Sentiment Analysis of German Twitter

Uladzimir Sidarenka

Dissertation eingereicht
bei der Humanwissenschaftlichen Fakultät
der Universität Potsdam

2019

This work is licensed under a Creative Commons License:
Attribution 4.0 International.
This does not apply to quoted content from other authors.
To view a copy of this license visit
https://creativecommons.org/licenses/by/4.0/

Datum der Einreichung: 2019/03/11

Wissenschaftlicher Betreuer: Prof. Dr. Manfred Stede
Gutachter: Prof. Dr. Jacob Eisenstein

Tag der mündlichen Prüfung: 2019/07/12

Published online at the
Institutional Repository of the University of Potsdam:
https://doi.org/10.25932/publishup-43742
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-437422

Ich erkläre hiermit, dass die Arbeit selbständig und ohne unzulässige Hilfe Dritter verfasst wurde und bei der Abfassung nur die in der Dissertation angegebenen Hilfsmittel benutzt sowie alle wörtlich oder inhaltlich übernommenen Stellen als solche gekennzeichnet wurden.

Potsdam, 2019
Vladimir Sidorenko
(Uladzimir Sidarenka)

To my family

Abstract

The immense popularity of online communication services in the last decade has not only upended our lives (with news spreading like wildfire on the Web, presidents announcing their decisions on Twitter, and the outcome of political elections being determined on Facebook) but also dramatically increased the amount of data exchanged on these platforms. Therefore, if we wish to understand the needs of modern society better and want to protect it from new threats, we urgently need more robust, higher-quality natural language processing (NLP) applications that can recognize such necessities and menaces automatically, by analyzing uncensored texts. Unfortunately, most NLP programs today have been created for standard language, as we know it from newspapers, or, in the best case, adapted to the specifics of English social media.

This thesis reduces the existing deficit by entering the new frontier of German online communication and addressing one of its most prolific forms—users' conversations on Twitter. In particular, it explores the ways and means by how people express their opinions on this service, examines current approaches to automatic mining of these feelings, and proposes novel methods, which outperform state-of-the-art techniques. For this purpose, I introduce a new corpus of German tweets that have been manually annotated with sentiments, their targets and holders, as well as lexical polarity items and their contextual modifiers. Using these data, I explore four major areas of sentiment research: (i) generation of sentiment lexicons, (ii) fine-grained opinion mining, (iii) message-level polarity classification, and (iv) discourse-aware sentiment analysis. In the first task, I compare three popular groups of lexicon generation methods: dictionary-, corpus-, and word-embedding–based ones, finding that dictionary-based systems generally yield better polarity lists than the last two groups. Apart from this, I propose a linear projection algorithm, whose results surpass many existing automatically-generated lexicons. Afterwords, in the second task, I examine two common approaches to automatic prediction of sentiment spans, their sources, and targets: conditional random fields (CRFs) and recurrent neural networks, obtaining higher scores with the former model and improving these results even further by redefining the structure of CRF graphs. When dealing with message-level polarity classification, I juxtapose three major sentiment paradigms: lexicon-, machine-learning–, and deep-learning–based systems, and try to unite the first and last of these method groups by introducing a bidirectional neural network with lexicon-based attention. Finally, in order to make the new classifier aware of microblogs' discourse structure, I let it separately analyze the elementary discourse units of each tweet and infer the overall polarity of a message from the scores of its EDUs with the help of two new approaches: latent-marginalized CRFs and Recursive Dirichlet Process.

Acknowledgments

At the beginning of this thesis, I would like to say a big thank you to all people who faithfully supported me throughout the years of working on my dissertation: my colleagues, my friends, my teachers, my advisor and, first of all, my family. I am heavily indebted to all these supporters and am absolutely convinced that this work would have never come into existence without them. I am also sure that it would take me another thesis to express the whole debt of gratitude that I owe each of my helpers, but am afraid that none of them would like to wait that long for me to finish it. Therefore, as hard as it is, I have to be brief in these acknowledgments and will describe all the infinite gratefulness that I feel with just few, but utterly sincere words.

First and foremost, I would like to thank my family, who were incessantly standing by my side throughout this difficult and exhausting endeavor. Unfortunately, they also were those who suffered most from my absence at home and the lack of spare time. So, taking the opportunity to speak to them from these pages, I would like to apologize for my heavily neglected family duties and promise that I will make up for them twofold after the defense.

Second, I would like to express my gratitude to my advisor, Manfred Stede, for his courage in accepting me as a Ph.D. student, for the infinite patience during the course of this work, for the numerous valuable suggestions, and, of course, for the time he invested in discussing and reviewing parts of this work.

Using the chance, I would also like to thank Jacob Eisenstein for his consent to be the reviewer of my dissertation and that he unknowingly, through his papers, has become my second, imaginary advisor (as a complex person I allow myself to have imaginary mentors).

For sure, writing this thesis would not have been half as fun if there had not been such wonderful colleagues and friends as Arne Neumann, André Herzog, Tatjana Scheffler, Thomas Hanneforth, Yulia Grishina, Andreas Peldszus, Sarah Hemmen, Daniel Schultheis, Misha Lukashyk, Polina Dovnar, Alexey Cheusov, and Nikolai Krot, who not only supported me in times of despair but also shared with me occasional moments of joy. There are definitely many others who should have been mentioned here, but as I said, I would like to be brief, and if I failed to name someone explicitly in these acknowledgments, please be assured that I did it not for lack of gratitude but because of the limited writing space and time. And at least to mitigate this omission, I would like to address my final big thank you to all those who remained unnamed on these pages, but feel that they have facilitated this work.

Vladimir Sidorenko
(Uladzimir Sidarenka)

Contents

List of Figures

List of Tables

Foreword

Das Internet ist für uns alle Neuland.

—Angela Merkel, 2013

As social media become more and more popular, the need for automatic analysis of their data rises. This analysis, however, is greatly complicated by the fact that the language style used on the Web is fundamentally different from the style of official documents and newspaper articles. Indeed, sentences like the ones shown in Example 0.0.1 (provided by Han and Baldwin, 2011) are very unlikely to appear in the transcript of an Oval Office address or in an editorial of The New York Times, even though such wording is commonplace on English Twitter.

Example 0.0.1

u must be talkin bout the paper but I was thinkin movies

. . . so hw many time remaining so I can calculate it?

These differences become even more marked when it comes to emotional speech, where people express their excitement, sadness, happiness, approval or disapproval. Compare, for instance, the following passages from Example 0.0.2, in which a Telegraph reporter and a Twitter user describe their feelings about the resignation of Boris Johnson, UK Foreign Secretary, who gave up his office in criticism of the government's Brexit plan.

Example 0.0.2

Je regrette. I cannot express how horrified I am that Boris Johnson stepped down. He was the standard-bearer of those who wanted not to get out of the single market, but to curtail the move to political union in a federal state run by the likes of Juncker. (Ayesha Vardag, The Telegraph)

That muffled sound is Boris Johnson kicking himself that he didn't resign before David Davis. Two down and he's the second (@Kevin_Maguire,

Twitter)

As you can see, not only the ways of expression are different, but the attitudes of the authors are contradictory as well. And nowadays it is the domain of social media that is steadily gaining popularity, and that wields more and more influence on the opinions of common people, predetermining their preferences, choices, and political views. This trend is inexorable; this trend is global; and, unfortunately, this trend opens up new possibilities for misuse of online services as an instrument of political deception.

One way to avert the looming danger of deliberate manipulation of public opinion is to monitor social networks in real time in order to discover suspicious activities or unexplainable fluctuations of people's attitudes. A crucial prerequisite for such monitoring though is reliable, high-quality NLP tools that can analyze users' dispositions automatically in a split second.

Motivation

Automatic mining of people's opinions from text is exactly what the field of knowledge called *sentiment analysis* or *opinion mining*[1] is concerned with, and what we[2] will work on in this dissertation. In particular, we are going to analyze users' attitudes on German Twitter—a linguistic register whose natural language processing is aggravated not only by the specifics of social media but also by the scarceness of resources, systems, and established baselines. Nevertheless, we decided to address precisely this domain because:

- German is the most spoken first language in the European Union, being the mother-tongue for 18% of EU citizens;[3]
- Germany has traditionally played a major role in the European Government, and, as such, it was one of the main driving forces in solving several European crises, including the Ukrainian conflict, the prevention of Greek sovereign default, and Brexit;
- Numerous internal problems (refugee crisis, rise of right-wing populism, and unstable ratings of political parties) make German politics susceptible to external influence.

Our choice of the Twitter platform was motivated by the following factors:

[1] Following Liu (2012), I consider the terms *sentiment analysis* and *opinion mining* as synonyms.

[2] Throughout this dissertation, I will use the pronoun "we" in recognition of the efforts made by all people mentioned in the acknowledgments, and in recognition of your efforts as a reader who will struggle with me through the pages of this work. This usage, however, does not imply that either you or any of my supporters share the same opinions or are responsible for any of the claims.

[3] `https://en.wikipedia.org/wiki/Languages_of_the_European_Union`

- First of all, Twitter is the second most popular social network in Germany,[4] with 4.9 million monthly active users (as of 2017);[5]
- Second, Twitter's sociolect is at the cutting edge of modern language development, and new linguistic phenomena introduced on this service are likely to percolate into other social media and might even find their way into the standard language as well;
- Finally, the abundance and accessibility of data on this platform allows the researchers to analyze virtually any topic, from North Korean nuclear weapons to Lady Gaga's dress, getting messages (and opinions) from users of different income, gender, and age.

Research Questions

Unfortunately, despite its popularity and social importance, German Twitter has largely been ignored by computational linguistics in general, and in particular by its opinion mining branch. With this dissertation, we hope to make up this leeway by presenting a new sentiment corpus of German microblogs and conducting an extensive study of existing and novel opinion mining methods on these data. By doing so, we want to answer the following questions:

- **Can we apply opinion mining methods devised for standard English to German Twitter?**

  Since there had been literally no attempts to analyze sentiments in German social media when we started working on this thesis, as a first step, we decided to check whether we could reuse existing English solutions without further ado.

- **Which groups of approaches are best suited for which sentiment tasks?**

  Because sentiment analysis is a wide research field, which operates on various linguistic levels and addresses many different problems with their own approaches and evaluation metrics, we want to know which approaches (rule-based or machine-learning ones, systems that operate on lexical taxonomies or those that utilize corpus data) work best for specific sentiment tasks;

- **How much do word- and discourse-level analyses affect message-level sentiment classification?**

  Despite the wide variety of problems addressed by opinion mining, one of them—message-level polarity classification—is commonly considered as the central task in sentiment analysis of social media. Due to its importance and central role, we would

[4] `https://digiday.com/marketing/state-social-platform-use-germany-5-charts/`
[5] `https://luckyshareman.com/blog/die-twitter-nutzung-in-deutschland/`

like to see which linguistic level (subsentential [*i.e.*, the level of word] or suprasentential [*i.e.*, the level of discourse]) contributes more to determining the overall polarity of a microblog.

- **Does text normalization help analyze sentiments?**

  Although many NLP researchers consider social media specifics as a hindrance and suggest converting them to the standard-language form, other scientists object that a straightforward conversion might loose many important details and consequently worsen classification. Brody and Diakopoulos (2011), for instance, claim that intentional prosodic lengthening of words, such as "*soooooooo strong*" or "*coooollllll*", serves as a vivid indicator of opinionated sentences, so that keeping these elongations in text would result in better predictions. Eisenstein (2013), in part, agrees with these claims by noting that a straightforward replacement of colloquial variants with their standard-language equivalents can considerably shift the original meaning. We admit that the arguments of these authors are correct, but it apparently depends on the magnitude by which non-standard language helps or hampers NLP applications. So, in this work, we would like to test whether text normalization does more harm than good to the analysis of opinions.

- **Can we do better than existing approaches?**

  Of course, simply evaluating existing methods on a new dataset would not be of much novelty and would not accelerate the progress of the research field, therefore, we are going to improve on existing results by suggesting our own solutions to various sentiment objectives.

Outline of this Work

We will answer these questions by proceeding in the following way:

- In Chapter 1, we will give a short introduction to sentiment analysis and make a digression into the history of this field;

- In Chapter 2, we will present the Potsdam Twitter Sentiment Corpus (PotTS), define selection criteria that we used in order to collect tweets for this dataset, describe its annotation scheme and labeling procedure, and also conduct an extensive inter-annotator agreement study, looking for messages that were most difficult to analyze for human experts;

- Afterwards, in Chapter 3, we will turn our attention to the first subsentential sentiment task—sentiment lexicon generation—in which we will compare three major paradigms: dictionary-, corpus-, and word-embedding–based methods, and also propose our own linear-projection solution;

- Chapter 4 will address the problem of fine-grained opinion mining, whose goal is to predict text spans of sentiments, sources, and targets. In particular, we will evaluate three popular approaches to this challenging task: conditional random fields (CRFs), long-short term memory (LSTM), and gated recurrent unit (GRU), checking the effect of various features on the first classifier and estimating the results of the last two systems with different word-embedding types;

- In Chapter 5, we will deal with one of the most prominent sentiment analysis tasks—message-level polarity classification. This time, again, we will juxtapose three main classes of methods: lexicon-based, machine-learning–based, and deep-learning ones, and will try to unite the first and the last of these groups by devising a recurrent neural network with lexicon-based attention;

- Finally, in Chapter 6, we will enhance the proposed system by making it aware of the microblogs' discourse structure. For this purpose, we will let the classifier predict the polarity scores of the elementary discourse units of each tweet and will then unite these scores using novel techniques: latent conditional and conditional-marginalized random fields and Recursive Dirichlet Process.

Chapter 1

Introduction to Sentiment Analysis

Interpersonal communication is not only a way to share objective information with other people but also a vibrant channel to convey one's subjective feelings, impressions, and attitudes. It is, in fact, this latter use that provides a personal touch to our conversations, making them more grasping, more entertaining, and more living. And it is often this use that significantly influences our preferences, decisions, and choices in everyday life. Therefore, a high-quality mining of people's opinions is often as important as retrieval of objective facts.

The field of knowledge that deals with the analysis of emotions, sentiments, evaluations, and attitudes is called *sentiment analysis* (SA) (Liu, 2012). The definition of this discipline, however, much like the definition of the term *sentiment* itself, is neither complete nor universally accepted. The main reasons for this are (i) a frequently blurred boundary between subjective and objective parts of information and (ii) the heterogeneity of the language system itself, to which the SA methods are applied.

The first factor, for instance, makes it difficult to delimit which statements actually belong to the jurisdiction of opinion mining and which ones should be ignored by its systems. A prominent example of such borderline cases are the so-called subjective facts, such as "terrorist attacks" or "anti-cancer drugs," which some people consider as polar terms, while others regard them as objective expressions.

The second factor complicates a precise definition of sentiment analysis because different language levels have their own notions of subjectivity (*e.g.*, a positive word is not the same as a positive text), which in turn necessitate different approaches. Depending on the analyzed linguistic level, researchers typically distinguish three main types of SA:

- *subsentential*, whose task is to determine polarities of single words and find opinions within a sentence,
- *sentential*, which tries to predict the semantic orientation of a statement,

- and, finally, *suprasentential*, which analyzes the polarity of the whole text.

Each of these types has its own specifics, and each of them needs to be addressed with its unique methods. Therefore, speaking of general difficulty of opinion mining for a specific domain is in the same way wrong as judging about the amenability of this domain to the whole natural language processing: one needs to specify a particular task and evaluate it with its own metrics. Thus, to estimate the complexity of sentiment analysis for German Twitter, we will address all three levels of SA: subsentential, sentential, and suprasentential.

1.1 Prehistory of the Field

But before we delve into the depths of contemporary sentiment research, let us first make a digression into the history of this field in order to understand its modern trends and theories better.

Like many other scientific disciplines, opinion mining has emerged from several other areas of research including philosophy, psychology, cognitive sciences, narratology, and linguistics. In *philosophy*, the questions about the nature of emotions, their interaction with human consciousness, and the influence on people's deeds have occupied the minds of many great scholars, starting from Plato and Aristotle. Plato, for instance, argued that the human soul consists of three fundamental parts: the rational, the appetitive, and the passionate (see Plato and Bloom, 1991, Book IV). The last part (the one by which we become angry or fly into a temper) determines our notion of justice by favoring either the rational or the appetitive aspect. Aristotle (1954), the most prominent student of Plato, extended this idea by providing a precise taxonomy of feelings that, in his opinion, constitute the passionate part of the mind.

As noted by de Sousa (2014), the variety and complexity of phenomena covered by the term "emotion" discouraged tidy philosophical ideas and was daunting the researchers for many hundred years since antiquity. A real renaissance of emotional studies happened in the late 19-th century in *psychology* with the introduction of the James-Lange theory (James, 1884; Lange, 2010), which argued that biological processes were the main and only reason for people's subjective opinions. This theory, however, was later criticized by Schachter and Singer (1962), who objected that bodily means alone were insufficient to express the full range of possible feelings, and that cognitive factors were a key determinant of emotional states.

These advances in psychology, reinforced by the nascent appraisal theory (Arnold, 1960), have significantly influenced many other scientific fields including *literary studies* and *linguistics*. Among the most prominent representatives of this direction in the former discipline

were Rorty (1980) and Banfield (1982), who analyzed how opinions were expressed by direct and indirect speech. Wiebe (1990, 1994) adapted Banfield's theory to the needs of *computational linguistics* by proposing an algorithm that identified subjective sentences in text and inferred the main characters of a narrative from such sentences. This work was presumably the first attempt to automatically detect sentiments on the sentential level.

A real breakthrough in the opinion mining field, however, happened with the introduction of the first sufficiently big corpora. Important contributions in this regard were made by Pang and Lee (2004, 2005), who released a dataset of $\approx 2,000$ movie reviews with their star ratings; Hu and Liu (2004), who presented a manually labeled set of Amazon and C|Net product comments; Thomas et al. (2006), who automatically labeled a collection of congressional debates; and, finally, Wiebe et al. (2005), who developed a manually annotated sentiment corpus of 535 news articles.

The availability of these resources has given rise to a plethora of new methods for both subsentential and sentential SA, making opinion mining one of the most challenging and competitive branches of computational linguistics. Fundamental cornerstones in this field have already been set by the works of Pang et al. (2002), Wiebe et al. (2005), Wilson et al. (2005), Breck et al. (2007), Choi and Cardie (2009, 2010), and Socher et al. (2011, 2012). Nevertheless, many challenges of sentiment research, such as domain adaptation or analysis of non-English texts, still pose considerable difficulties.

1.2 Sentiment Analysis of Social Media

One of the main problems that people working on opinion mining are usually confronted with in the first place is the choice of the domain to deal with. Since sentiment analysis is a highly domain-dependent task (see Aue and Gamon, 2005; Blitzer et al., 2007; Li and Zong, 2008)—*i.e.*, systems trained on one text genre can hardly generalize to other linguistic variations—a natural question that arises in this context is which of the domains should be addressed first in this case.

