{cases} \tag{A.8}$$

References

Agarwal, R., T.V., P., & Chakrabarty, S. (2008). "I Know What You Feel": Analyzing the Role of Conjunctions in Automatic Sentiment Analysis. In *Proceedings of the International Conference on Natural Language Processing (GoTAL).*

Akkaya, C., Wiebe, J., & Mihalcea, R. (2009). Subjectivity Word Sense Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Andreevskaia, A., & Bergler, S. (2008). When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT).* Columbus, OH, USA.

Aue, A., & Gamon, M. (2005). Customizing Sentiment Classifiers to New Domains: a Case Study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP).* Borovets, Bulgaria.

Banea, C., Mihalcea, R., & Wiebe, J. (2008). A Bootstrapping Method for Building Subjectivity Lexicons for Language with Scarce Resources. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).*

Banea, C., Mihalcea, R., Wiebe, J., & Hassan, S. (2008). Multilingual Subjectivity Analysis Using Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Beineke, P., Hastie, T., & Vaithyanathan, S. (2004). The Sentimental Factor: Improving Review Classification via Human-Provided Information. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., & Jurafsky, D. (2004). Extracting Opinion Propositions and Opinion Holders using Syntactic and Lexical Cues. In *Computing Attitude and Affect in Text: Theory and Applications.* Springer-Verlag.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Prague, Czech Republic.

Bloom, K., Garg, N., & Argamon, S. (2007). Extracting Appraisal Expressions. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).*

Bloom, K., Stein, S., & Argamon, S. (2007). Appraisal Extraction for News Opinion Analysis at NTCIR-6. In *Proceedings of the NTCIR-6 Workshop Meeting.* Tokyo, Japan.

Breck, E., Choi, Y., & Cardie, C. (2007). Identifying Expressions of Opinion in Context. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Brooke, J., & Hurst, M. (2009). Patterns in the Stream: Exploring the Interaction of Polarity, Topic, and Discourse in a Large Opinion Corpus. In *Proceedings of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement (TSA).* Hong Kong, China.

Brooke, J., Tofiloski, M., & Taboada, M. (2009). Cross-Linguistic Sentiment Analysis: From English to Spanish. In *Proceedings of Recent Advances in Natural Language Processing (RANLP).*

Bunescu, R. C., & Mooney, R. J. (2005). Subsequence Kernels for Relation Extraction. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS).* Vancouver, Canada.

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of the Conference on North American Chapter of the Association for Computational Linguistics (ANLP)* (pp. 132 – 139). Seattle, Washington.

Chesley, P., Vincent, B., Li Xu, F., & Srihari, R. K. (2005). Using Verbs and Adjectives to Automatically Classify Blog Sentiment. In *Proceedings of the AAAI Symposium on Computational Approaches to Analysing Blogs (AAAI-CAAW).*

Choi, Y., Breck, E., & Cardie, C. (2006). Joint Extraction of Entities and Relations for Opinion Recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Sydney, Australia.

Choi, Y., & Cardie, C. (2008). Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Choi, Y., & Cardie, C. (2009). Adapting a Polarity Lexicon using Integer Linear Programming for Domain-Specific Sentiment Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005). Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP).* Vancouver, Canada.

Collins, M., & Duffy, N. (2002). New Ranking Algorithms for Parsing and Tagging. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Philadelphia, USA.

Dang, H. T. (2009). Overview of the TAC 2008 Opinion Question Answering and Summarization Tasks. In *Proceedings of the Text Analysis Conference (TAC).* Gaithersburg, MD, USA.

Dasgupta, S., & Ng, V. (2009). Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL/IJCNLP).* Suntec, Singapore.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proceedings of the*

International World Wide Web Conference (WWW).

Dias, G., Lambov, D., & Noncheva, V. (2009). High-level Features for Learning Subjective Language across Domains. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM).*

Ding, X., & Liu, B. (2007). The Utility of Linguistic Rules in Opinion Mining. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR).*

Du, W., & Tan, S. (2009). An Iterative Reinforcement Approach for Fine-Grained Opinion Mining. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).*

Eguchi, K., & Lavrenko, V. (2006). Sentiment Retrieval using Generative Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Esuli, A., & Sebastiani, F. (2006a). Derminining Term Subjectivity and Term Orientation for Opinion Mining. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL).*

Esuli, A., & Sebastiani, F. (2006b). SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).* Genova, Italy.

Esuli, A., & Sebastiani, F. (2007). PageRanking WordNet Synsets: An Application to Opinion Mining. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Fillmore, C. J., Johnson, C. R., & Petruck, M. R. (2003). Background to FrameNet. *International Journal of Lexicography*, *16*, 235 – 250.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Ann Arbor, USA.

Gamon, M. (2004). Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis. In *Proceedings of*

the International Conference on Computational Linguistics (COLING). Geneva, Switzerland.

Gerani, S., Carman, M., & Crestani, F. (2009). Investigating Learning Approaches for Blog Post Opinion Retrieval. In *Proceedings of the European Conference in Information Retrieval (ECIR).*

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the Semantic Orientation of Adjectives. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL)* (pp. 174–181). Madrid, Spain.

He, B., Macdonald, C., He, J., & Ounis, I. (2008). An Effective Statistical Approach to Blog Post Opinion Retrieval. In *Proceedings of the Conference on Information and Knowledge Management (CIKM).*

Hiroshi, K., Tetsuya, N., & Hideo, W. (2004). Deeper Sentiment Analysis Using Machine Translation Technology. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Hu, M., & Liu, B. (2004). Mining and Summarizing Customer Reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD).* Seattle, WA, USA.

Hyland, K. (1998). *Hedging in Scientific Research Articles.* John Benjamins.

Jakob, N., & Gurevych, I. (2010a). Extracting Opinion Targets in a Single- and Cross-Domain Setting with Conditional Random Fields. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Boston, MA, USA.

Jakob, N., & Gurevych, I. (2010b). Using Anaphora Resolution to Improve Opinion Target Idenfication in Movie Reviews. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Uppsala, Sweden.

Jason, G. (1988). Hedging as a Fallacy of Language. *Informal Logic*, *10*(3), 169 – 175.

Jia, L., Yu, C., & Meng, W. (2009). The Effect of Negation on Sentiment Analysis and Retrieval Effectiveness. In *Proceedings of the Conference on Information and Knowledge Management (CIKM).*

Joachims, T. (1999a). Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning.* MIT Press.

Joachims, T. (1999b). Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings the International Conference on Machine Learning (ICML).*

Joachims, T. (2003). Transductive Learning via Spectral Graph Partitioning. In *Proceedings the International Conference on Machine Learning (ICML).* Washington, D.C., USA.

Karlgren, J., Eriksson, G., Täckström, O., & Sahlgren, M. (2010). Between Bags and Trees - Constructional Patterns in Text Used for Attitude Identification. In *Proceedings of the European Conference in Information Retrieval (ECIR).*

Kennedy, A., & Inkpen, D. (2005). Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. In *Proceedings of the Workshop on the Analysis of Formal and Informal Information Exchange during Negotiations (FINEXIN)* (Vol. 22).

Kessler, J. S., Eckert, M., Clarke, L., & Nicolov, N. (2010). The ICWSM JDPA 2010 Sentiment Corpus for the Automotive Domain. In *Proceedings of the International AAAI Conference on Weblogs and Social Media Data Challange Workshop (ICWSM-DCW).*

Kessler, J. S., & Nicolov, N. (2009). Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM).* San Jose, CA, USA.

Kilicoglu, H., & Bergler, S. (2008). Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective. *BMC Bioinformatics, 9 Supplement.*

Kim, J., Li, J.-J., & Lee, J.-H. (2009). Discovering the Discriminative Views: Measuring Term Weights for Sentiment Analysis. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation*

of Natural Language Processing (ACL/IJCNLP).

Kim, S.-M., & Hovy, E. (2005). Identifying Opinion Holders for Question Answering in Opinion Texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains.* Pittsburgh, USA.

Kim, S.-M., & Hovy, E. (2006). Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text.* Sydney, Australia.

Kingsbury, P., & Palmer, M. (2002). From TreeBank to PropBank. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).* Las Palmas, Spain.

Klein, D., & Manning, C. D. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Sapporo, Japan.

Klenner, M., Petrakis, S., & Fahrni, A. (2009). Robust Compositional Polarity Classification. In *Proceedings of Recent Advances in Natural Language Processing (RANLP).* Borovets, Bulgaria.