While earlier sentiment works were primarily concerned with narratives (Wiebe, 1990, 1994) or newspaper texts (Wiebe et al., 2003, 2005; Bautin et al., 2008), it soon became clear that social media provide a much more fertile ground for mining people's attitudes. Among the first who tried to extract users' opinions from online forums were Das and Chen (2001). For this purpose, they manually annotated a collection of 500 messages from an economic chat board with three polarity classes (*buy*, *sell*, and *null*) and then trained five different classifiers on these data. Using the trained systems, the authors classified the semantic orientation of the remaining 25,000 forum posts, trying to predict stock prices based on the polarities of these snippets.

Automatic business intelligence has rapidly won the ground in the opinion mining field, with further notable works introduced by Glance et al. (2005), who analyzed users' opinions on Usenet with hand-crafted rules; Antweiler and Frank (2004), who investigated how postings on message boards correlated with stock volatility; Ghose et al. (2007), who examined the effect of opinions on pricing in online marketplaces; and, finally, Turney (2002), who classified Epinions reviews into *recommended* (thumbs up) and *not recommended* (thumbs down) ones based on the pointwise mutual information of their adjectives.

Due to its high commercial impact, sentiment analysis of customer feedback soon became one of the most popular topics in natural language processing. Dave et al. (2003), for example, classified users' comments on Amazon as positive or negative with the help of SVM and Naïve Bayes systems. Hu and Liu (2004) developed a three-stage application that produced concise summaries of positive and negative opinions about each particular product feature. Funk et al. (2008) proposed a supervised SVM classifier that predicted the polarity of product reviews and then used these results in a business intelligence application. Other important contributions were made Popescu and Etzioni (2005), Ding et al. (2009), Wei and Gulla (2010), Mukherjee and Bhattacharyya (2012), etc.

Although opinion analysis of product reviews still plays an important role in e-commerce, the increased popularity of the blogosphere and social networks has motivated many sentiment researchers to shift the focus of their work to these new Web genres. Among the first who followed the new trend were Mishne (2005) and Mishne et al. (2007), who tried to predict the moods of LiveJournal blogs (*e.g.*, *amused*, *tired*, *happy*), achieving 58% accuracy with an SVM classifier. Another SVM system was used by Chesley et al. (2006), who classified users' blogs into positive, negative, and objective ones, outperforming the majority class baseline by 15% with this method. Drawing on these works, Gill et al. (2008) analyzed the agreement of human experts on blogs' moods, finding that the annotators had a much better consensus about feelings that were described in longer blogs.

Speaking of text length, we should certainly say that the inception of the micro-blogging service Twitter in 2006 was a real game changer to the whole opinion mining field. The sudden availability of huge amounts of data, the presence of all possible social and national groups, combined with the uniqueness of the language on this service, have given rise to numerous scientific projects, studies, and publications, which we will briefly summarize in the next section.

1.3 Sentiment Analysis of Twitter

One of the first attempts to automatically classify users' opinions on Twitter was made by Go et al. (2009), who collected a set of 1.6 M microblogs containing emoticons. Considering these

smileys as noisy labels (positive or negative), the authors trained three different classifiers (Naïve Bayes, Maximum Entropy, and SVM), obtaining the best results (0.82 F_1) with the SVM system. Similar approaches were taken by Pak and Paroubek (2010), who performed a three-class prediction with a Naïve Bayes system, and Davidov et al. (2010), who trained a k-NN–like classifier on weakly supervised data. Another way to create a sentiment corpus was proposed by Barbosa and Feng (2010), who analyzed a set of 200,000 microblogs with three publicly available opinion mining services and then used the majority votes of these systems as silver labels for their dataset.

Some time later, Kouloumpis et al. (2011) experimented with the AdaBoost classifier on the noisy collection of Go et al. (2009) and the Edinburgh Twitter corpus of Petrović et al. (2010), coming to the conclusion that microblog-specific features (such as presence of intensifiers, abbreviations, or emoticons) were the most reliable attributes for this classification. Agarwal et al. (2011), however, questioned this finding, arguing that POS-specific polarity features were a better alternative.

As in the case of opinion mining of product reviews, sentiment analysis of Twitter could not go unnoticed by the economic and sociological communities. An attempt to address political issues using this platform was made by Tumasjan et al. (2010), who analyzed users' opinions about German federal elections in 2009 by automatically translating 100,000 tweets into English and subsequently classifying these messages with the proprietary LIWC software (Pennebaker et al., 2007). This study showed that not only sentiments but even the mere numbers of microblogs mentioning political parties strongly correlated with the results of election polls.

Nevertheless, the real rise of interest in this domain happened with the release of the SemEval corpus (Nakov et al., 2013), a collection of 15,000 tweets that have been manually annotated by Amazon mechanical turkers with their message-level polarities and contextual semantic orientations of their polar terms. The best performing system in the first run of the SemEval competition was a supervised SVM classifier of Mohammad et al. (2013), which won three out of four subtasks (message-level classification of SMSs and tweets and contextual polarity prediction of polar terms in Twitter). This solution relied on an extensive set of hand-crafted features, powered by multiple manually- and automatically-generated sentiment lexicons. The authors emphasized the crucial importance of lexical resources, which increased the classification scores by almost 8.5%. Other competing submissions (Becker et al., 2013; Günther and Furrer, 2013; Kökciyan et al., 2013) used a similar approach, but had considerably fewer feature attributes.

The success of this shared task, which had more than 40 participants, motivated the organizers to continue the competition. With slight modifications (addition of new tweets, inclusion of sarcastic microblogs and LiveJournal sentences), they rerun both subtasks in

the following four years (Rosenthal et al., 2014, 2015; Nakov et al., 2016; Rosenthal et al., 2017), attracting more and more competitors every time.[1]

As you can see, despite its relatively short history, sentiment analysis of Twitter has already received much attention from NLP researchers. But, with a few exceptions (*e.g.*, Basile and Nissim, 2013; Bosco et al., 2013; Araque et al., 2015; Cesteros et al., 2015), most of existing works were primarily concerned either with English messages or with automatically translated microblogs. In the following chapters, we will explore whether conclusions that have been drawn from English data (the difficulty of the Twitter domain for automatic SA, the utility of sentiment lexicons, and the need to normalize the text input) apply to German tweets as well.

[1] A detailed overview of these iterations is provided in Chapter 5.

Chapter 2

Sentiment Corpus

A crucial prerequisite for proving any hypotheses in computational linguistics is the existence of sufficiently big manually annotated datasets, on which these conjectures could be tested. Since there were no human-labeled sentiment data for German Twitter that we were aware of at the time of writing this chapter, we decided to create our own corpus, which we will introduce in this part of the thesis.

We begin our introduction by describing the selection criteria and tracking procedure that we used to collect the initial corpus data. After presenting the annotation scheme, we perform an extensive analysis of the inter-annotator agreement. For this purpose, we introduce two new versions of the popular κ metric (Cohen, 1960)—binary and proportional kappa—which have been specifically adjusted to the peculiarities of our annotation task. Using these measures, we check the inter-coder reliability of annotated sentiments, their targets and sources, polar terms and their modifying elements (intensifiers, diminishers, and negations). In the final step, we estimate the correlation between the initial selection rules and the number of labeled elements as well as the difficulty of their annotation.

2.1 Data Collection

A common question that typically arises first when one starts creating a new dataset is which selection criteria should be used in order to collect the initial data. Whereas for low-level NLP applications, such as part-of-speech tagging or syntactic parsing, it typically suffices to define the language domain to sample from (since the phenomena of interest are usually frequent and uniformly spread), for semantically demanding tasks with many diverse ways of expression one also needs to consider various in-domain factors, which might significantly affect the final distribution, making the resulting corpus either utterly sparse or excessively biased.

In order to minimize both of these risks (sparseness and bias), we decided to use a compromise approach by gathering one part of the new dataset from microblogs that were a priori more likely to have sentiments (thus increasing the recall) and sampling the rest of the corpus uniformly at random (thus reducing the bias).

As criteria that could help us get more opinions, we considered topic and form of the tweets, assuming that some subjects, especially social or political issues, would be more amenable to subjective statements. Because we started creating the corpus in spring 2013, obvious choices of opinion-rich topics to us were *the papal conclave*, which took place in March of that year, and *the German federal elections*, which were held in autumn. Since both of these events implied some form of voting, we decided to counterbalance the election specifics by including *general political discussions* as the third subject in our dataset. Finally, to obey the second principle, *i.e.*, to keep the corpus bias low, we sampled the rest of the data from *casual everyday conversations* without any prefiltering.

We collected messages for the first and third groups by tracking German microblogs between March and September 2013 via the public Twitter API[1] with the help of extensive keyword lists describing these topics.[2] Tweets for the second topic (German federal elections) were provided to us by a research group of communication scientists from the University of Münster, who were our cooperation partner in a joint BMBF project "Discourse Analysis in Social Media." Finally, for the fourth category (casual everyday conversations), we used the complete German Twitter snapshot of Scheffler (2014), which includes $\approx 97\%$ of all German microblogs posted in April 2013. This way, we obtained a total of 27.4 M messages, with the snapshot corpus being by far the most prolific source of the data.

In the next step, we divided all tweets of the same topic into three groups based on the following formal criteria:

- We put all messages that contained at least one polar term from the sentiment lexicon of Remus et al. (2010) into the first group;
- Microblogs that did not satisfy the first condition, but had at least one exclamation mark or emoticon were allocated to the second group;
- All remaining microblogs were assigned to the third category.

A detailed breakdown of the resulting distribution across topics and formal groups is given in Table 2.1.

[1] `https://pypi.python.org/pypi/tweetstream`

[2] A full list of tracking keywords is available at `https://github.com/WladimirSidorenko/PotTS/blob/master/docs/tracking_keywords.pdf`.

Topic	Formal Criterion				Sample Keywords
	Polar Terms	Emoticons	Remaining Tweets	Total	
Federal Elections	537,083 (22.38%)	50,567 (2.1%)	1,811,742 (75.5%)	2,399,392	*Abgeordnete* (*representative*), *Regierung* (*government*)
Papal Conclave	7,859 (15.11%)	1,260 (2.42%)	42,879 (82.46%)	51,998	*Papst* (*pope*), *Pabst* (*pobe*)
Political Discussions	10,552 (25.8%)	777 (1.9%)	29,555 (72.29%)	40,884	*Politik* (*politics*), *Minister* (*minister*)
General Conversations	3,201,847 (18.7%)	813,478 (4.7%)	13,088,008 (76.5%)	17,103,333	*den* (*the*), *sie* (*she*)

Table 2.1: Distribution of downloaded messages across topics and formal groups (percentages are given with respect to the total number of tweets pertaining to the given topic)

To create the final corpus, we randomly sampled 666 tweets from each of the three formal classes for each of the four topics, getting a total of 7,992 messages (666 microblogs × 3 formal criteria × 4 topics).

2.2 Annotation Scheme

In the next step, we devised an annotation scheme for our data. To maximally cover all relevant sentiment aspects, we came up with an extensive list of elements that had to be annotated by our experts. This list included:

- `sentiment`s, which were defined as *polar subjective evaluative opinions about people, entities, or events*. According to our definition, a `sentiment` always had to evaluate an entity that was explicitly mentioned in text—the target; and the annotators had to label both the target and its respective evaluative expression with the `sentiment` tag. Apart from tagging the text span, they also had to specify the following attributes of opinions:
  - *`polarity`*, which reflected the attitude of opinion's holder to the evaluated entity. Following Jindal and Liu (2006a,b), we distinguished between *positive*, *negative*, and *comparative* sentiments;
  - *`intensity`*, which showed the emotional strength of an opinion. Possible values for this attribute were: *weak*, *medium*, and *strong*;
  - finally, drawing on the works of Bosco et al. (2013) and Rosenthal et al. (2014), we introduced a special boolean attribute *`sarcasm`* in order to distinguish sarcastically meant statements;

- we specified `targets` as *(real, hypothetical, or collective) entities, properties, or propositions (states or events) evaluated by opinions*. For this item, we introduced the following three attributes:
  - a boolean property ***preferred***, which distinguished entities that were favored in comparisons;
  - a link attribute ***anaphref***, which had to point to the antecedent of a pronominal target;
  - and, finally, another edge feature, ***sentiment-ref***, which had to link `targets` to their respective `sentiments` in the cases when the `target` span was located at intersection of two opinions;
- another important component of `sentiments` were `sources`, which denoted *the immediate author(s) or holder(s) of opinions*. The only property associated with this element was ***sentiment-ref***, which was defined the same way as for `targets`.

To help our annotators identify exact boundaries of these elements, we explicitly asked them to annotate *smallest complete syntactic or discourse-level units*, *i.e.*, noun phrases or sentences with all their grammatical dependents.

A sample tweet analyzed according to this rule is shown in Example 2.2.1.

Example 2.2.1

[[Diese Milliardeneinnahmen]$_{target}$ sind selbst [Schäuble]$_{source}$ peinlich]$_{sentiment}$

[[*These billions of revenue*]$_{target}$ *are embarrassing even for* [*Schäuble*]$_{source}$]$_{sentiment}$

In this message, we assigned the `sentiment` tag to the complete sentence because this grammatical unit is the smallest syntactic constituent that simultaneously includes both the target of the opinion ("Milliardeneinnahmen" [*billions of revenue*]) and its evaluation ("peinlich" [*embarrassing*]). Furthermore, we also labeled the whole noun phrase "diese Milliardeneinnahmen" (*these billions of revenue*), including the demonstrative pronoun "diese" (*these*), as `target`, since this pronoun syntactically depends on the main target word "Milliardeneinnahmen" (*billions of revenue*).

Apart from `sentiments`, `targets`, `sources`, we also asked the annotators to label elements that could significantly affect the intensity and polarity of an opinion. These elements were:

- `polar terms`, which we defined as *words or idioms that had a distinguishable evaluative lexical meaning*. Typical examples of such terms were lexemes or set phrases such as

"ekelhaft" (*disgusting*), "lieben" (*to love*), "Held" (*hero*), "wie die Pest meiden" (*to avoid like the pest*). In contrast to `target`s and `source`s, which only could occur in the presence of a `sentiment`, `polar term`s were independent of other tags and always had to be labeled in the corpus.

The main attributes of this element (*polarity*, *intensity*, and *sarcasm*) largely coincided with the corresponding properties of `sentiment`s, with the only difference that, in the case of `polar term`s, these features had to reflect the lexical meaning of a word without taking into account its context (*i.e.*, *prior* polarity and intensity), whereas for `sentiment`s, they had to show the compositional meaning of the whole opinion (*i.e.*, its *contextual* polarity and intensity).

Besides these common properties, `polar term`s also had their specific attributes: two boolean features (*subjective-fact* and *uncertain*) and a link attribute (*sentiment-ref*). The first feature showed whether a polar term denoted a factual entity with a clear emotional connotation, *e.g.*, "Atombombe" (*A-bomb*) or "Naturschutz" (*nature protection*); the second property signified cases in which the annotators were unsure about their decisions; finally, the last attribute was defined in the same way as it was previously specified for `target`s and `source`s;

- *elements that increased the expressivity and subjective sense of polar terms* had to be labeled as `intensifier`s. Typical examples of such expressions were adverbial modifiers such as "sehr" (*very*), "super" (*super*), "stark" (*strongly*);

- `diminisher`s, on the contrary, were *words or phrases that reduced the strength of a polar term.* Like `intensifier`s, these elements were usually expressed by adverbs, *e.g.*, "weniger" (*less*), "kaum" (*hardly*), "fast" (*almost*).

  Both of these tags (`intensifier`s and `diminisher`s) only had two attributes: a binary feature *degree* with two possible values: *medium* and *strong*; and a link attribute *polar-term-ref*, which connected the modifier to its `polar-term`;

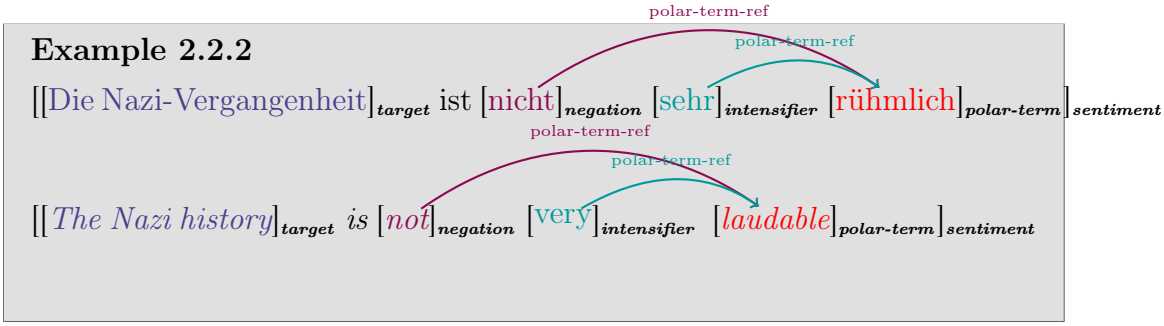
- the final element, `negation`s, was defined as *grammatical or lexical means that reversed the semantic orientation of a polar term.* These were typically represented by the negative particle "nicht" (*not*) or indefinite pronoun "keine" (*no*). The only attribute associated with this tag was a mandatory link *polar-term-ref*.

In contrast to sentiment-level tags, which had to be assigned to syntactic or discourse-level units, `polar term`s and their modifiers were defined as lexemes and, correspondingly, had to mark only single words or set phrases without their grammatical dependents.

A complete tweet annotated with sentiment- and term-level elements is shown in Example 2.2.2. In this case, we again labeled the whole sentence as `sentiment` because only the

main verb-phrase simultaneously covers both the evaluated target ("Die Nazi-Vergangenheit" [*The Nazi history*]) and its respective polar expression ("nicht sehr rühmlich" [*not very laudable*]). The boundaries of `sentiment` and `target` are determined on the syntactic level, spanning the whole clause in the former case and including the complete noun phrase in the latter. The polarity of the opinion is set to *negative*. The polar term "rühmlich" (*laudable*), its intensifier "sehr" (*very*), and negation "nicht" (*not*), on the other hand, only mark single words. The polarity of the term, *i.e.*, its primary semantic orientation without the context, is *positive*.

Example 2.2.2



A more detailed description of all annotation elements and their possible attributes is given in the original annotation guidelines in Appendix A of this thesis.

2.3 Annotation Tool and Format

For annotating the collected data, we used `MMAX2`, a freely available text-markup tool.[3] Because this program uses a token-oriented stand-off format, where all annotated spans are stored in a separate file and only refer to the ids of words in the original text, we first had to split all corpus messages into tokens. To this end, we applied a minimally modified version of Christopher Potts' social media tokenizer,[4] which had been slightly adjusted to the peculiarities of the German spelling (we allowed for the capitalized form of common nouns, *e.g.*, "Freude" [*joy*], and the period at the end of ordinal numbers, *e.g.*, "7." [*7th*]).

To ease the annotation process and minimize possible data loss, we split the corpus into 80 smaller project files with 99–109 tweets each. In each such file, we put microblogs pertaining to the same topic, ensuring an equal proportion of formal groups.

[3] `http://mmax2.sourceforge.net/`

[4] `http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py`

2.4 Inter-Annotator Agreement Metrics

For estimating the inter-annotator agreement (IAA), we adopted the popular κ metric (Cohen, 1960). Following the standard practice, we computed this term as:

$$\kappa = \frac{p_o - p_c}{1 - p_c},$$

where p_o denotes the observed agreement, and p_c stands for the agreement by chance. We estimated the observed reliability in the normal way as the ratio of tokens with matching annotations to the total number of tokens:

$$p_o = \frac{T - A_1 + M_1 - A_2 + M_2}{T},$$

where T represents the total token count, A_1 and A_2 are the numbers of tokens annotated with the given class by the first and second annotators respectively, and the M terms mean the numbers of tokens with matching annotations. As usual, we computed the chance agreement p_c as:

$$p_c = c_1 \times c_2 + (1.0 - c_1) \times (1.0 - c_2).$$

where c_1 and c_2 are the proportions of tokens annotated with the given class in the first and second annotations, respectively, *i.e.*, $c_1 = \frac{A_1}{T}$ and $c_2 = \frac{A_2}{T}$.

Two questions that arose during this computation though were (i) whether tokens belonging to multiple overlapping annotation spans of the same class had to be counted several times in one annotation when computing the A scores (for instance, whether we had to count the words "dieses" [*this*], "schöne" [*nice*], and "Buch" [*book*] in Example 2.4.1 twice as sentiments when computing A_1 and A_2), and (ii) whether we had to assume that two annotated spans from different experts agreed on all of their tokens if these spans had at least one word in common (*e.g.*, whether we had to consider the annotation of the token "Mein" [*My*] in the example as matching, regarding that the rest of the corresponding `sentiment`s agreed).

Example 2.4.1

Annotation 1:

[Mein Vater hasst [dieses schöne Buch]$_{sentiment}$.]$_{sentiment}$

[*My father hates* [*this nice book*]$_{sentiment}$.]$_{sentiment}$

Annotation 2:

Mein [Vater hasst [dieses schöne Buch]$_{sentiment}$.]$_{sentiment}$

My [*father hates* [*this nice book*]$_{sentiment}$.]$_{sentiment}$

To address these issues, we introduced two different agreement metrics—*binary* and *proportional* kappa. With the former variant, we counted tokens belonging to overlapping annotation spans of the same class multiple times (*i.e.*, A_1 and A_2 would amount to 10 and

9, respectively, in the above tweet) and considered all tokens belonging to the given annotated element as matching if this span agreed with the annotation from the other expert on at least one token (*i.e.*, M_1 and M_2 would have the same values as A_1 and A_2 in this case). With the latter metric, every labeled token was counted only once (*i.e.*, the numbers of labeled words in the first and second annotations would be 7 and 6, respectively), and we only calculated the actual number of tokens with matching labels when computing the M scores (*i.e.*, both M_1 and M_2 would be equal to 6). The final value of the binary kappa in Example 2.4.1 would consequently run up to 1.0 because this metric would consider both annotations as perfectly matching, since every labeled `sentiment` agreed with the other annotation on at least one token. The proportional kappa, however, would be equal to 0.0, since this metric would emphasize the fact that the observed reliability p_o is the same as the agreement by chance p_c, and would therefore deem both labelings as fortuitous.

2.5 Annotation Procedure

After defining the agreement metrics, we finally let our experts annotate the data. The annotation procedure was performed in three steps:

- At the beginning, both annotators labeled one half of the corpus after only minimal training. Unfortunately, their mutual agreement at this stage was relatively low, reaching only 31.21% proportional-κ for `sentiment`s;
- In the second step, in order to improve the inter-rater reliability, we automatically determined all differences between the two annotations and highlighted non-matching tokens with a separate class of tags. Then, we let the experts resolve these discrepancies by either correcting their own decisions or rejecting the variants of the other coder. As in the previous stage, we allowed the annotators to consult their supervisor (the author of this thesis), also updating the FAQ section of the guidelines based on their questions, but did not let them communicate with each other directly. This adjudication step significantly improved all annotations: The agreement on `sentiment`s increased by 30.73%, reaching 61.94%. Similar effects were observed for `target`s, `source`s, `polar term`s, and their modifiers;
- After resolving all differences, our assistants proceeded with the annotation of remaining files. Working completely independently, one of the experts has annotated 78.8% of the corpus, whereas the second annotator has labeled the complete dataset.

2.6 Evaluation

2.6.1 Initial Annotation Stage

The agreement results of the initial annotation stage are shown in Table 2.2.

Element	Binary κ					Proportional κ				
	M_1	A_1	M_2	A_2	κ	M_1	A_1	M_2	A_2	κ
Sentiment	4,215	7,070	3,484	9,827	**38.05**	3,269	6,812	3,269	9,796	**31.21**
Target	1,103	1,943	1,217	4,162	**35.48**	898	1,905	898	4,148	**26.85**
Source	159	445	156	456	**34.53**	153	439	153	456	**33.75**
Polar Term	1,951	2,854	2,029	3,188	**64.29**	1,902	2,851	1,902	3,180	**61.36**
Intensifier	57	101	59	123	**51.71**	57	101	57	123	**50.81**
Diminisher	3	10	3	8	**33.32**	3	10	3	8	**33.32**
Negation	21	63	21	83	**28.69**	21	63	21	83	**28.69**

Table 2.2: Inter-annotator agreement after the initial annotation stage ($M1$ – number of tokens with matching labels in the first annotation, $A1$ – total number of tokens labeled with that class in the first annotation, $M2$ – number of tokens with matching labels in the second annotation, $A2$ – total number of tokens labeled with that class in the second annotation)

As we can see from the table, the inter-rater reliability of `sentiment`s strongly correlates with the inter-annotator agreement on `target`s and `source`s, setting an upper bound for these elements in the binary-κ case. With the proportional metric, however, both `sentiment`s and `target`s show worse results than `source`s: 31.21% and 26.85% versus 33.75%. We explain this difference by the fact, that `sentiment`s and `target`s are typically represented by syntactic or discourse-level constituents (noun phrases or clauses) and, even though the experts agreed on the presence of these elements more often (as suggested by the binary-κ metric), reaching a consensus about the exact boundaries of these elements was still a challenging task for them despite an explicit clarification of this problem in the annotation guidelines; `source`s, on the other hand, are usually expressed by pronouns, which rarely accept syntactic attributes, so that their boundaries were easier to determine. Nevertheless, even with the binary metric, the agreement of all sentiment-level elements is significantly below the 40% threshold, which means only a slight reliability according to the Landis and Koch scale (Landis and Koch, 1977).

A different situation is observed for `polar terms` and `intensifiers`. The inter-annotator agreement of these elements is above 50%, for both κ-measures. Obviously, defining these entities as lexical units has significantly eased the detection of their boundaries. This effect becomes even more evident if we look at `diminisher`s and `negation`s, where the A and M scores are absolutely identical for both metrics. It means that both annotators always agreed

on the boundaries of these elements when they agreed on their presence. Unfortunately, due to a rather small number of these tags in the corpus (with only 3 cases of `diminishers` and 21 cases of `negations`), the overall agreement on these labels is relatively small too, amounting to 33.32% and 28.69%, respectively.

2.6.2 Adjudication Step

Since these scores were unacceptable for running further experiments, we decided to revise diverging annotations by letting our experts recheck each other's decisions. As we can

Element	Binary κ					Proportional κ				
	M_1	A_1	M_2	A_2	κ	M_1	A_1	M_2	A_2	κ
Sentiment	8,198	8,530	8,260	14,034	**67.92**	7,435	8,243	7,435	13,714	**61.94**
Target	3,088	3,407	2,814	5,303	**65.66**	2,554	3,326	2,554	5,212	**57.27**
Source	573	690	545	837	**72.91**	539	676	539	833	**71.12**
Polar Term	3,164	3,298	3,261	4,134	**85.68**	3,097	3,290	3,097	4,121	**82.64**
Intensifier	111	219	113	180	**56.01**	111	219	111	180	**55.51**
Diminisher	9	16	10	16	**59.37**	9	16	9	15	**58.05**
Negation	68	84	67	140	**60.21**	67	83	67	140	**60.03**

Table 2.3: Inter-annotator agreement after the adjudication step ($M1$ – number of tokens with matching labels in the first annotation, $A1$ – total number of labeled tokens in the first annotation, $M2$ – number of tokens with matching labels in the second annotation, $A2$ – total number of labeled tokens in the second annotation)

see from the results in Table 2.3, this procedure has significantly improved the inter-rater reliability of all annotated elements: the binary scores of `sentiments` and `targets` increased by 29.87% and 30.18%, respectively. An even greater improvement is observed for `sources`, whose binary kappa improved by remarkable 38.38%. A similar tendency applies to the proportional metric, where the agreement of `sentiments` gained 30.73%, reaching 61.94%. Likewise, the reliability of opinion targets and holders improved by 30.42% and 37.37%, running up to 57.27% and 71.12%.

As in the previous step, the highest agreement scores are attained by `polar terms`, whose reliability notably surpasses the 80% benchmark, which means an almost perfect agreement. Interestingly enough, only 193 out of 3,290 terms annotated by the first expert did not match the labelings of the second annotator. Another interesting observation is that the difference between the binary and proportional scores of `polar terms` only amounts to 3.04%, which implies that the assistants could unproblematically determine the boundaries of these elements in most of the cases.

Somewhat surprisingly, the agreement of `intensifiers` improved notably less. A closer look at the annotated cases revealed that the majority of their disagreements stemmed from

different takes of exclamation marks: the first expert ignored these punctuation marks, whereas the second annotator considered them as valid intensifying elements. Nevertheless, even despite these diverging interpretations, the reliability of `intensifier`s is above 55%, which means a moderate level.

2.6.3 Final Annotation Stage

After ensuring that our annotators could reach an acceptable quality of annotation, we eventually let them label the remaining part of the data. The agreement results of the final stage computed on the files annotated by both experts are given in Table 2.4.

Element	**Binary κ**					**Proportional κ**				
	M_1	A_1	M_2	A_2	κ	M_1	A_1	M_2	A_2	κ
Sentiment	14,748	15,929	14,969	26,047	**65.03**	13,316	15,375	13,316	25,352	**58.82**
Target	5,765	6,629	5,292	9,852	**64.76**	4,789	6,462	4,789	9,659	**56.61**
Source	966	1,207	910	1,619	**65.99**	898	1,180	898	1,604	**64.1**
Polar Term	5,574	5,989	5,659	7,419	**82.83**	5,441	5,977	5,441	7,395	**80.29**
Intensifier	192	432	194	338	**49.97**	192	432	192	338	**49.71**
Diminisher	16	30	17	34	**51.55**	16	30	16	33	**50.78**
Negation	111	132	110	243	**58.87**	110	131	110	242	**58.92**

Table 2.4: Inter-annotator agreement of the final corpus
($M1$ – number of tokens with matching labels in the first annotation, $A1$ – total number of labeled tokens in the first annotation, $M2$ – number of tokens with matching labels in the second annotation, $A2$ – total number of labeled tokens in the second annotation)

This time, we can observe a slight decrease of the results: the proportional score for `sentiment`s dropped by 3.12%, whereas the agreement on `target`s was more persistent and lost only 0.66%, going down to 56.61%. The most dramatic changes occurred for `source`s, whose proportional value deteriorated by notable 7.02%, sinking to 64.1%. Nonetheless, the average proportional agreement of all these elements is around 60.5%, which is almost twice as high as the mean reliability achieved in the first stage.

As before, the scores of `polar term`s are in the ballpark of almost perfect results. Their modifying elements, however, show a decrease: the agreement of `intensifier`s deteriorated by 5.8%, sinking to 49.71% proportional kappa. A similar situation is observed for `diminisher`s, whose kappa worsened from 58.05% to 50.78%. The best persistence in this regard is shown by `negation`s, where the quality dropped by only 1.11%, which can be considered as a very good result, regarding the small number of these elements in the corpus.

In general, we can see that the reliability of all elements in the final dataset is at least

moderate, with `polar terms` being the most reliably annotated elements ($\kappa_p = 80.29\%$), and `intensifiers` setting a lower bound on the agreement ($\kappa_p = 49.71\%$).

2.6.4 Qualitative Analysis

In order to understand the reasons for remaining conflicts, we decided to have a closer look at the diverging cases. A sample sentence with different analyses of `sentiments` is shown in Example 2.6.1:

Example 2.6.1

Annotation 1:

@TinaPannes immerhin ist die #afd nicht dabei ☺

Annotation 2:

@TinaPannes [[immerhin ist die #afd nicht dabei]$_{target}$ ☺]$_{sentiment}$

@TinaPannes [[*anyway the #afd is not there*]$_{target}$ ☺]$_{sentiment}$

In this tweet, the first annotator obviously overlooked the emoticon ☺ at the end of the message, whereas the second expert correctly recognized it as an evaluation of the previous sentence. Because the first assistant did not label any `sentiment` at all, she also automatically disagreed on the `target` of this opinion.

A much rarer case of diverging `target` annotations was when both experts actually marked a `sentiment` span. An example of such situation is shown in the following message:

Example 2.6.2

Annotation 1:

[Koalition wirft der SPD [Blockadehaltung]$_{target}$ vor]$_{sentiment}$

[*Coalition accuses the SPD of* [*blocking politics*]$_{target}$]$_{sentiment}$

Annotation 2:

[Koalition wirft [der SPD]$_{target}$ Blockadehaltung vor]$_{sentiment}$

[*Coalition accuses* [*the SPD*]$_{target}$ *of blocking politics*]$_{sentiment}$

In this sentence, the first expert considered *blocking politics* as the main object of criticism, whereas the second annotator regarded the political party accused of such behavior as sentiment's target. In our opinion, both of these interpretations are correct and, ideally,

two `sentiments` had to be labeled in this message: one with the target "Blockadehaltung" (*blocking politics*) and another one with the target "die SPD" (*the SPD*).

Although our annotators were much more consistent about the analysis of `polar terms`, we still decided to have a look at disagreeing labels of these elements. A sample case of differently annotated `polar terms` is given in Example 2.6.3

Example 2.6.3

Annotation 1:

Syrien vor dem Angriff—bringen diese Bomben den Frieden?

Syria facing an attack—will these bombs bring peace?

Annotation 2:

Syrien vor dem [Angriff]$_{polar\text{-}term}$—bringen diese [Bomben]$_{polar\text{-}term}$ den [Frieden]$_{polar\text{-}term}$?

Syria facing an [*attack*]$_{polar\text{-}term}$*—will these* [*bombs*]$_{polar\text{-}term}$ *bring* [*peace*]$_{polar\text{-}term}$*?*

The obvious reason for the misclassifications in this message is the notorious subjective facts: As you can see, the first assistant ignored the words "Angriff" (*attack*), "Bombe" (*bomb*), and "Frieden" (*peace*), while the second annotator considered them as polar items. We should, however, admit that this difference is partially due to the adjudication procedure that we used in step two; because at the initial stage, our experts had had opposite preferences regarding these entities: the first annotator had labeled these terms, whereas the second assistant had usually skipped them. During the revision, however, both assistants have changed their minds after looking at the decisions of the other linguist. Therefore, one needs to keep in mind the risk of mutual concession when applying the adjudication method in the future.

2.6.5 Attributes Agreement

In order to see whether our annotators also agreed on the attributes of the tags, we estimated the Cohen's kappa for the polarity and the Krippendorff's alpha (Krippendorff, 2007) for the intensity of matching `sentiments` and `polar terms`. The reason for choosing two different metrics in this case is that ***`polarity`*** is a categorical feature, whose value takes on one of the predefined classes (*positive*, *negative*, or *comparison*), whereas ***`intensity`*** is an ordinal attribute, whose value can range on a scale from zero (*weak*) to two (*strong*). Since disagreements that are further apart on the scale need to be penalized more strongly than small divergences, we decided to use the α-measure for this attribute, as it explicitly addresses this problem.

Element	Polarity κ	Intensity α
Sentiment	58.8	73.54
Polar Term	87.12	78.79

Table 2.5: Inter-annotator agreement on polarity and intensity of sentiments and polar terms

As we can see from the results in Table 2.5, reaching a consensus about the polarities of `polar terms` is a much easier task than agreeing on the semantic orientation of `sentiments`. As in Example 2.6.1, one of the main reasons for these disagreements is opinions containing emoticons, especially in the cases when the polarity of the smiley contradicts the polarity of the preceding text, *e.g.*, "Ich hasse die Piratenpartei ☺" (*I hate the Pirate Party* ☺).

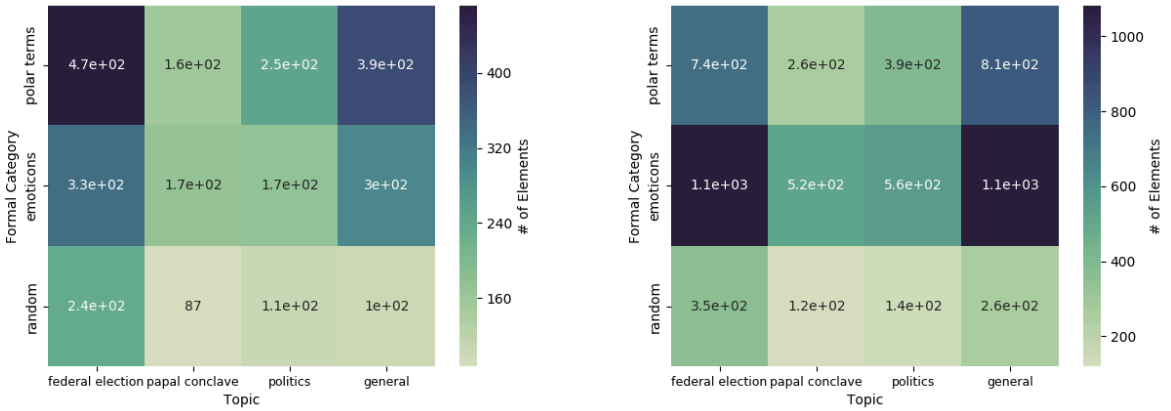
Interestingly enough, the inter-rater agreement on the intensity of `sentiments` ($\alpha = 73.54$) is notably higher than the corresponding score for their polarity ($\kappa = 58.8$), although the opposite situation is observed for `polar terms`, whose α-value (78.79) is almost ten percent lower than κ (87.12). This means that the annotators could easily determine the semantic orientation of a single word, but had difficulties with agreeing on the strength of its meaning. Vice versa, when dealing with targeted opinions, they usually assigned the *medium* intensity to most `sentiments`, but could disagree on the polarity of these statements.

For the sake of completeness, we compared these results with the scores obtained on the MPQA dataset (see Wilson, 2007, pp. 38, 80). The average α-agreement on the intensity of direct subjective and objective speech events (a rough counterpart of our `sentiments`) in this corpus was around 79%; the corresponding results for the intensity of expressive subjective elements (`polar terms` in our case), however, were much worse, amounting to only 46%, even though the κ-value for their polarity run up to 72%. Hence, the reliability of annotated attributes in our corpus still outperforms the respective agreement in MPQA on almost all aspects except for sentiment intensity.