Kobayakawa, T. S., Kumano, T., Tanaka, H., Okazaki, N., Kim, J.-D., & Tsujii, J. (2009). Opinion Classification with Tree Kernel SVM Using Linguistic Modality Analysis. In *Proceedings of the Conference on Information and Knowledge Management (CIKM).*

Koppel, M., & Schler, J. (2006). The Importance of Neutral Examples for Learning Sentiment. *Computational Intelligence*, *22*(2), 100 – 109.

Kudo, T., & Matsumoto, Y. (2005). A Boosting Algorithm for Classification of Semi-Structured Text. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP).*

Li, Y., Bontcheva, K., & Cunningham, H. (2007). Experiments of Opinion Analysis on the Corpora MPQA and NTCIR-6. In *Proceedings of the NTCIR-6 Workshop Meeting.*

Light, M., Qiu, X. Y., & Srinivasan, P. (2004). The Language of Bioscience: Facts, Speculations, and Statements in Between. In *Proceedings of BioLINK.*

Lin, D. (1998). Dependency-based Evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems.* Granada, Spain.

Liu, B. (2006). Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. In (chap. 11: Opinion Mining). Springer-Verlag.

Liu, B., Hu, M., & Cheng, J. (2005). Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the International World Wide Web Conference (WWW)* (pp. 342–351).

Liu, F., Li, B., & Liu, Y. (2009). Finding Opinionated Blogs Using Statistical Classifiers and Lexical Features. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM).*

Macdonald, C., & Ounis, I. (2006). *The TREC Blog06 Collection* (Tech. Rep. No. TR-2006-226).

Macdonald, C., Ounis, I., & Soboroff, I. (2008). Overview of the TREC-2007 Blog Track. In *Proceedings of the Text Retrieval Conference (TREC).* Gaithersburg, MD, USA: NIST.

Macleod, C., Grishman, R., Meyers, A., Barrett, L., & Reeves, R. (1998). NOMLEX: A Lexicon of Nominalizations. In *Proceedings of EURALEX.* Liege, Belgium.

Matsumoto, S., Takamura, H., & Okumura, M. (2005). Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD).*

McDonald, R., Hannan, K., Neylon, T., Wells, M., & Reynar, J. (2007). Structured Models for Fine-to-Coarse Sentiment Analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Medlock, B., & Briscoe, T. (2007). Weakly Supervised Learning for Hedge Classification in Scientific Literature. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Meena, A., & Prabhabkar, T. (2007). Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis. In *Proceedings of the European Conference in Information Retrieval (ECIR).* Rome, Italy.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In *Proceedings of the International World Wide Web Conference (WWW)*.

Mihalcea, R., Banea, C., & Wiebe, J. (2007). Learning Multilingual Subjective Language via Cross-Lingual Projections. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, *3*, 235–244.

Mohammad, S., Dunne, C., & Dorr, B. (2009). Generating High-Coverage Semantic Orientation Lexicons from Overtly Marked Words and a Thesaurus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Moilanen, K., & Pulman, S. (2007). Sentiment Construction. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*. Borovets, Bulgaria.

Moschitti, A. (2006a). Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proceedings of the European Conference on Machine Learning (ECML)*. Berlin, Germany.

Moschitti, A. (2006b). Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL)*. Trento, Italy.

Moschitti, A. (2008). Kernel Methods, Syntax and Semantics for Relational Text Categorization. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. Napa Valley, USA.

Moschitti, A., Pighin, D., & Basili, R. (2008). Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, *34*(2), 193 – 224.

Mullen, T., & Collier, N. (2004). Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Narayanan, R., Liu, B., & Choudhary, A. (2009). Sentiment Analysis of Conditional Sen-

tences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Nasukawa, T., & Yi, J. (2003). Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In *Proceedings of the International Conference on Knowledge Capture (K-CAP).* Sanibel Island, FL, USA.

Ng, V., Dasgupta, S., & Arifin, S. M. N. (2006). Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING/ACL).* Sydney, Australia.

Nguyen, T.-V. T., Moschitti, A., & Riccardi, G. (2009). Convolution Kernels on Constituent, Dependency and Sequential Structures for Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Singapore.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, *39*(2), 103–134.

Nowson, S. (2009). Scary Films Good, Scary Flights Bad - Topic Driven Feature Selection for Classification of Sentiment. In *Proceedings of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement (TSA).*

Ounis, I., Macdonald, C., & Soboroff, I. (2008). Overview of the TREC-2007 Blog Track. In *Proceedings of the Text Retrieval Conference (TREC).* Gaithersburg, MD, USA.