2.6.6 Effect of the Selection Criteria

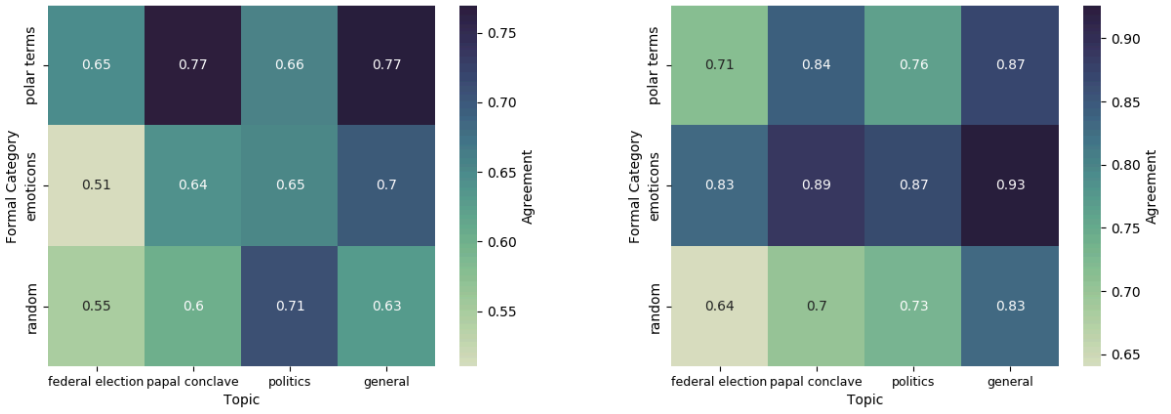
Finally, in order to check how the selection criteria that we applied initially when sampling the corpus data affected the resulting distribution of `sentiments` and `polar terms` in the final dataset, we plotted the frequencies and agreement scores of these elements across topics and formal groups, and present these statistics in Figures 2.1 and 2.2.

As we can see from the plots, topics and form clearly affect both the number of opinions and the difficulty of their interpretation. According to Figure 2.1, the greatest number of `sentiments` occur in tweets pertaining to the federal elections and in messages representing casual everyday conversations. A similar tendency is observed for `polar terms`, but in this



(a) `Sentiments` *(b)* `Polar Terms`

Figure 2.1: Distribution of sentiments and polar terms across topics and formal groups



(a) `Sentiments` *(b)* `Polar Terms`

Figure 2.2: Inter-annotator agreement on sentiments and polar terms across topics and formal groups

case, the form of the microblogs seems to have more impact on the elements' distribution than their topics.

Regarding the inter-annotator agreement, we can see that `sentiment`s and `polar term`s are most reliably annotated in messages from the German Twitter Snapshot. Moreover, the former elements are apparently easiest to annotate in tweets that were preselected using a sentiment lexicon, whereas `polar term`s are easiest to analyze in microblogs that contain emoticons.

To confirm the correlation between the topics and formal groups on the one hand and the number and reliability of `sentiment`s and `polar term`s on the other hand, we computed the correlation coefficients (ρ) of these factors, considering each particular topic and formal group as a binary variable and measuring the association of this variable with the number and agreement of annotated elements.

Selection Criteria	**Correlation Coefficients**			
	Sentiment		**Polar Term**	
	# of elements	agreement	# of elements	agreement
Topical Groups				
Federal Elections	**0.312**	0.169	0.356	0.289
Papal Conclave	0.149	0.124	0.182	0.264
Political Discussions	0.195	0.148	0.218	0.244
General Conversations	0.183	**0.19**	**0.372**	**0.452**
Formal Categories				
Polar Terms	**0.445**	**0.352**	0.38	0.301
Emoticons	0.127	0.096	**0.47**	**0.615**
Random	0.216	0.134	0.143	0.138

Table 2.6: Correlation coefficients of topics and formal selection criteria with the number and agreement scores of sentiments and polar terms

As we can see from the results in Table 2.6, both criteria (topics and form) have a positive correlation with the number of annotated elements and their reliability. The highest ρ-score for `sentiment`s is achieved by tweets describing federal elections and messages containing polar terms, where it amounts to 0.312 and 0.445, respectively. A slightly different situation is observed for `polar term`s: the highest scores for this element both in terms of the number of annotated items and their reliability are achieved by casual everyday conversations and tweets that contain emoticons.

2.7 Summary and Conclusions

Now that we have reached the end of the second chapter, we would like to remind the reader that in this part of the thesis we have presented the Potsdam Twitter Sentiment Corpus (PotTS), a collection of 7,992 German microblogs that had been manually annotated by two human experts with sentiments, targets, sources, polar terms, and their modifying elements.

We obtained initial data for this corpus by tracking tweets about German federal elections, papal conclave, discussions of general political topics, and casual everyday conversations between spring and autumn 2013. Afterwards, we grouped these messages into three classes (tweets containing polar terms, microblogs containing exclamation marks or emoticons, and the rest of the messages) and randomly sampled 666 posts from each of these classes for each topic.

The annotation process was performed in three steps: first, the annotators labeled one half of the data after minimal training; then, we automatically highlighted their divergent analyses and asked them to resolve these differences; finally, our assistants continued with the analysis of the remaining files.

To estimate the inter-rater reliability, we introduced two modified versions of the established κ-metric—binary and proportional kappa—which differ in the way how they treat overlapping annotations and partial matches. Using these measures, we estimated the inter-annotator agreement of our experts at different stages of their work. This study showed that, initially, our assistants could hardly agree on the mere notion of targeted opinions, but their disagreements could be resolved with the help of the adjudication procedure that we applied in step two. Despite a small drop of the IAA scores in the final stage, all κ-values still remained at the level of at least moderate reliability.

Finally, we demonstrated that our initial selection criteria had a strong impact on the number and agreement of annotated sentiments and polar terms, with tweets about federal elections and messages without prefiltered topics being the most prolific sources of these elements.

That way, we not only contributed to the inventory of available sentiment and social-media resources for German but also provided new insights into different sampling methods that could be used to create an opinion dataset and described the consequences of applying these methods in practice. A detailed inter-annotator agreement study showed precisely which topics yield most subjective opinions (elections and casual conversations) and which groups of messages are especially difficult to annotate (tweets containing emoticons and microblogs without polar terms or emoticons). In the next step, we are going to check whether our dataset can also serve as a basis for building and evaluating automatic opinion mining applications.

Chapter 3

Sentiment Lexicons

The first avenue that we are going to explore using the obtained data is automatic prediction of polar terms. For this purpose, we will first evaluate existing German sentiment lexicons on our corpus. Since most of these resources, however, were created semi-automatically by translating English polarity lists and then manually post-editing and expanding these translations, we will also look whether methods that were used to produce original English lexicons would yield comparable results when applied to German data directly. Finally, we will analyze whether one of the most popular areas of research in contemporary computational linguistics, distributed vector representations of words (Mikolov et al., 2013), can produce better polarity lists than previous approaches. In the concluding step, we will investigate the effect of different hyper-parameters and seed sets on these systems, summarizing and concluding our findings in the last part of this chapter.

3.1 Data

As *development set* for our experiments, we will use *400 randomly selected tweets* annotated by the first expert. As gold *test set* for evaluating the lexicons, we will utilize the *complete corpus labeled by the second linguist*. These test data comprise a total of 6,040 positive and 3,055 negative terms. But because many of these expressions represent emoticons, which, on the one hand, are a priori absent in common lexical taxonomies such as WordNet (Miller, 1995; Miller and Fellbaum, 2007) or GermaNet (Hamp and Feldweg, 1997) and therefore not amenable to methods that rely on these resources, but on the other hand, can be easily captured by regular expressions, we decided to exclude non-alphabetic smileys altogether from our study. This left us with a set of 3,459 positive and 2,755 negative labeled terms (1,738 and 1,943 unique expressions, respectively), whose κ-agreement run up to 59%.

3.2 Evaluation Metrics

An important question that needs to be addressed before we proceed with the experiments is which evaluation metrics we should use to measure the quality of sentiment lexicons. Usually, this quality is estimated either *intrinsically* (by taking a lexicon in isolation and immediately assessing its accuracy) or *extrinsically* (by considering the lexicon within the scope of a bigger application, *e.g.*, a supervised classifier that uses lexicon's entries as features).

Traditionally, intrinsic evaluation of English polarity lists amounts to comparing these resources with the General Inquirer lexicon (GI; Stone et al., 1966), a manually compiled list of 11,895 words annotated with their semantic categories. For this purpose, researchers usually take the intersection of the two sets and estimate the percentage of matches in which automatically induced polar terms have the same polarity as corresponding GI entries. This evaluation, however, is somewhat problematic: First of all, it is not easily transferable to other languages because even a manual translation of GI is not guaranteed to cover all language- and domain-specific polar expressions. Second, since it only considers the intersection of the two sets, it does not penalize for low recall, so that a polarity list that consists of just two terms $good^+$ and bad^- will always have the highest possible score, often surpassing other lexicons with a greater number of entries. Finally, such comparison does not account for polysemy. As a result, an ambiguous word only one of whose (possibly rare) senses is subjective will always be ranked the same as an obvious polar term.

Unfortunately, an extrinsic evaluation does not always provide a remedy in this case because different extrinsic applications might yield different results, and a polarity list that performs best with one system can produce fairly low scores with another application.

Instead of using these methods, we decided to evaluate sentiment lexicons directly on our corpus by comparing their entries with the annotated polar terms, since such approach would allow us to solve at least three of the aforementioned problems, namely, (i) it would account for recall, (ii) it would distinguish between different senses of polysemous words,[1] and (iii) it would preclude intermediate modules that could artificially improve or worsen the results.

In particular, in order to evaluate a lexicon on our dataset, we represent this polarity list as a case-insensitive trie (Knuth, 1998, pp. 492–512) and compare this trie with the original and lemmatized[2] corpus tokens, successively matching them from left to right. We consider a match as correct if a lexicon term completely agrees with the (original or lemmatized) tokens of an annotated `polar term` and has the same polarity as the labeled element. All corpus

[1] The annotators of our corpus were asked to label a polar term iff the actual sense of this term in the given context was polar.

[2] All lemmatizations in our experiments were performed using the TreeTagger of Schmid (1995).

tokens that are not marked as `polar terms` in the corpus are considered as gold neutral words; similarly, all terms that are absent from the lexicon, but present in the corpus are assumed to have a predicted neutral polarity.

3.3 Semi-Automatic Lexicons

Using this metric, we first estimated the quality of existing German polarity lists:

- **German Polarity Clues** (GPC; Waltinger, 2010), which contains 10,141 polar terms from the English sentiment lexicons Subjectivity Clues (Wilson et al., 2005) and SentiSpin (Takamura et al., 2005) that were automatically translated into German and then manually revised by the author. Apart from that, Waltinger also manually enriched these translations with their frequent synonyms and 290 negated phrases;[3]
- **SentiWS** (SWS; Remus et al., 2010), which includes 1,818 positively and 1,650 negatively connoted terms along with their part-of-speech tags and inflections, which results in a total of 32,734 word forms. As in the previous case, the authors obtained the initial entries for their resource by translating an English polarity list (the General Inquirer lexicon) and then manually correcting these translations. In addition to this, they expanded the translated set with words and phrases that frequently co-occurred with positive and negative seed terms in a corpus of 10,200 customer reviews or in the German Collocation Dictionary (Quasthoff, 2010);[4]
- and, finally, the only the lexicon that was not obtained through translation—the **Zurich Polarity List** (ZPL; Clematide and Klenner, 2010), which features 8,000 subjective entries extracted from GermaNet synsets (Hamp and Feldweg, 1997). These synsets had been manually annotated by human experts with their prior polarities. Since the authors, however, found the number of polar adjectives obtained this way to be insufficient for their classification experiments, they automatically enriched this lexicon with more attributive terms, using the collocation method of Hatzivassiloglou and McKeown (1997).

For our evaluation, we tested each of the three lexicons in isolation, and also evaluated their union and intersection in order to check for possible "synergy" effects. The results of this computation are shown in Table 3.1.

[3] In our experiments, we excluded the auxiliary words "aus" (*from*), "der" (*the*), "keine" (*no*), "nicht" (*not*), "sein" (*to be*), "was" (*what*), and "wer" (*who*) with their inflection forms from the German Polarity Clues, because these entries significantly worsened the evaluation results.

[4] Unfortunately, the authors do not provide a breakdown of how many terms were obtained through translation and how many of them were added during the expansion.

Lexicon	Positive Expressions			Negative Expressions			Neutral Terms			Macro F_1	Micro F_1
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1		
GPC	0.209	0.535	0.301	0.195	0.466	0.275	0.983	0.923	0.952	0.509	0.906
SWS	0.335	0.435	0.379	0.484	0.344	**0.402**	0.977	0.975	0.976	0.586	0.952
ZPL	0.411	0.424	0.417	0.38	0.352	0.366	0.977	0.979	0.978	0.587	0.955
GPC ∩ SWS ∩ ZPL	**0.527**	0.372	**0.436**	**0.618**	0.244	0.35	0.973	**0.99**	**0.982**	**0.589**	**0.964**
GPC ∪ SWS ∪ ZPL	0.202	**0.562**	0.297	0.195	**0.532**	0.286	**0.985**	0.917	0.95	0.51	0.901

Table 3.1: Evaluation of semi-automatic German sentiment lexicons
GPC — German Polarity Clues, SWS — SentiWS, ZPL — Zurich Polarity List

As we can see from the table, the intersection of all three polarity lists achieves the best results for the positive and neutral classes, and also attains the highest macro- and micro-averaged F_1-scores. One of the main reasons for this success is a relatively high precision of this set for all polarities except neutral, where the intersection is outperformed by the union of three lexicons. The last fact is also not surprising, as the union has the highest recall of positive and negative terms. Among all compared lexicons, the results of the Zurich Polarity List come closest to the scores of the intersected set: its macro-F_1 is lower by 0.002, and its micro-average is less by 0.009. The second-best lexicon is SentiWS, which reaches the highest F_1-score for the negative class, but has a lower precision of positive entries. Finally, German Polarity Clues is the least reliable sentiment resource, which is also mainly due to the low precision of its polar terms.

3.4 Automatic Lexicons

A natural question that arises upon evaluation of existing semi-automatic lexicons is how well fully automatic methods can perform in comparison with these resources. According to Liu (2012, p. 79), most automatic sentiment lexicon generation (SLG) algorithms can be grouped into two main classes: dictionary- and corpus-based ones. The former systems induce polarity lists from monolingual thesauri or lexical databases such as the Macquarie Dictionary (Bernard, 1986) or WordNet (Miller, 1995). A clear advantage of these methods is their relatively high precision, as they operate on manually annotated, carefully verified data. At the same time, this precision might come at the price of reduced recall, especially in domains whose language changes very rapidly and where new terms are coined in a flash. In contrast to this, corpus-based systems operate directly on unlabeled in-domain texts and, consequently, have access to all neologisms; but the downside of these approaches is that they often have to deal with extremely noisy input and might therefore have low accuracy. Since it was unclear to us which of these pros and cons would have a stronger influence on the net results, we decided to reimplement the most popular algorithms from both of these

groups and evaluate them on our corpus.

3.4.1 Dictionary-Based Methods

The presumably first SLG system that inferred a sentiment lexicon from a lexical database was proposed by Hu and Liu (2004). In their work on automatic classification of customer reviews, the authors automatically compiled a list of polar adjectives (which were supposed to be the most relevant part of speech for mining people's opinions) by taking a set of seed terms with known semantic orientations and propagating the polarity scores of these seeds to their WordNet synonyms. A similar procedure was also applied to antonyms, but the polarity values were reversed in this case. This expansion continued until no more adjectives could be reached via synonymy-antonymy links.

Blair-Goldensohn et al. (2008) refined this approach by considering polarity scores of all WordNet terms as a single vector $\vec{v}$; the values of all negative seeds in this vector were set to -1, and the scores of all positive seed terms were fixed to $+1$. To derive their polarity list, the authors multiplied $\vec{v}$ with an adjacency matrix A. Each cell a_{ij} in this matrix was set to $\lambda = 0.2$, if there was a synonymy link between synsets i and j, and to $-\lambda$, if these synsets were antonymous to each other. By performing this multiplication multiple times and storing the results of the previous iterations in the $\vec{v}$ vector, the authors ensured that all polarity scores were propagated transitively through the network, decaying by a constant factor (λ) as the length of the paths starting from the original seeds increased.

With various modifications, the core idea of Hu and Liu (2004) was adopted by almost all dictionary-based works: For example, Kim and Hovy (2004, 2006) estimated the probability of word w belonging to polarity class $c \in \{\text{positive, negative, neutral}\}$ as:

$$P(c|w) = P(c)P(w|c) = P(c)\frac{\sum_{i=1}^{n} count(syn_i, c)}{count(c)},$$

where $P(c)$ is the prior probability of that class (estimated as the number of words belonging to class c divided by the total number of words); $count(syn_i, c)$ denotes the number of times a seed term with polarity c appeared in a synset of w; and $count(c)$ means the total number of synsets that contain seeds with this polarity. Using this formula, the authors successively expanded their initial set of 34 adjectives and 44 verbs to a list of 18,192 polar terms.

Another popular dictionary-based resource, SentiWordNet, was created by Esuli and Sebastiani (2006a), who enriched a small set of positive and negative seed adjectives with their WordNet synonyms and antonyms in $k \in \{0, 2, 4, 6\}$ iterations, considering the rest of the terms as neutral if they did not have a subjective tag in the General Inquirer lexicon. In each of these k steps, the authors optimized two ternary classifiers (Rocchio and SVM)

that used tf-idf–vectors of synset glosses as features. Afterwards, they predicted polarity scores for all WordNet synsets using an ensemble of all trained classifiers.

Graph-based SLG algorithms were proposed by Rao and Ravichandran (2009), who experimented with three different methods:

- *deterministic min-cut*, in which the authors propagated the polarity values of seeds to their WordNet synonyms and hypernyms and then determined a minimum cut between the polarity clusters using the algorithm of Blum and Chawla (2001);
- since this approach, however, always partitioned the graph in the same way even if there were multiple possible splits with the same cost, the authors also proposed a *randomized* version of this method, in which they randomly perturbed edge weights;
- finally, they compared both min-cut systems with the *label propagation algorithm* of Zhu and Ghahramani (2002), which can be considered as a probabilistic variant of Blair-Goldensohn et al.'s approach.

Further notable contributions to dictionary-based methods were made by Mohammad et al. (2009), who compiled an initial set of polar terms by using antonymous morphological patterns (*e.g.*, *logical* — *illogical*, *honest* — *dishonest*, *happy* — *unhappy*) and then expanded this set with the help of the Macquarie Thesaurus (Bernard, 1986); Awadallah and Radev (2010), who adopted a random walk approach, estimating word's polarity as a difference between the average number of steps a random walker had to make in order to reach a seed term from the positive or negative set; and Dragut et al. (2010), who computed words' polarities using manually specified inference rules.

For our experiments, we reimplemented the approaches of Hu and Liu (2004), Blair-Goldensohn et al. (2008), Kim and Hovy (2004, 2006), Esuli and Sebastiani (2006a), Rao and Ravichandran (2009), and Awadallah and Radev (2010), and applied these methods to GermaNet[5] (Hamp and Feldweg, 1997), the German equivalent of the WordNet taxonomy.

In order to make this comparison more fair, we used the same set of initial seeds for all tested methods. For this purpose, we translated the list of 14 polar English adjectives proposed by Turney and Littman (2003) ($good^+$, $nice^+$, $excellent^+$, $positive^+$, $fortunate^+$, $correct^+$, $superior^+$, bad^-, $nasty^-$, $poor^-$, $negative^-$, $unfortunate^-$, $wrong^-$, and $inferior^-$) into German, getting a total of 20 terms (10 positive and 10 negative adjectives) due to multiple possible translations of the same words. Furthermore, to settle the differences between binary and ternary approaches (*i.e.*, methods that only distinguished between positive

[5]Throughout our experiments, we will use GermaNet Version 9.

and negative terms and systems that could also predict the neutral class), we extended the translated seeds with 10 neutral adjectives (*neutral*^{0}, *objective*^{0}, *technical*^{0}, *chemical*^{0}, *physical*^{0}, *material*^{0}, *bodily*^{0}, *financial*^{0}, *theoretical*^{0}, and *practical*^{0}), letting all classifiers work in the ternary mode. Finally, since several algorithms had different takes of synonymous relations (*e.g.*, Hu and Liu only considered two words as synonyms if they appeared in the same synset, whereas Esuli and Sebastiani, Rao and Ravichandran, and Awadallah and Radev also considered hypernyms and hyponyms as valid links for polarity propagation), we decided to unify this aspect as well. To this end, we established an edge between any two terms that appeared in the same synset, and also linked all words whose synsets were connected via `has_participle`, `has_pertainym`, `has_hyponym`, `entails`, or `is_entailed_by` relations. We intentionally ignored relations `has_hypernym` and `is_related_to`, because hypernyms were not guaranteed to preserve the polarity of their children (*e.g.*, "bewertungsspezifisch" [*appraisal-specific*] is a neutral term in contrast to its immediate hyponyms "gut" [*good*] and "schlecht" [*bad*]), and `is_related_to` could connect both synonyms and antonyms of the same term (*e.g.*, this relation holds between words "Form" [*shape*] and "unförmig" [*misshapen*], but at the same time, it also connects noun "Dame" [*lady*] to its derived adjective "damenhaft" [*ladylike*]).

We fine-tuned the hyper-parameters of all approaches by using grid search and optimizing the macro-averaged F_1-score on the development set. In particular, instead of waiting for the full convergence of the eigenvector in the approach of Blair-Goldensohn et al. (2008), we constrained the maximum number of multiplications to five. Our experiments showed that this limitation had a crucial impact on the quality of the resulting polarity list (*e.g.*, after five multiplications, the average precision of its positive terms amounted to 0.499, reaching an average F_1-score of 0.26 for this class; after ten more iterations though, this precision decreased dramatically to 0.043, pulling the F_1-score down to 0.078). Furthermore, we limited the maximum number of iterations in the label-propagation method of Rao and Ravichandran (2009) to 300, although the effect of this setting was much weaker than in the previous case (by comparison, the scores achieved after 30 runs differed only by a few hundredths from the results obtained after 300 iterations). Finally, in the method of Awadallah and Radev (2010), we allowed for seven simultaneous walkers with a maximum number of 17 steps each, considering a word as polar if more than a half of these walkers agreed on the same polarity class.

As we can see from the results in Table 3.2, the scores of all automatic systems are significantly lower than the values achieved by semi-automatic lexicons. The best macro-averaged F_1-result for all three classes (0.479) is attained by the method of Blair-Goldensohn et al. (2008), which is still 0.11 points below the highest score obtained by the intersection of GPC, SentiWS, and the Zurich Polarity List. Moreover, in general, the situation with dictionary-based lexicons is more complicated than in the case of manually curated polarity

Lexicon	# of Terms	Positive Expressions			Negative Expressions			Neutral Terms			Macro	Micro
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1	F_1
Seed Set	20	**0.771**	0.102	0.18	0.568	0.017	0.033	0.963	**0.999**	**0.981**	0.398	**0.962**
HL	5,745	0.161	0.266	0.2	0.2	0.133	0.16	0.969	0.96	0.965	0.442	0.93
BG	1,895	0.503	0.232	**0.318**	0.285	0.093	0.14	0.968	0.991	0.979	**0.479**	0.959
KH	356	0.716	0.159	0.261	0.269	0.044	0.076	0.965	0.997	**0.981**	0.439	**0.962**
ES	39,181	0.042	**0.564**	0.078	0.033	**0.255**	0.059	**0.981**	0.689	0.81	0.315	0.644
RR_{mincut}	8,060	0.07	0.422	0.12	0.216	0.073	0.109	0.972	0.873	0.92	0.383	0.849
$RR_{lbl\text{-}prop}$	1,105	0.567	0.176	0.269	**0.571**	0.046	0.085	0.965	0.997	**0.981**	0.445	**0.962**
AR	23	0.768	0.1	0.176	0.568	0.017	0.033	0.963	**0.999**	**0.981**	0.397	**0.962**
HL ∩ BG ∩ RR_{lbl}	752	0.601	0.165	0.259	0.567	0.045	0.084	0.965	0.997	**0.981**	0.441	**0.962**
HL ∪ BG ∪ RR_{lbl}	6,258	0.166	0.288	0.21	0.191	0.146	**0.165**	0.97	0.958	0.964	0.446	0.929

Table 3.2: Results of dictionary-based approaches
HL — Hu and Liu (2004), BG — Blair-Goldensohn et al. (2008), KH — Kim and Hovy (2004), ES — Esuli and Sebastiani (2006a), RR — Rao and Ravichandran (2009), AR — Awadallah and Radev (2010)

lists, as every system demonstrates a better score on only one metric, but fails to convincingly outperform its competitors on several (let alone all) aspects. Nevertheless, we still can notice at least the following main trends:

- the method of Esuli and Sebastiani (2006a) achieves the highest recall of positive and negative terms, but these entries have a very low precision;
- simultaneously five approaches attain the same best F_1-results for the neutral class, which, in turn, leads to the best micro-averaged F_1-scores for these systems;
- and, finally, the solution of Blair-Goldensohn et al. (2008) achieves the highest macro-averaged F_1 despite a rather low recall of negative expressions.

3.4.2 Corpus-Based Methods

An alternative way of generating polarity lists is provided by corpus-based approaches. In contrast to dictionary-based methods, these systems operate immediately on raw texts and are therefore virtually independent of any manually annotated resources.

A pioneering work on these algorithms was done by Hatzivassiloglou and McKeown (1997). Assuming that coordinately conjoined attributes would typically have the same semantic orientation, these authors trained a supervised logistic classifier that predicted the degree of dissimilarity between two co-occurring adjectives. Afterwards, they constructed a word collocation graph, drawing a link between any two adjectives that appeared in the same coordinate pair, and using predicted dissimilarity score between these words as the respective

edge weight. In the final stage, Hatzivassiloglou and McKeown (1997) partitioned this graph into two clusters and assigned the positive label to the bigger part.

An attempt to unite dictionary- and corpus-based methods was made by Takamura et al. (2005), who adopted the Ising spin model from statistical mechanics, considering words found in WordNet, the Wall Street Journal, and the Brown corpus as electrons in a ferromagnetic lattice. The authors established a link between any two electrons whose terms appeared in the same WordNet synset or coordinately conjoined pair in the corpora. In the final step, they approximated the most probable orientation of all spins in this graph, considering these orientations as polarity scores of the respective terms.

Another way of creating a sentiment lexicon was proposed by Turney and Littman (2003), who induced a list of polar terms by computing the difference between their point-wise mutual information (PMI) with the positive and negative seeds. In particular, the authors estimated the polarity score of word w as:

$$\text{SO-A}(w) = \sum_{w_p \in \mathcal{P}} PMI(w, w_p) - \sum_{w_n \in \mathcal{N}} PMI(w, w_n),$$

where $\mathcal{P}$ represents the set of all positive seeds; $\mathcal{N}$ denotes the collection of known negative words; and PMI is computed as a log-ratio $PMI(w, w_x) = \log_2 \frac{p(w,w_x)}{p(w)p(w_x)}$. The joint probability $p(w, w_x)$ in the last term was calculated as the number of hits returned by the AltaVista search engine for the query "w NEAR w_x" divided by the total number of documents in the search index.

This method was later successfully adapted to Twitter by Kiritchenko et al. (2014), who harnessed the corpus of Go et al. (2009) and an additional set of 775,000 tweets to create two sentiment lexicons, Sentiment140 and Hashtag Sentiment Base, using frequent emoticons as seeds for the first lexicons and taking common emotional hashtags such as "#joy", "#excitement", "#fear" as seed terms for the second list.

Another Twitter-specific approach, which also relied on the corpus of Go et al. (2009), was presented by Severyn and Moschitti (2015a). To derive their lexicon, the authors trained an SVM classifier that used token n-grams as features and then included n-grams with the greatest learned feature weights into their final polarity list.

Graphical methods for corpus-based SLG were advocated by Velikovich et al. (2010) and Feng et al. (2011). The former work adapted the label-propagation algorithm of Rao and Ravichandran (2009) by replacing the average of all incident scores for a potential subjective term with their maximum value. The latter approach induced a sentiment lexicon using two popular techniques from information retrieval, PageRank (Brin and Page, 1998) and HITS (Kleinberg, 1999).

For our experiments, we reimplemented the approaches of Takamura et al. (2005), Velikovich et al. (2010), Kiritchenko et al. (2014) and Severyn and Moschitti (2015b), and applied these methods to the German Twitter Snapshot (Scheffler, 2014), a collection of 24 M German microblogs, which we previously used for sampling one part of our sentiment corpus.

We normalized all messages of this snapshot with the rule-based normalization pipeline of Sidarenka et al. (2013), which will be described in more detail in the next chapter, and lemmatized all tokens with the TreeTagger of Schmid (1995). Afterwards, we constructed a collocation graph from all normalized lemmas that appeared at least four times in the snapshot. For the method of Takamura et al. (2005), we additionally used GermaNet in order to add more links between electrons. As in the previous experiments, all hyperparameters (including the size of the lexicons) were fine-tuned on the development set by maximizing the macro-averaged F_1-score on these data.

The results of this evaluation are presented in Table 3.3.

Lexicon	**# of Terms**	**Positive Expressions**			**Negative Expressions**			**Neutral Terms**			**Macro**	**Micro**
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1	F_1
Seed Set	20	**0.771**	0.102	0.18	**0.568**	0.017	0.033	0.963	**0.999**	**0.981**	0.398	**0.962**
TKM	920	0.646	**0.134**	**0.221**	0.565	**0.029**	**0.055**	**0.964**	0.998	**0.981**	**0.419**	**0.962**
VEL	60	0.764	0.102	0.18	**0.568**	0.017	0.033	0.963	0.999	0.98	0.398	**0.962**
KIR	320	0.386	0.106	0.166	**0.568**	0.017	0.033	0.963	0.996	0.979	0.393	0.959
SEV	60	0.68	0.102	0.177	**0.568**	0.017	0.033	0.963	**0.999**	**0.981**	0.397	**0.962**
TKM ∩ VEL ∩ SEV	20	**0.771**	0.102	0.18	**0.568**	0.017	0.033	0.963	**0.999**	**0.981**	0.398	**0.962**
TKM ∪ VEL ∪ SEV	1,020	0.593	**0.134**	0.218	0.565	**0.029**	**0.055**	**0.964**	0.998	0.98	0.418	**0.962**

Table 3.3: Results of corpus-based approaches
TKM — Takamura et al. (2005), VEL — Velikovich et al. (2010), KIR — Kiritchenko et al. (2014), SEV — Severyn and Moschitti (2015b)

This time, we can observe a clear superiority of the system of Takamura et al. (2005), which not only achieves the best recall and F_1 for the positive and negative classes but also yields the highest micro- and macro-averaged results for all three polarities. The sizes and the scores of other lexicons, however, are much smaller than the cardinalities and the results of the Takamura et al.'s polarity list. Moreover, these lexicons can hardly outperform the original seed set on the negative class.

Because the last result was somewhat unexpected, we decided to investigate the reasons for potential problems in these systems. A closer look at their learning curves revealed that the macro-averaged F_1-values on the development data rapidly decreased from the very beginning of their work. Since we considered the lexicon size as one of the parameters, we rapidly stopped populating these lists. As a consequence, only few highest ranked terms

(all of which were positive) were included into the final resource. As it turned out the main reason for this degradation was the ambiguity of the seed terms: While adapting the original seed list of Turney and Littman (2003) to German, we translated the English word "correct" as "richtig." This German word, however, also has another reading—*real* (as in "ein richtiges Spiel" [*a real game*] or "ein richtiger Rennwagen" [*a real sports car*]), which was much more frequent in the analyzed snapshot and typically appeared in a negative context, *e.g.*, "ein richtiger Bombenanschlag" (*a real bomb attack*) or "ein richtiger Terrorist" (*a real terrorist*). As a consequence, methods that relied on weak supervision had to deal with extremely unbalanced training data (716,210 positive instances versus 92,592 negative ones) and got stuck in a local optimum from the very beginning of their training.

3.4.3 NWE-Based Methods

Finally, the last group of methods that we are going to explore in this chapter are algorithms that operate on distributed vector representations of words (neural word embeddings [NWEs]). First introduced by Bengio et al. (2003) and significantly improved by Collobert et al. (2011) and Mikolov et al. (2013), NWEs had a great "tsunami"-like effect on many downstream NLP applications (Manning, 2015). Unfortunately, these advances have largely bypassed the generation of sentiment lexicons, up to a few exceptions introduced by the works of Tang et al. (2014a) and Vo and Zhang (2016). In the former approach, the authors used a large collection of weakly labeled tweets in order to learn hybrid word embeddings. In contrast to standard word2vec vectors (Mikolov et al., 2013) and purely task-specific representations (Collobert et al., 2011), such embeddings were optimized with respect to both objectives—predicting the occurrence of nearby words and classifying the overall polarity of a message. Using these hybrid vectors, Tang et al. trained a one-layer feed-forward neural network that predicted the polarity of a microblog, and subsequently applied this classifier separately to each word embedding, considering the predicted value as polarity score for the respective term. In contrast to this approach, Vo and Zhang immediately optimized two-dimensional task-specific embeddings, and regarded the two dimensions of these learned vectors as positive and negative scores of corresponding words.

In order to evaluate these systems, we reimplemented both methods, extending them to three-way classification (positive, negative, and neutral), and applied them to weakly labeled snapshot tweets.

Apart from these solutions, we also came up with the following alternative ways of generating polarity lists from neural word embeddings:

- the method of the nearest centroids,

- k-NN clustering;
- principal component analysis (PCA),
- and a new linear-projection algorithm.

In the first method, we computed the centroids of the positive, negative, and neutral clusters by taking the arithmetic mean of the respective seed-term vectors and then assigned word w to the polarity group whose centroid was closest to the embedding of that word. We considered the distance to the cluster center as the respective polarity score for that word and sorted the resulting sentiment lexicon in ascending order of these values.

A similar procedure was used in k-NN, where we first determined k seed vectors that were closest to the embedding of word w and then allocated this word to the polarity class whose seeds were nearest and appeared most frequently in w's neighborhood.

A different technique was used for the principal component analysis. After normally decomposing the embedding matrix $E \in \mathbb{R}^{d \times |V|}$ (with $d = 300$ denoting the dimension of word vectors, and $|V|$ representing the vocabulary size) into singular components:

$$E = U \Sigma V^T,$$

we looked for a row axis $u_{\text{subj}} \in U$ that maximized the distance between the embeddings of polar (positive or negative) and neutral seeds projected on that line. In the same way, we determined a polarity axis u_{pol} that maximized the distance between projected positive and negative embeddings. After finding both axes, we projected all word vectors on these two lines, considering the distances between these projections and lines' origins as the respective subjectivity and polarity scores. The pseudo-code of this approach is shown in Algorithm 1.

Since PCA, however, was not guaranteed to find the optimum projection axes (the orthogonal bases U and V in SVD are typically computed using the covariance matrix of E, and might therefore not reflect semantic orientations of words, if terms with opposite polarities occur in similar contexts), we devised our own *linear projection method*, in which we explicitly encoded the above objective: Namely, given two sets of vectors with opposite semantic orientations (let us denote the set of positive vectors as $\mathcal{P} = \{\vec{p}_{+_1}, \ldots, \vec{p}_{+_m}\}$ and the set of negative embeddings as $\mathcal{N} = \{\vec{p}_{-_1}, \ldots, \vec{p}_{-_n}\}$), we were looking for a line $\vec{b}$ that maximized the distance between the projections of embeddings from these sets on that line, i.e.:

$$\begin{aligned} \vec{b} &= \text{argmax} \frac{1}{2} \sum_{\vec{p}_+} \sum_{\vec{p}_-} \left\| \frac{\vec{b} \cdot \vec{p}_+}{\vec{b}^2} \vec{b} - \frac{\vec{b} \cdot \vec{p}_-}{\vec{b}^2} \vec{b} \right\|^2 \\ &= \text{argmax} \frac{1}{2} \sum_{\vec{p}_+} \sum_{\vec{p}_-} \left\| \frac{\vec{b} \cdot (\vec{p}_+ - \vec{p}_-)}{\vec{b}^2} \vec{b} \right\|^2, \end{aligned} \tag{3.1}$$