Ounis, I., Macdonald, C., & Soboroff, I. (2009). Overview of the TREC Blog Track 2008. In *Proceedings of the Text Retrieval Conference (TREC).* Gaithersburg, MD, USA.

Ounis, I., Rijke, M. de, Macdonald, C., Mishne, G., & Soboroff, I. (2007). Overview of the TREC-2006 Blog Track. In *Proceedings of the Text Retrieval Conference (TREC).* Gaithersburg, MD, USA.

Pang, B., & Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

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

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Philadelphia, USA.

Picard, R. W. (1997). *Affective Computing*. MIT Press.

Popescu, A.-M., & Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

Qiu, L., Zhang, W., Hu, C., & Zhao, K. (2009). SELC: A Self-Supervised Model for Sentiment Classification. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. Hong Kong, China.

Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman.

Raaijmakers, S., Troung, K., & Wilson, T. (2008). Multimodal Subjectivity Analysis of Multiparty Conversation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rao, D., & Ravichandran, D. (2009). Semi-Supervised Polarity Lexicon Induction. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL)*. Athens, Greece.

Ravichandran, D., Hovy, E., & Och, F. J. (2003). Statistical QA - Classifier vs. Re-ranker: What's the Difference. In *Proceedings of the ACL Workshop on Multilingual Summarization and Question Answering*.

Riloff, E. (1996). An Empirical Study of Automated Dictionary Construction for Information Extraction. *Artificial Intelligence*, *85*.

Riloff, E., & Wiebe, J. (2003). Learning Extraction Patterns for Recognizing Subjective Expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Salvetti, F., Reichenbach, C., & Lewis, S. (2006). Opinion Polarity Identification of

Movie Reviews. In J. G. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing Attitude and Affect in Text: Theory and Applications* (pp. 303–316). Springer-Verlag.

Santos, R. L., He, B., Macdonald, C., & Ounis, I. (2009). Integrating Proximity to Subjective Sentences for Blog Opinion Retrieval. In *Proceedings of the European Conference in Information Retrieval (ECIR).*

Sarmento, L., Carvalho, P., Silva, M. J., & Oliveira, E. de. (2009). Automatic Creation of a Reference Corpus for Political Opinion Mining in User-Generated Content. In *Proceedings of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement (TSA).*

Scott, S., & Matwin, S. (1998). Text Classification Using WordNet Hypernyms. In S. Harabagiu (Ed.), *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference* (pp. 38 – 44). Somerset, New Jersey: Association for Computational Linguistics.

Seki, Y., Evans, D. K., Ku, L.-W., Chen, H.-H., Kando, N., & Lin, C.-Y. (2007). Overview of Opinion Analysis Pilot Task at NTCIR-6. In *Proceedings of the NTCIR-6 Workshop Meeting.* Tokyo, Japan.

Shen, D., Leidner, J. L., Merkel, A., & Klakow, D. (2007). The Alyssa System at TREC 2006: A Statistically-Inspired Question Answering System. In *Proceedings of the Text Retrieval Conference (TREC).* Gaithersburgh, MD, USA: NIST.

Snyder, B., & Barzilay, R. (2007). Multiple Aspect Ranking using the Good Grief Algorithm. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).*

Somasundaran, S., Namata, G., Wiebe, J., & Getoor, L. (2009). Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Somasundaran, S., & Wiebe, J. (2009). Recognizing Stances in Online Debates. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natu-*

ral Language Processing of the Asian Federation of Natural Language Processing (ACL/IJCNLP).

Somasundaran, S., Wilson, T., Wiebe, J., & Stoyanov, V. (2007). QA with Attitude: Exploiting Opinion Type Analysis for Improving Question Answering in On-line Discussions and the News. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM).*

Spertus, E. (1997). Smokey: Automatic Recognition of Hostile Messages. In *Proceedings of Innovation Applications in Artificial Intelligence (IAAI)* (pp. 1058 – 1065).

Stone, P. J., Dumphy, D. C., Smith, M. S., Ogilvie, D. M., & associates. (1966). *The General Inquirer: A Computer Approach to Content Analysis.* MIT Press.

Stoyanov, V., & Cardie, C. (2008). Annotating Topics of Opinions. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).*

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In *Proceedings of the European Conference in Information Retrieval (ECIR).*

Tan, S., Wang, Y., & Cheng, X. (2008). Combining Learn-based and Lexicon-based Techniques for Sentiment Detection without using Labeled Examples. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR).* Singapore.