Algorithm 1 Sentiment lexicon generation with the PCA algorithm

function ExpandPCA($\mathcal{P}, \mathcal{N}, \mathcal{O}, \mathtt{E}$) ▷ $\mathcal{P}$ – indices of positive terms, ▷ $\mathcal{N}$ – indices of negative terms, ▷ $\mathcal{O}$ – indices of objective terms, $\mathtt{E}$ – embedding matrix
  $\mathtt{U}, \Sigma, \mathtt{V}^\top \leftarrow \text{SVD}(\mathtt{E})$; ▷ obtain singular components of $\mathtt{E}$
  $\mathtt{E}' \leftarrow \left(\mathtt{E}^\top \cdot \mathtt{U}\right)^\top$; ▷ project $\mathtt{E}$ onto the eigenvectors of its row space
  $\mathcal{S} \leftarrow \mathcal{P} \cup \mathcal{N}$; ▷ get the set of subjective terms
  $\mathtt{u}_{subj}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}} \leftarrow \text{FindMeanAxis}(\mathtt{E}', \mathcal{S}, \mathcal{O})$; ▷ find subjectivity axis
  $\mathtt{u}_{pol}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}} \leftarrow \text{FindAxis}(\mathtt{E}', \mathcal{P}, \mathcal{N})$; ▷ find polarity axis
  return $\text{ComputePolScores}(\mathtt{E}', \mathcal{S} \cup \mathcal{O}, \mathtt{u}_{subj}, \mathtt{u}_{pol}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}})$;
end function

function FindMeanAxis($\mathtt{E}, \mathcal{S}_1, \mathcal{S}_2$)
  $\mu_1 \leftarrow 0$; $\mu_2 \leftarrow 0$; $\mathtt{axis} \leftarrow 0$; $\mathtt{max\_dist} \leftarrow 0$;
  for $\mathtt{i} \leftarrow 1$; $\mathtt{i} <= \text{nrows}(\mathtt{E})$; $\mathtt{i} \leftarrow \mathtt{i} + 1$ **do**
    $\mathtt{e} \leftarrow \mathtt{M[i]}$;
    $\mathtt{dist} \leftarrow \sum_{j_1 \in \mathcal{S}_1, j_2 \in \mathcal{S}_2} |\mathtt{e}[j_1] - \mathtt{e}[j_2]|$;
    if $\mathtt{dist} > \mathtt{max\_dist}$ **then**
      $\mathtt{axis} \leftarrow \mathtt{i}$; $\mathtt{max\_dist} \leftarrow \mathtt{dist}$;
      $\mu_1 \leftarrow \frac{\sum_{j_1 \in \mathcal{S}_1} \mathtt{e}[j_1]}{|\mathcal{S}_1|}$; $\mu_2 \leftarrow \frac{\sum_{j_2 \in \mathcal{S}_2} \mathtt{e}[j_2]}{|\mathcal{S}_2|}$;
    end if
  end for
  return axis, μ_1, μ_2;
end function

function ComputePolScores($\mathtt{E}, \mathcal{S}, \mathtt{u}_{subj}, \mathtt{u}_{pol}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}}$)
  $\mathtt{scores} \leftarrow []$;
  $\mathtt{O}_{subj} \leftarrow \mu_{\mathcal{O}} + \frac{\mu_{\mathcal{S}} - \mu_{\mathcal{O}}}{2}$; ▷ Compute the origin of the subjectivity axis.
  $\mathtt{max\_score}_{subj} \leftarrow \max\left(\{|\mathtt{n}[\mathtt{u}_{subj}] - \mathtt{O}_{subj}|| \forall \mathtt{n} \in \text{cols}(\mathtt{E})\}\right)$;
  $\mathtt{O}_{pol} \leftarrow \mu_{\mathcal{N}} + \frac{\mu_{\mathcal{P}} - \mu_{\mathcal{N}}}{2}$; ▷ Compute the origin of the polarity axis.
  $\mathtt{max\_score}_{pol} \leftarrow \max\left(\{|\mathtt{n}[\mathtt{u}_{pol}] - \mathtt{O}_{pol}|| \forall \mathtt{n} \in \text{cols}(\mathtt{M})\}\right)$;
  for $\mathtt{i} \leftarrow 1$; $\mathtt{i} <= \text{ncols}(\mathtt{M})$; $\mathtt{i} \leftarrow \mathtt{i} + 1$ **do**
    if $\mathtt{i} \in \mathcal{S}$ **then**
      continue; ▷ known seeds will be added later by default
    end if
    $\mathtt{n} \leftarrow \mathtt{M[:, i]}^\top$; ▷ assign i-th column to $\mathtt{n}$
    if $|\mathtt{n}[\mathtt{u}_{subj}] - \mu_{\mathcal{O}}| > |\mathtt{n}[\mathtt{u}_{subj}] - \mu_{\mathcal{S}}|$ **then** ▷ if the i-th word is subjective
      $\mathtt{score}_{subj} \leftarrow 1 + \frac{|\mathtt{n}[\mathtt{u}_{subj}] - \mathtt{O}_{subj}|}{\mathtt{max\_score}_{subj}}$;

else
  $\texttt{score}_{subj} \leftarrow 1 - \frac{|\texttt{n}[\texttt{u}_{subj}] - \texttt{O}_{subj}|}{\texttt{max\_score}_{subj}}$;
end if
if $|\texttt{n}[\texttt{u}_{pol}] - \mu_{\mathcal{N}}| > |\texttt{n}[\texttt{u}_{pol}] - \mu_{\mathcal{P}}|$ **then** ▷ if the i-th word is positive
  $\texttt{polarity} \leftarrow$ positive;
else
  $\texttt{polarity} \leftarrow$ negative;
end if
$\texttt{score}_{pol} \leftarrow \frac{|\texttt{n}[\texttt{u}_{pol}] - \texttt{O}_{pol}|}{\texttt{max\_score}_{pol}}$;
append(scores, (i, polarity, $\frac{1}{\texttt{score}_{\text{subj}} + \texttt{score}_{pol}}$)); ▷ The total score is inversed
▷ as we sort the resulting polarity list in the ascending order.
end for
return scores;
end function

where $\frac{\vec{b}\cdot\vec{p}_+}{\vec{b}^2}\vec{b}$ is the projection of a word embedding with the positive polarity on line $\vec{b}$, and $\frac{\vec{b}\cdot\vec{p}_-}{\vec{b}^2}\vec{b}$ is the respective projection of a negative seed term. Considering the argmax argument in Expression 3.1 as our objective function f, we computed the gradient of f with respect to $\vec{b}$ as follows:

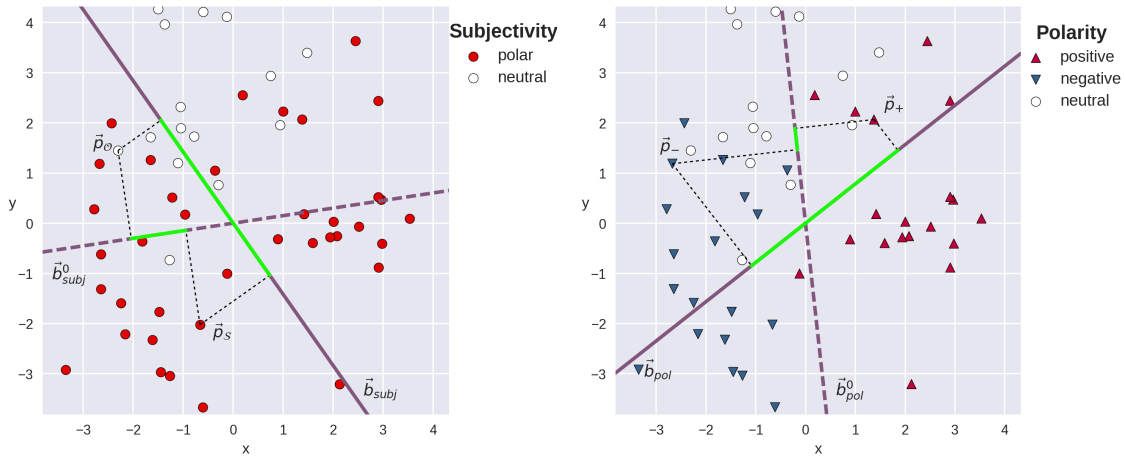
$$\nabla_{\vec{b}} f = \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \left(\Delta - \gamma \vec{b} \right), \tag{3.2}$$

where Δ stands for the difference between the positive and negative vectors $\vec{p}_+$ and $\vec{p}_-$: $\Delta := \vec{p}_+ - \vec{p}_-$; and γ denotes the dot product of this difference with vector $\vec{b}$: $\gamma := \Delta \cdot \vec{b}$.[6] With this gradient, we then optimized $\vec{b}$ using gradient ascent until we reached a maximum of function f.[7]

As in PCA, we first used this approach to determine an optimal subjectivity axis $\vec{b}_{subj}$, which maximized the distance between the sets of polar and neutral embeddings. After finding this axis and projecting on it all remaining word vectors, we classified all words into polar and neutral ones, depending on whether their projections appeared closer to the mean of the former or latter set. We then computed the subjectivity score for polar terms as $s^{pol}_{subj} = 1 + \frac{\delta^i_{subj}}{\delta^{\max}_{subj}}$, where δ^i_{subj} stands for the distance between the projection of the i-th term and the origin O_{subj}, and $\delta^{\max}_{subj}$ means the maximum such distance observed in the data. Similarly, we estimated these values for neutral items as $s^{neut}_{subj} = 1 - \frac{\delta^i_{subj}}{\delta^{\max}_{subj}}$.

[6] The details of this gradient computation are given in Appendix B.

[7] Since function f is neither convex nor concave, we can only speak about *a* maximum. We hypothesize, however, based on our experiments and preliminary calculations, that this local maximum will simultaneously be the global one because the optimized projection vector will have two possible solutions, which will lie on the same line, but point to the opposite directions.



(a) Subjectivity line *(b) Polarity line*

Figure 3.1: Visualization of the linear projection method in the two-dimensional vector space with unnormalized vectors
($\vec{b}^0_{subj}$ and $\vec{b}^0_{pol}$ – initial guesses of the subjectivity and polarity lines; $\vec{b}_{subj}$ and $\vec{b}_{pol}$ – optimal projection vectors; the distances between the projections of sample seeds with opposite semantic orientations ($\vec{p}_{\mathcal{S}}$ vs. $\vec{p}_{\mathcal{O}}$ and $\vec{p}_+$ vs. $\vec{p}_-$ respectively) on these lines are highlighted in green)

In the same way, we estimated the polarity score for the i-th word by first finding a polarity line $\vec{b}_{pol}$ and then computing the respective score as $s_{pol} = 1 + \frac{\delta^i_{pol}}{\delta^{max}_{pol}}$.

In the final step, we united both values s_{subj} and s_{pol} into a single score $s = \frac{1}{s_{subj}+s_{pol}}$ and sorted the resulting polarity list in ascending order of these unified scores. The pseudo-code of our approach is given in Algorithm 2, and a visualization of the line optimization step (in the two-dimensional vector space) is shown in Figure 3.1.

We applied all methods to word2vec embeddings, which had been previously learned on the snapshot data, normalizing the length and scaling the means of these vectors before passing them to our algorithms. The results of all systems are shown in Table 3.4.

As we can see from the table, linear projection not only outperforms all other NWE-based systems in terms of the micro-averaged F_1-score but also surpasses the results of dictionary-, corpus-based, and semi-automatic lexicons, being only 0.1 percent below the overall best F_1-value achieved by the intersection of SentiWS, German Polarity Clues, and Zurich Polarity List. Our method also achieves the second-best macro-averaged F_1-result, being outmatched by k-NN. The third-best micro-averaged F_1 is attained by the approach of Tang et al. (2014a), which, however, suffers from low precision of its positive entries. The results of the remaining systems are, unfortunately, even lower and hardly improve on the scores of the initial seed set.

Algorithm 2 Sentiment lexicon generation with the linear projection algorithm

function ExpandLinProj($\mathcal{P}, \mathcal{N}, \mathcal{O}, \mathtt{E}$) ▷ $\mathcal{P}$ – indices of positive terms, ▷ $\mathcal{N}$ – indices of negative terms, ▷ $\mathcal{O}$ – indices of objective terms, $\mathtt{E}$ – embedding matrix
  $\mathcal{S} \leftarrow \mathcal{P} \cup \mathcal{N}$; ▷ get the set of subjective terms
  $\mathtt{proj}_{subj}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}} \leftarrow$ Project$(\mathcal{S}, \mathcal{O}, \mathtt{E})$;
  $\mathtt{proj}_{pol}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}} \leftarrow$ Project$(\mathcal{P}, \mathcal{N}, \mathtt{E})$;
  return ComputePolScores$(\mathtt{E}, \mathcal{S} \cup \mathcal{O}, \mathtt{proj}_{subj}, \mathtt{proj}_{pol}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}})$;
end function

function Project$(\mathcal{S}_1, \mathcal{S}_2, \mathtt{E})$
  $\mathtt{projs} \leftarrow []$, $\mu_1 \leftarrow \vec{0}$; $\mu_2 \leftarrow \vec{0}$;
  $\vec{b} \leftarrow$ FindAxis$(\mathcal{S}_1, \mathcal{S}_2)$; ▷ Determine the optimum polarity/subjectivity axis
  for $\mathtt{i} \leftarrow 1$; $\mathtt{i} <=$ nrows$(\mathtt{E})$; $\mathtt{i} \leftarrow \mathtt{i} + 1$ **do**
    $\vec{p} \leftarrow \vec{b} * \mathtt{E}[i] \cdot \vec{b}$; ▷ Project i-th embedding onto the axis
    append$(\mathtt{projs}, \vec{p})$;
    if $\mathtt{j} \in \mathcal{S}_1$ **then** ▷ Update means of the polarity/subjectivity classes
      $\mu_1 \leftarrow \mu_1 + \vec{p}$;
    else
      if $\mathtt{k} \in \mathcal{S}_2$ **then**
        $\mu_2 \leftarrow \mu_2 + \vec{p}$;
      end if
    end if
  end for
  return $\mathtt{projs}$, $\frac{\mu_1}{|\mathcal{S}_1|}$; $\frac{\mu_2}{|\mathcal{S}_2|}$;
end function

function FindAxis$(\mathcal{S}_1, \mathcal{S}_2, \mathtt{E})$
  $\vec{b} \leftarrow \vec{1}$; $\mathtt{prev\_dist} \leftarrow \infty$;
  for $\mathtt{i} \leftarrow 1$; $\mathtt{i} \leqslant \mathtt{MAX\_ITERS}$; $\mathtt{i} \leftarrow \mathtt{i} + 1$ **do**
    $\vec{b} \leftarrow \frac{\vec{b}}{\|\vec{b}\|}$; ▷ Ensure the projection line has a unit length
    $\mathtt{dist} \leftarrow 0$;
    for $\mathtt{j} \in \mathcal{S}_1$ **do**
      $\vec{p_1} \leftarrow \vec{b} * \mathtt{E}[j] \cdot \vec{b}$; ▷ Project a seed term from the first set
      for $\mathtt{k} \in \mathcal{S}_2$ **do**
        $\vec{p_2} \leftarrow \vec{b} * \mathtt{E}[k] \cdot \vec{b}$; ▷ Project a seed term from the second set
        $\mathtt{dist} \leftarrow \mathtt{dist} + \|\vec{p_1} - \vec{p_2}\|$; ▷ Accumulate the distance between the two projections
      end for
    end for

35: **if** $\texttt{prev\_dist} \neq \infty$ **and** $\texttt{dist} - \texttt{prev\_dist} < \epsilon$ **then**
36: **break**; ▷ Break if the convergence criterion was reached
37: **end if**
38: $\texttt{prev\_dist} \leftarrow \texttt{dist}$;
39: $\vec{b} \leftarrow \vec{b} + \alpha * \nabla\vec{b}$; ▷ Optimize the projection line
40: **end for**
41: **return** $\vec{b}$;
42: **end function**

43: **function** ComputePolScores($\texttt{E}, \mathcal{S}, \texttt{projections}_{subj}, \texttt{projections}_{pol}, \mu_{\mathcal{S}}, \mu_{\mathcal{O}}, \mu_{\mathcal{P}}, \mu_{\mathcal{N}}$)
44: $\texttt{scores} \leftarrow []$;
45: $\texttt{O}_{subj} \leftarrow \mu_{\mathcal{O}} + \frac{\mu_{\mathcal{S}} - \mu_{\mathcal{O}}}{2}$; ▷ Compute the origin of the subjectivity axis.
46: $\texttt{max\_score}_{subj} \leftarrow \max\left(\{|\texttt{p}_{subj} - \texttt{O}_{subj}||\forall \texttt{p}_{subj} \in \texttt{projections}_{subj}\}\right)$;
47: $\texttt{O}_{pol} \leftarrow \mu_{\mathcal{N}} + \frac{\mu_{\mathcal{P}} - \mu_{\mathcal{N}}}{2}$; ▷ Compute the origin of the polarity axis.
48: $\texttt{max\_score}_{pol} \leftarrow \max\left(\{|\texttt{p}_{pol} - \texttt{O}_{pol}||\forall \texttt{p}_{pol} \in \texttt{projections}_{pol}\}\right)$;
49: **for** $\texttt{i} \leftarrow 1$; $\texttt{i} <= \text{ncols}(\texttt{M})$; $\texttt{i} \leftarrow \texttt{i} + 1$ **do**
50: **if** $\texttt{i} \in \mathcal{S}$ **then**
51: **continue**; ▷ known seeds will be added later by default
52: **end if**
53: $\texttt{p}_{subj} \leftarrow \texttt{projections}_{subj}[i]$;
54: **if** $|\texttt{p}_{subj} - \mu_{\mathcal{O}}| > |\texttt{p}_{subj} - \mu_{\mathcal{S}}|$ **then** ▷ if the i-th word is subjective
55: $\texttt{score}_{subj} \leftarrow 1 + \frac{|\texttt{p}_{subj} - \texttt{O}_{subj}|}{\texttt{max\_score}_{subj}}$;
56: **else**
57: $\texttt{score}_{subj} \leftarrow 1 - \frac{|\texttt{p}_{subj} - \texttt{O}_{subj}|}{\texttt{max\_score}_{subj}}$;
58: **end if**
59: $\texttt{p}_{pol} \leftarrow \texttt{projections}_{pol}[i]$;
60: **if** $|\texttt{p}_{pol} - \mu_{\mathcal{N}}| > |\texttt{p}_{pol} - \mu_{\mathcal{P}}|$ **then** ▷ if the i-th word is positive
61: $\texttt{polarity} \leftarrow$ positive;
62: **else**
63: $\texttt{polarity} \leftarrow$ negative;
64: **end if**
65: $\texttt{score}_{pol} \leftarrow 1 + \frac{|\texttt{p}_{pol} - \texttt{O}_{pol}|}{\texttt{max\_score}_{pol}}$;
66: append($\texttt{scores}$, ($\texttt{i}$, $\texttt{polarity}$, $\frac{1}{\texttt{score}_{\text{subj}} + \texttt{score}_{pol}}$));
67: **end for**
68: **return** $\texttt{scores}$;
69: **end function**

Lexicon	# of Terms	Positive Expressions			Negative Expressions			Neutral Terms			Macro F_1	Micro F_1
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1		
Seed Set	20	**0.771**	0.102	0.18	0.568	0.017	0.033	0.963	**0.999**	0.981	0.398	0.962
TNG	1,600	0.088	0.153	0.112	0.193	**0.155**	**0.172**	**0.966**	0.953	0.959	0.414	0.921
VO	40	0.117	0.115	0.116	0.541	0.017	0.033	0.963	0.98	0.971	0.374	0.944
NC	5,200	**0.771**	0.102	0.18	0.568	0.017	0.033	0.963	**0.999**	0.981	0.398	0.962
k-NN	420	0.486	**0.182**	**0.265**	**0.65**	0.091	0.16	**0.966**	0.995	0.98	**0.468**	0.961
PCA	40	0.771	0.102	0.18	0.529	0.017	0.033	0.963	**0.999**	0.981	0.398	0.962
LP	6,340	0.741	0.156	0.257	0.436	0.088	0.147	**0.966**	0.998	**0.982**	0.462	**0.963**

Table 3.4: Results of NWE-based approaches
TNG – Tang et al. (2014a), VO – Vo and Zhang (2016), NC – nearest centroids,
k-NN – k-nearest neighbors, PCA – principal component analysis, LP – linear projection

Word Embeddings

An important aspect that could significantly affect the results of NWE-algorithms was the type of word embeddings that we provided to these systems as input. As we already noted at the beginning of this section, two most common kinds of such representations are standard word2vec and task-specific vectors (Mikolov et al., 2013; Collobert et al., 2011). The former type seeks to find a word representation that maximizes the probability of other tokens appearing in the nearby context, whereas the latter type optimizes these representations with respect to a specific custom task, such as polarity prediction of the whole text.

In order to see how these differences could affect the results of our approaches, we have trained task-specific embeddings on snapshot tweets, considering positive and negative emoticons that appeared in these messages as their noisy polarity labels, and re-evaluated our methods using these vectors. Apart from that, we additionally explored two in-between solutions:

- *hybrid embeddings*, which were trained by simultaneously optimizing two objectives—predicting the surrounding context and classifying the polarity of the tweet;
- and *least-squares embeddings*, for which we first obtained both embedding types, word2vec (V_{W2V}) and task-specific ones (V_{TS}). Since task-specific vectors, however, could only be learned on messages that contained emoticons or neutral seeds, many terms that had a word2vec representation did not have a task-specific counterpart. To derive these missing embeddings, we computed a transformation matrix W using the method of the ordinary least squares:

$$W = \underset{W}{\operatorname{argmin}} \|V_{TS} - W^T \cdot V^*_{W2V}\|_F, \tag{3.3}$$

where V^*_{W2V} represents a matrix of word2vec vectors whose words have both representations, and $\|\cdot\|_F$ means the Frobenius norm. Afterwards, we approximated task-specific

representations for all terms that were missing in V_{TS} by multiplying their word2vec vectors with matrix W.

As we can see from the results in Table 3.5, k-NN and linear projection work best with the standard word2vec embeddings (with the overall best score [0.468] achieved by k-NN), but their performance degrades as the input vectors become more and more aware of the polarity-prediction task. An opposite situation is observed for the nearest centroids and PCA, which show an improvement in combination with task-specific and least-squares vectors.

Lexicon	**Embedding Type**			
	word2vec	task-specific + word2vec	task-specific + least squares	task-specific
NC	0.398	0.398	**0.401**	0.399
k-NN	**0.468**	0.43	0.398	0.392
PCA	0.398	0.398	0.404	**0.409**
LP	**0.462**	0.441	0.398	0.399

Table 3.5: Macro-averaged F_1-scores of NWE-based methods with different embedding types

In order to understand the reasons for these differences, we projected the embeddings of all tokens that appeared in our corpus onto the two-dimensional vector space using the t-SNE method of van der Maaten and Hinton (2008), and visualized these vectors, highlighting polar terms from the Turney and Littman's seed set. Following the recommended practices for analyzing t-SNE (Wattenberg et al., 2016), we generated these lower-dimensional projections for different perplexity values: $p \in \{5, 30, 50\}$; and present the results of this visualization in Figure 3.2.

As we can see from the right column of the figure, task-specific representations of polar terms tend to appear close to each other, but apart from the rest of the vectors. Consequently, the centroids of these terms will be far away from the center of neutral words, which partially explains better results of the nearest centroids achieved with this embedding type. At the same time, because polar terms are far away from the majority of embeddings, only few of them will appear in the neighborhood of other words, which causes the k-NN classifier to consider most terms as neutral. Similarly, in the linear projection method, the optimal polarity line will run parallel to both polar and neutral lexemes, assigning high scores to both of these classes. Unfortunately, the subjectivity axis, which is supposed to help distinguish between subjective and objective instances, is apparently not strong enough to overcome this confusion.

Vector Normalization

Another factor that influenced the quality of NWE-induced polarity lists was length normalization and mean scaling of input vectors, which we applied at the very beginning of the

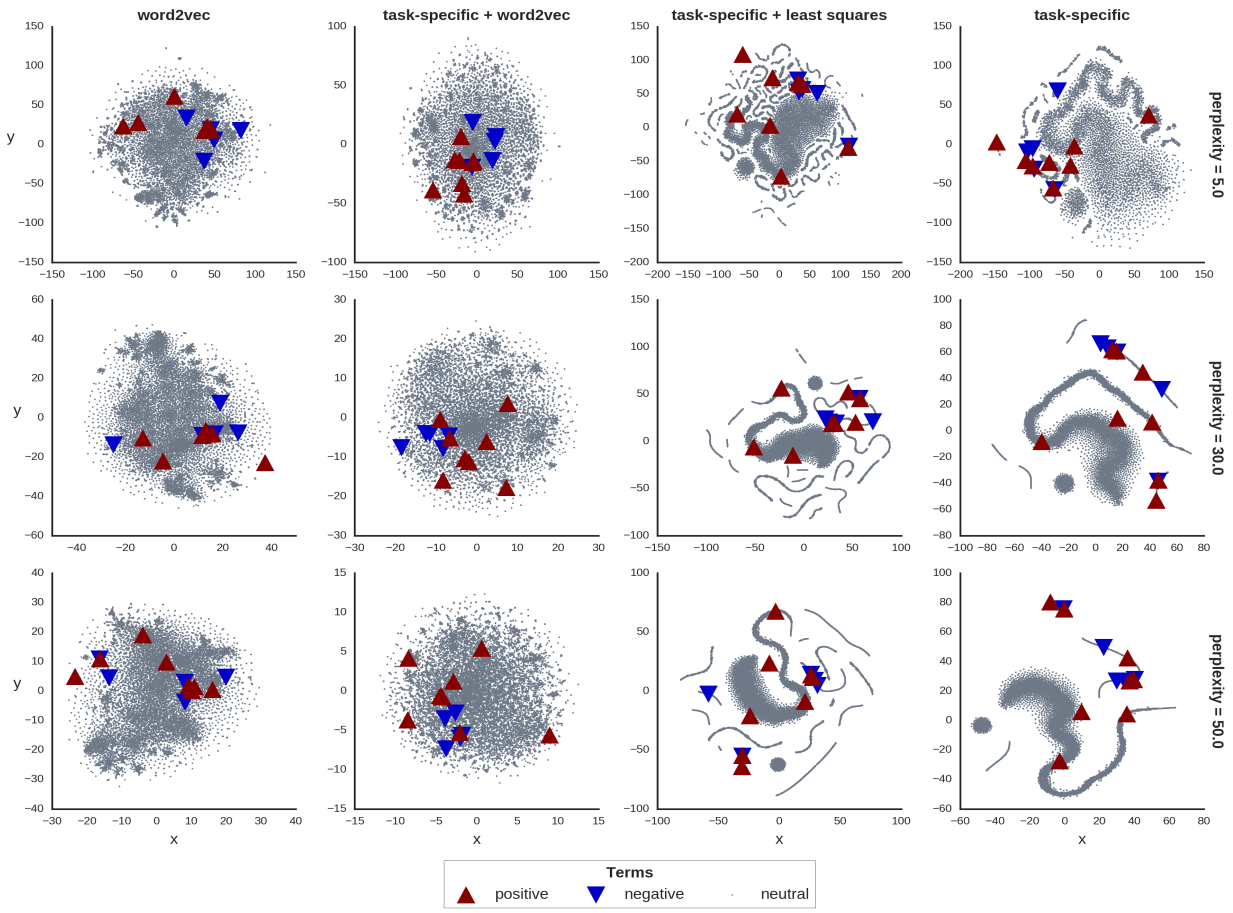


Figure 3.2: t-SNE visualization of PotTS' tokens and Turney and Littman's seed set with different embedding types

training.

To check the effect of this procedure, we reran our algorithms without vector normalization, and present the results of our evaluation in Table 3.6.

As we can see from the scores, nearest centroids are almost indifferent to this preprocessing, showing the same scores for all settings. But the remaining three approaches (k-NN, PCA, and linear projection) achieve their best results when both normalization steps are used. We should, however, admit that this success is mostly due to the length normalization rather than mean scaling. We can recognize this by comparing the scores in the first and third columns of the table, where PCA shows identical results, and the scores of k-NN and linear projection differ by only 0.001. We also can observe that mean-scaling alone has a strong negative effect on the linear projection system, pulling its scores down by 0.026 in comparison with unnormalized vectors.

SLG Method	Vector Normalization			
	mean normalization + length normalization	mean normalization	length normalization	no normalization
NC	**0.398**	**0.398**	**0.398**	**0.398**
k-NN	**0.468**	0.418	0.467	0.417
PCA	**0.398**	0.396	**0.398**	0.396
LP	**0.462**	0.416	0.461	0.442

Table 3.6: Macro-averaged F_1-scores of NWE-based methods with different vector normalizations

3.5 Seed Sets

Finally, the presumably most important factor that significantly affected the quality of all sentiment lexicons was the set of seed terms that we used to initialize these polarity lists. In order to estimate the impact of this setting, we rerun our experiments using the seed sets proposed by Hu and Liu (2004), Kim and Hovy (2004), Esuli and Sebastiani (2006a), and Remus et al. (2010). Since Hu and Liu (2004), however, only provided a few examples from their initial polarity list, and Kim and Hovy (2004) did not specify any seeds at all, we filled missing entries in these resources with common polar German words that we came up with in order to match the reported cardinalities of these sets. Moreover, in the cases where the above seed lists were missing the neutral category, we explicitly added a number of objective terms proportional to the number of their polar entries. Furthermore, because the seed set of Esuli and Sebastiani (2006a) had a total of 4,122 neutral terms,[8] which were difficult to translate manually, we automatically translated these entries by using a publicly available online dictionary[9] and taking the first suggested German translation for each neutral entry.[10] A short statistics on the cardinalities and compositions of the resulting seed sets is presented in Table 3.7.

The results of dictionary-based approaches obtained with these seeds are shown in Figure 3.3. This time, we again can notice better scores achieved by the method of Blair-Goldensohn et al. (2008), which not only outperforms other systems on average but is also less susceptible to the varying quality and cardinalities of different sets. The remaining methods typically achieve their best macro-averaged F_1-results with the polarity list of Kim and Hovy (2004) or seed set of Esuli and Sebastiani (2006a). The former option works best for the label-propagation approach of Rao and Ravichandran (2009) and the random walk algorithm of Awadallah and Radev (2010). The latter seeds yield best results for the approach of Hu and Liu (2004) and the SentiWordNet system of Esuli and Sebastiani

[8] The authors considered as neutral all terms from the General Inquirer lexicon (Stone et al., 1966) that were not marked there as either positive or negative.

[9] `http://www.dict.cc`

[10] We also tried using all possible translations of original terms, but it considerably increased the number of neutral items (45,252 words) and lead to a substantial decrease of the final system scores.

Seed Set	Cardinality	Part of Speech	Examples	Comments
Hu and Liu (2004)	14 positive, 15 negative, and 10 neutral terms	adjectives	*fantastisch, lieb, sympathisch, böse, dumm, schwierig*	polar terms translated from the original paper (Hu and Liu, 2004); neutral terms added by us;
Kim and Hovy (2004)	60 positive, 60 negative, and 60 neutral terms	any	*fabelhaft, Hoffnung, lieben, hässlich, Missbrauch, töten*	devised by us to match the cardinality of the original set with neutral terms added extra;
Esuli and Sebastiani (2006a)	16 positive, 35 negative, and 4,122 neutral terms	any	*angenehm, ausgezeichnet, freundlich, arm, bedauernswert, dürftig*	polar terms translated from the seed set of Turney and Littman (2003); neutral terms automatically translated from objective entries in the General Inquirer lexicon (Stone et al., 1966);
Remus et al. (2010)	12 positive, 12 negative, and 10 neutral terms	adjectives	*gut, schön, richtig, schlecht, unschön, falsch*	polar terms translated from the seed set of Turney and Littman (2003); neutral terms added by us;

Table 3.7: Overview of alternative seed sets
(all cardinalities are given with respect to the resulting German translations)

(2006a).

A different situation is observed for corpus-based methods, whose results are shown in Figure 3.4. Except for the approach of Takamura et al. (2005), which achieves its best score with the seed set of Hu and Liu (2004), all other systems (VEL, KIR, and SEV) show very similar (though not identical) scores as the ones reached with the seed set of Turney and Littman (2003) in our initial experiments. The primary reason for this is again the ambiguity of translated seeds, which leads to an early stopping of these algorithms.

As to NWE-based methods, whose scores are presented in Figure 3.5, we again can notice

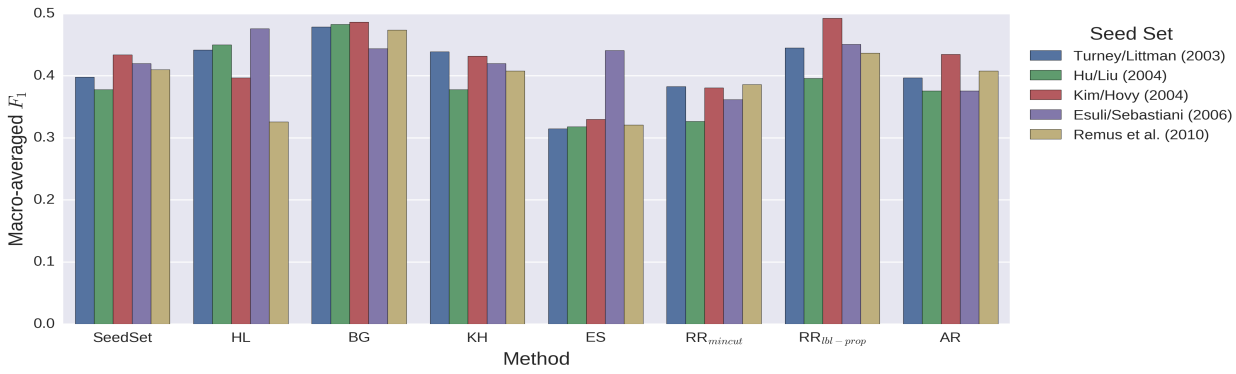


Figure 3.3: Macro-averaged F_1-scores of dictionary-based approaches with different seed sets

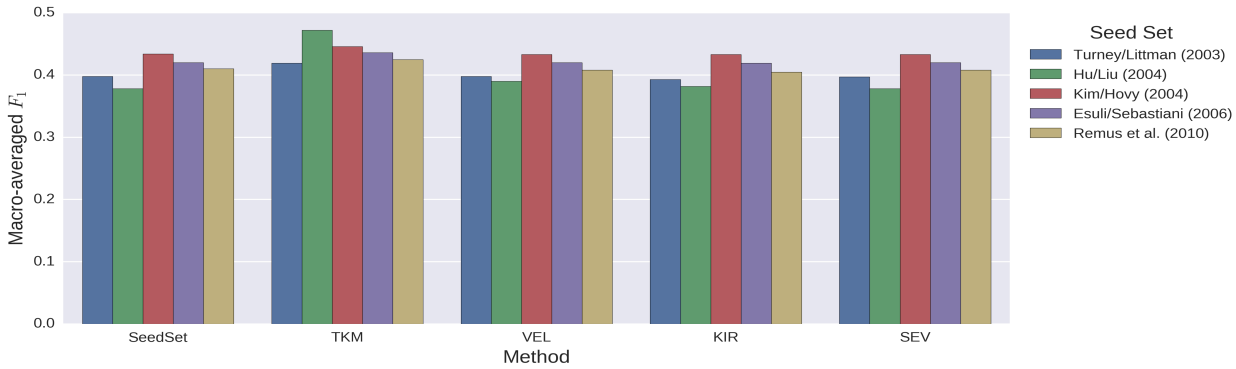


Figure 3.4: Macro-averaged F_1-scores of corpus-based approaches with different seed sets

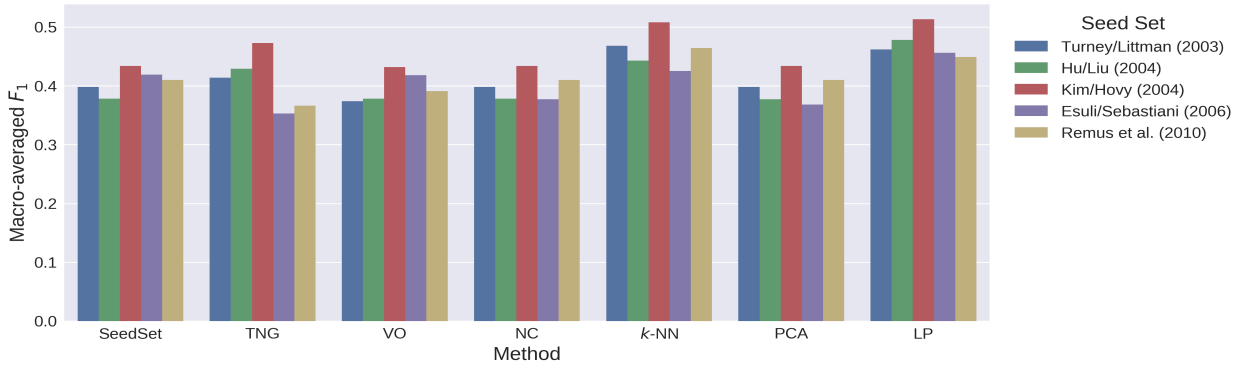


Figure 3.5: Macro-averaged F_1-scores of NWE-based approaches with different seed sets

the superior results of k-NN and linear projection, which both obtain their best macro-averages (0.508 for k-NN and 0.513 for the linear projection) with the seed set of Kim and Hovy (2004). Moreover, the F_1-score of the linear projection system achieved with these seeds outperforms the results of all other SLG approaches, setting a new state of the art on our corpus. The remaining NWE-based systems also attain their best scores with this seed list, which is not surprising regarding the much bigger number of polar terms in this set.

3.6 Analysis of Entries

Besides investigating the effects of different hyper-parameters and seed sets, we also decided to have a closer look at the actual results produced by the tested methods. For this purpose, we extracted ten highest-scored entries (not counting the seed terms) from each dictionary-based automatic lexicon and present them in Table 3.8.

As we can see from the table, the approaches of Hu and Liu (2004), Blair-Goldensohn et al. (2008), Kim and Hovy (2004), and the label-propagation algorithm of Rao and Ravichandran (2009) produce almost perfect polarity lists. The SentiWordNet approach of Esuli and Sebastiani (2006a), however, already features some spurious terms among its top-scored entries (*e.g.*, "absichtslos" [*unintentional*]). Finally, the min-cut approach of Rao and Ravichandran

Rank	HL	BG	KH	ES	$\mathbf{RR}^{**}_{\text{mincut}}$	$\mathbf{RR}_{\text{lbl-prop}}$
1	perfekt	fleißig	anrüchig	namenlos	planieren	prunkvoll
	perfect	*diligent*	*indecent*	*nameless*	*to plane*	*splendid*
2	mustergültig	böse	unecht	ruhelos	Erdschicht	sinnlich
	immaculate	*evil*	*artificial*	*restless*	*stratum*	*sensual*
3	vorbildlich	beispielhaft	irregulär	unbewaffnet	gefallen	pompös
	commendable	*exemplary*	*irregular*	*unarmed*	*please*	*ostentatious*
4	beispielhaft	edel	drittklassig	interesselos	Zeiteinheit	unappetitlich
	exemplary	*noble*	*third-class*	*indifferent*	*time unit*	*unsavory*
5	exzellent	tüchtig	sinnlich	reizlos	Derivat	befehlsgemäß
	excellent	*proficient*	*sensual*	*unattractive*	*derivate*	*as ordered*
6	exzeptionell	emsig	unprofessionell	würdelos	Oberfläche	vierschrötig
	exceptional	*busy*	*unprofessional*	*undignified*	*surface*	*beefy*
7	außergewöhnlich	eifrig	abgeschlagen	absichtslos	Essbesteck	regelgemäß
	extraordinary	*eager*	*exhausted*	*unintentional*	*cutlery*	*regularly*
8	außerordentlich	arbeitsam	gefällig	ereignislos	ablösen	wahrheitsgemäß
	exceptionally	*hardworking*	*pleasing*	*uneventful*	*to displace*	*true*
9	viertklassig	mustergültig	mustergültig	regellos	Musikveranstaltung	fettig
	fourth-class	*exemplary*	*exemplary*	*irregular*	*music event*	*greasy*
10	sinnreich	vorbildlich	unrecht	fehlerfrei	Gebrechen	lumpig
	ingenious	*commendable*	*wrong*	*accurate*	*afflictions*	*shabby*

Table 3.8: Top-10 polar terms produced by dictionary-based methods
*** – the min-cut method of Rao and Ravichandran (2009) returns an unsorted set*

(2009) returns mostly objective terms, which, however, is due to the fact that this method performs a cluster-like partitioning of the lexical graph without actually ranking the words assigned to the clusters.