Taylor, J., & Christianini, N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Thet, T. T., Na, J.-C., Khoo, C. S., & Shakthikumar, S. (2009). Sentiment Analysis of Movie Reviews on Discussion Boards using a Linguistic Approach. In *Proceedings of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement (TSA).*

Toprak, C., Jakob, N., & Gurevych, I. (2010). Sentence and Expression Level Annotation of Opinions in User-Generated Discourse. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).* Uppsala, Sweden.

Turney, P. (2002). Thumbs up or Thumbs down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the Annual Meeting of*

the Association for Computational Linguistics (ACL) (p. 417-424). Philadelphia, Pennsylvania.

Turney, P., & Littman, M. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. In *Proceedings of ACM Transactions on Information Systems (TOIS).*

Velikovich, L., Blair-Goldensohn, S., Hannan, K., & McDonald, R. (2010). The Viability of Web-derived Polarity Lexicons. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).*

Whitelaw, C., Garg, N., & Argamon, S. (2005). Using Appraisal Groups for Sentiment Analysis. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)* (pp. 625–631). Bremen, Germany.

Wiebe, J. (1994). Tracking Point of View in Narrative. *Computational Linguistics*, *20*(2), 233 – 287.

Wiebe, J., & Mihalcea, R. (2006). Word Sense and Subjectivity. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING/ACL).*

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

Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning Subjective Language. *Computational Linguistics*, *30*(3).

Wiebe, J., Wilson, T., & Cardie, C. (2003). Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, *1*, 2.

Wiegand, M., & Klakow, D. (2009a). Predictive Features in Semi-Supervised Learning for Polarity Classification and the Role of Adjectives. In *Proceedings of the Nordic Conference on Computational Linguistics (NoDaLiDa).* Odense, Denmark.

Wiegand, M., & Klakow, D. (2009b). The Role of Knowledge-based Features in Polarity Classification at Sentence Level. In *Proceedings of the International FLAIRS*

conference (FLAIRS).

Wiegand, M., & Klakow, D. (2009c). Topic-Related Polarity Classification of Blog Sentences. In *Proceedings of the Portuguese Conference on Artificial Intelligence (EPIA)* (pp. 658 – 669). Springer-Verlag.

Wiegand, M., & Klakow, D. (2010a). Bootstrapping Supervised Machine-learning Polarity Classifiers with Rule-based Classification. In *Proceedings of the Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA).*

Wiegand, M., & Klakow, D. (2010b). Convolution Kernels for Opinioin Holder Extraction. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).*

Wiegand, M., & Klakow, D. (2010c). Predictive Features for Detecting Indefinite Polar Sentences. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).*

Wilson, T. (2008a). Annotating Subjective Content in Meetings. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).*

Wilson, T. (2008b). *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States.* Unpublished doctoral dissertation, University of Pittsburgh.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP).* Vancouver, Canada.

Wilson, T., Wiebe, J., & Hwa, R. (2004). Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).*

Yang, Y., & Pederson, J. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings the International Conference on Machine Learning (ICML)* (pp. 412–420). Nashville, US.

Zagibalov, T., & Carroll, J. (2008). Automatic Seed Word Selection for Unsupervised

Sentiment Classification of Chinese Text. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Zhai, C., & Lafferty, J. (2001). A Study of Smoothing Methods for Language Models Applied to Information Retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR).* New Orleans, USA.

Zhang, D., & Lee, W. S. (2003). Question Classification using Support Vector Machines. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR).* Toronto, Canada.

Zhang, M., & Ye, X. (2008). A Generation Model to Unify Topic Relevance and Lexicon-based Sentiment for Opinion Retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR).*

Zhang, M., Zhang, J., & Su, J. (2006). Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL).* New York City, USA.

Zhang, Y., Wang, R., & Uszkoreit, H. (2008). Hybrid Learning of Dependency Structures from Heterogeneous Linguistic Resources. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL).* Manchester, United Kingdom.

Zhao, J., Liu, K., & Wang, G. (2008). Adding Redundant Features for CRFs-based Sentence Sentiment Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Zhuang, L., Jing, F., & Zhu, X.-Y. (2006). Movie Review Mining and Summarization. In *Proceedings of the Conference on Information and Knowledge Management (CIKM).* Arlington, VA, USA.