A different situation is observed with corpus-based systems as shown in Table 3.9: The top-scoring polarity lists returned by all of these approaches not only include many apparently neutral terms but are also difficult to interpret in general, as they contain a substantial number of slang and advertising expressions (*e.g.*, "BMKS65," "#gameinsight," "#androidgames").

We can also observe a similar trend for the most NWE-based methods, whose results are presented in Table 3.10. As we can see from the examples, many of these systems obviously overrate Internet-specific terms (*e.g.*, "%user-playlist," "%user-video," "www.op"), and assign higher weights to foreign words (*e.g.*, "nerelere," "good," "nativepride") and interjections (*e.g.*, "niedlichgähn" [*cuteyawn*], "vrrrum"). Two notable exceptions from this trend are k-NN and linear projection, whose top-scoring entries contain exclusively polar terms. At the same time, we can notice a slight susceptibility of these approaches to the negative polarity class as eight out of ten highest ranked words in their results have negative semantic orientation. One possible explanation for this could be a more pronounced distribution of negative expressions, which pushes the vectors of these terms to more distinguishable regions than in the case of positive lexemes.

Rank	TKM	VEL	KIR	SEV
1	Stockfotos	Wahlkampfgeschenk	Suchmaschinen	Scherwey
	stock photos	*election gift*	*search engines*	*Scherwey*
2	BMKS65	Ordensgeschichte	#gameinsight	krebsen
	BMKS65	*order history*	*#gameinsight*	*to crawl*
3	Ziya	Indologica	#androidgames	kaschieren
	Ziya	*Indologica*	*#androidgames*	*to conceal*
4	Shoafoundation	Indologie	Selamat	Davis
	shoah found.	*Indology*	*selamat*	*Davis*
5	T1199	Energieverbrauch	Pagi	#Klassiker
	T1199	*energy consumption*	*Pagi*	*#classics*
6	Emilay55	Schimmelbildung	#Sparwelt	Nationalismus
	Emilay55	*mold formation*	*#savingsworld*	*nationalism*
7	Eneramo	Hygiene	#Seittest	Kraftstoff
	Eneramo	*hygiene*	*#Seittest*	*fuel*
8	GotzeID	wasserd	Gameinsight	inaktiv
	GotzeID	*waterp*	*Gameinsight*	*idle*
9	BSH65	heizkostensparen	#ipadgames	8DD
	BSH65	*saving heating costs*	*#ipadgames*	*8DD*
10	Saymak.	Referenzarchitekturen	Fitnesstraining	Mailadresse
	Saymak.	*reference architectures*	*fitness training*	*mail address*

Table 3.9: Top-10 polar terms produced by corpus-based methods

Rank	TNG	VO	NC	k-NN	PCA	LinProj
1	internetvorräte	guz	paion	eklig	gwiyomi.	dumm
	Internet inventories	*guz*	*paion*	*yukky*	*gwiyomi.*	*stupid*
2	%user-playlist	nerelere	auf⌟	ätzend	seitens	eklig
	%user playlist	*nerelere*	*on⌟*	*lousy*	*on the part of*	*yukky*
3	dumm	www.op	folgen!	lächerlich	kritisieren	fies
	stupid	*www.op*	*follow!*	*ridiculous*	*to criticize*	*nasty*
4	wunderschön	fernsehfestival	teil8	doof	nanda	doof
	gorgeous	*TV festival*	*part8*	*dumb*	*nanda*	*dumb*
5	ölgemälde	positip	stanzmesser	dumm	@deinskysport	blöd
	oil painting	*positip*	*punch knife*	*stupid*	*@deinskysport*	*stupid*
6	%user-video	arn	niedlichgähn	wunderbar	doubts	komisch
	%user video	*arn*	*cuteyawn*	*winderful*	*doubts*	*funny*
7	verlosen	asri	vrrrum	toll	temos	traurig
	to raffle	*asri*	*vrrrum*	*great*	*temos*	*sad*
8	wünschen	bewerten	good⌟	widerlich	temas	dämlich
	to wish	*to rate*	*good⌟*	*disgusting*	*temas*	*silly*
9	dämlich	nacht	nativepride	nervig	balas	peinlich
	silly	*night*	*nativepride*	*annoying*	*balas*	*embarrassing*
10	peinlich	morgen	⌜whistle⌟	schrecklich	hepi	scheißen
	embarrassing	*morning*	*⌜whistle⌟*	*awful*	*hepi*	*to crap*

Table 3.10: Top-10 polar terms produced by NWE-based methods

3.7 Summary and Conclusions

Concluding this chapter, we would like to recapitulate that, in this part, we have presented a thorough review of the most popular sentiment lexicon generation methods. For this purpose, we first revised existing lexicon evaluation techniques and suggested our own (stricter) metric, in which we explicitly counted all false positive, false negative, and true positive occurrences of positive, negative, and neutral terms on a real-life sentiment corpus, and also computed the macro- and micro-averaged F_1-results of these polarity classes. Using our procedure, we first evaluated the most popular semi-automatic German lexicons: German Polarity Clues (Waltinger, 2010), SentiWS (Remus et al., 2010), and the Zurich Polarity List (Clematide and Klenner, 2010), finding the last resource working best in terms of the macro-F_1–score. Afterwards, we estimated the quality of automatic polarity lists that were created with dictionary- and corpus-based methods, coming to the conclusion that the former group generally produced better lexicons and was less susceptible to noisy Twitter domain. In the next step, we introduced several novel SLG approaches that operate on neural embeddings of words, showing that at least two of them (k-nearest neighbors and linear projection) outperformed all other compared automatic SLG algorithms. Last but not least, we explored the effect of different hyper-parameters and settings on the net results of these methods, rerunning them with alternative sets of initial seed terms, checking their performance on different kinds of embeddings, and estimating the impact of various vector normalization techniques.

Based on these observations and experiments, we can formulate the main conclusions of this chapter as follows:

- semi-automatic translations of common English polarity lists notably outperform purely automatic SLG methods, which are applied to German data directly;
- despite their allegedly worse ability to accommodate new domains, dictionary-based approaches are still better than corpus-based systems (at least in terms of our intrinsic metric);
- a potential weakness of these algorithms though is their dependence on various types of hyper-parameters and manually annotated linguistic resources, which might not necessarily be present for every language;
- in this regard, a viable alternative to dictionary-based methods are SLG systems that induce polar lexicons from neural word embeddings, which not only avoid the above limitations but also yield competitive (or even better) results;

- with at least two of such methods (k-NN and linear projection), we were able to establish a new state of the art for the macro- and micro-averaged F_1-scores of automatically induced sentiment lexicons;

- we also checked how different types of embeddings affected the performance of NWE-based SLG systems, noticing that the k-NN and linear projection methods worked best with standard word2vec vectors, while nearest centroids and PCA yielded better results when using task-specific representations;

- furthermore, we saw that all NWE-based approaches benefited from mean-scaling and length normalization of input vectors, improving by up to 5% on their macro-averaged F_1-scores;

- finally, an extensive evaluation of various sets of seed terms revealed that the results of almost all tested SLG algorithms crucially depend on the quality of their initial seeds, with larger and more balanced seed sets, *e.g.*, like the one proposed by Kim and Hovy (2004), typically leading to much higher scores.

Bearing this knowledge in mind, we will now move on to exploring further opinion-mining fields: fine-grained and message-level sentiment analysis, in which sentiment lexicons are traditionally considered as one of the most valuable building blocks.

Chapter 4

Fine-Grained Sentiment Analysis

The task of fine-grained sentiment analysis (FGSA) is to automatically recognize subjective evaluative opinions (*sentiments*), holders of these opinions (*sources*), and their respective evaluated entities (*targets*) in text. Since an accurate automatic prediction of these elements would allow us to track public's attitude towards literally any object (*e.g.*, a product, a service, or a political decision), FGSA is traditionally considered as one of the most attractive, necessary, but, unfortunately, also challenging objectives in the opinion-mining field.

Researchers usually interpret this goal as a sequence labeling (SL) objective, and address it with one of two most popular SL techniques: conditional random fields (CRFs) or recurrent neural networks (RNNs). The former approach represents a discriminative probabilistic graphical model, which relies on an extensive set of hand-crafted features, whereas the latter methods use a recursive computational loop and learn their feature representations completely automatically. In this chapter, we are going to evaluate each of these solutions in detail in order to find out which of these algorithms is better suited for the domain of German Twitter. But before we proceed with our experiments, we should first briefly discuss evaluation metrics that we are going to use to estimate the quality of these systems.

4.1 Evaluation Metrics

Because fine-grained sentiment analysis operates on *spans* of sentiment labels, which typically consist of multiple contiguous tags, we cannot straightforwardly apply metrics that are used for evaluation of single independent instances to this objective, as it is unclear which instances should be measured—single tokens or complete spans—and how partial matches should be counted in the latter case.

One possibility to estimate the quality of FGSA prediction is to compute precision, recall, and F_1-scores of predicted spans by using *binary-overlap* or *exact-match* metrics (see Choi

et al., 2006; Breck et al., 2007). The first method considers an automatically labeled span as correct if it has at least one token in common with a labeled element from the gold annotation. The second metric only regards an automatic span as true positive if its boundaries are absolutely identical with the span annotated by the human expert. Unfortunately, both of these approaches are problematic to a certain extent: While binary overlap might be overly optimistic, always assigning perfect scores to automatic spans that cover the whole sentence; exact match might, vice versa, be too drastic, considering the whole assignment as false if only one (possibly irrelevant) token is classified incorrectly.

Instead of relying on these measures, we decided to use a "golden mean" solution proposed by Johansson and Moschitti (2010), in which they penalize predicted spans proportionally to the number of tokens whose labels are different from the gold annotation. More precisely, given two sets of manually and automatically tagged spans ($\mathcal{S}$ and $\widehat{\mathcal{S}}$, respectively), Johansson and Moschitti estimate the precision of automatic assignment as:

$$P(\mathcal{S}, \widehat{\mathcal{S}}) = \frac{C(\mathcal{S}, \widehat{\mathcal{S}})}{|\widehat{\mathcal{S}}|}, \tag{4.1}$$

where $C(\mathcal{S}, \widehat{\mathcal{S}})$ stands for the proportion of overlapping tokens across all pairs of manually (s_i) and automatically (s_j) annotated spans:

$$C(\mathcal{S}, \widehat{\mathcal{S}}) = \sum_{s_i \in \mathcal{S}} \sum_{s_j \in \widehat{\mathcal{S}}} c(s_i, s_j),$$

and the $|\widehat{\mathcal{S}}|$ term denotes the total number of spans automatically labeled with the given tag.

Similarly, the recall of this assignment is estimated as:

$$R(\mathcal{S}, \widehat{\mathcal{S}}) = \frac{C(\mathcal{S}, \widehat{\mathcal{S}})}{|\mathcal{S}|}.$$

Using these two values, one can normally compute the F_1-measure as:

$$F_1 = 2 \times \frac{P \times R}{P + R}.$$

Because this estimation adequately accommodates both extrema of automatic annotation (too long and too short spans) and also penalizes erroneous labels, we will rely on this measure throughout our subsequent experiments.

4.2 Data Preparation

In order to evaluate CRFs and RNNs on our dataset, we split the complete corpus annotated by the second annotator, which we will henceforth consider as gold standard in all subsequent

experiments, into three parts, using 70% of it for training, 10% as development data, and the remaining 20% as a test set. We tokenized all tweets with the same adjusted version of Potts' tokenizer that we used previously while creating the initial corpus files, and preprocessed these microblogs with the rule-based normalization pipeline of Sidarenka et al. (2013). In this procedure, we:

- *unified Twitter-specific phenomena* such as @-mentions, hyperlinks, and e-mail addresses by replacing these entities with special tokens that represented their semantic classes (*e.g.*, "%Username" for @-mentions, "%URI" for hyperlinks). We removed these elements from the input, if they were grammatically independent from the rest of the tweet and did not play a potential role for the expression of sentiments (*e.g.*, we stripped off all retweet mentions and hyperlinks appearing at the very end of the microblog if they were not preceded by a preposition). Furthermore, we substituted all emoticons with special placeholders representing their semantic orientation (*e.g.*, ☺ → "%PosSmiley," ☹ → "%NegSmiley," `:-O` → "%Smiley"), and removed the hash sign (#) from all hashtags (*e.g.*, "#glücklich" → "glücklich");
- In addition to this, we *restored frequent misspellings* (*e.g.*, "zuguckn" → "zugucken" [*to watch*], "Tach" → "Tag" [*day*]), using a set of manually-defined heuristic rules;
- and, finally, *replaced frequent slang terms and abbrebiations with their standard-language equivalents* (*e.g.*, "n bissl" → "ein bisschen" [*a bit of*], "iwie" → "irgendwie" [*somehow*], "nix" → "nichts" [*nothing*]).

Afterwards, we labeled all normalized sentences with part-of-speech tags using TreeTagger[1] (Schmid, 1995), and parsed them with the Mate dependency parser[2] (Bohnet et al., 2013).[3] Finally, since `MMAX2` did not provide a straightforward support for character offsets of annotated tokens and because automatically tokenized data could disagree with the original corpus tokenization, we aligned manual annotation with automatically split words with the help of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970).

4.3 Conditional Random Fields

The first method that we evaluated using the obtained data was conditional random fields. First introduced by Lafferty et al. (2001), CRFs have rapidly grown in popularity, turning

[1] In particular, we used TreeTagger Version 3.2 with the German parameter file UTF-8.

[2] We used Mate Version `3.61` with the German parameter model 3.6.

[3] The choice of these tools was motivated by their better results in our evaluation study, which we conducted while working on the normalization module (Sidarenka et al., 2013).

into one of the most widely used probabilistic frameworks, which was dominating the NLP field for almost a decade.

The main reasons for the success of this model are:

1) the *structural nature* of CRFs, which, in contrast to single-entity classifiers, such as logistic regression or SVM, make their predictions over structured input, trying to find the most likely label assignment to the whole structure (typically a chain or a tree) and not only its individual elements;

2) the *discriminative power* of this framework, which, in contrast to generative probabilistic models such as HMMs (Rabiner and Juang, 1986), optimizes conditional probability $P(\boldsymbol{Y}|\boldsymbol{X})$ instead of joint distribution $P(\boldsymbol{X}, \boldsymbol{Y})$ and consequently can efficiently deal with overlapping and correlated features;

3) and, finally, the *avoidance of the label bias problem*, which other discriminative classifiers, such as maximum entropy Markov networks (McCallum et al., 2000), are known to be susceptible to.

Example 4.3.1 (Label Bias Problem)

The label bias problem arises in the cases where a locally optimal decision outweighs globally superior solutions. Consider, for example, the sentence "Aber gerade Erwachsene haben damit Schwierigkeiten." (But especially adults have difficulties with it.), for which we need to compute the most probable sequence of part-of-speech tags.

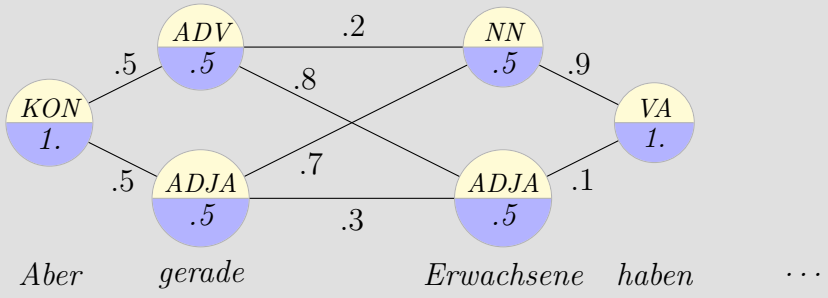


Figure 4.1: Example of a CRF graph

Using features weights shown in Figure 4.1, we will first estimate the probability of the correct label sequence for the initial part of this sentence using the Maximum Entropy Markov Model (MEMM)—the predecessor of the Conditional Random Fields. According to the MEMM's definition, the probability of correct labeling $(KON-ADV-NN-VA)$

is equal to:

$$\begin{aligned} P(KON, ADV, NN, VA) &= P(KON) \times P(ADV|KON) \\ &\times P(NN|ADV) \times P(VA|NN) \\ &= \frac{\exp(1)}{\exp(1)} \times \frac{\exp(0.5+0.5)}{\exp(0.5+0.5)+\exp(0.5+0.5)} \\ &\times \frac{\exp(0.2+0.5)}{\exp(0.2+0.5)+\exp(0.8+0.5)} \\ &\times \frac{\exp(0.9+1.)}{\exp(0.9+1.)} \approx 0.177 \end{aligned}$$

At the same time, the probability of the wrong variant ($KON-ADV-ADJA-VA$) amounts to $\approx$ 0.323 and will therefore be preferred by the automatic tagger.

A different situation is observed with CRFs, where the normalizing factor in the denominator is computed over the whole input sequence without factorizing into individual terms for each transition as it is done in MEMM. This way, the probability of correct labels will run up to:

$$\begin{aligned} P(KON, ADV, NN, VA) =& P(KON) \times P(ADV|KON) \times P(NN|ADV) \\ &\times P(VA|NN) \\ =& \frac{\exp(1+0.5\times 3+0.2+0.9+1)}{Z} \approx 0.252, \end{aligned}$$

where $Z = \exp(1+0.5\times 3+0.2+0.9+1)+\exp(1+0.5\times 3+0.8+0.1+1)+\exp(1+0.5\times 3+0.7+0.9+1)+\exp(1+0.5\times 3+0.3+0.1+1)$ *is the total score of all possible label assignments; the incorrect alternative ($KON-ADV-ADJA-VA$), however, will get a probability score of $\approx$ 0.207, which is less than the score of the correct labeling.*

Training. CRFs have these useful properties due to a neatly formulated objective function in which they seek to optimize the global log-likelihood of gold labels $\mathbf{Y}$ conditioned on training data $\mathbf{X}$. In particular, given a set of training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$, where $\mathbf{x}^{(n)}$ stands for the covariates of the n-th instance, and $\mathbf{y}^{(n)}$ denotes its respective gold labels, CRFs try to find feature coefficients $\mathbf{w}$ that maximize the log-probabilities ℓ of $\mathbf{y}^{(i)}$ given $\mathbf{x}^{(i)}$ over the whole corpus:

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{n=1}^{N} \ell\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}\right). \tag{4.2}$$

The log-likelihood $\ell(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})$ in this equation is commonly estimated as the logarithm of globally (*i.e.*, w.r.t. to the whole instance) normalized softmax function:

$$\ell\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}\right) = \ln\left(P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})\right) = \ln\left(\frac{\exp\left(\sum_{m=1}^{M}\sum_{j} w_j \cdot f_j(x_m, y_{m-1}, y_m)\right)}{Z}\right), \quad (4.3)$$

in which M means the length of the n-th training example; $f_j(x_m, y_{m-1}, y_m)$ denotes the value of the j-th feature function f at position m; w_j represents the corresponding weight of this feature; and Z is a normalization factor calculated over all possible label assignments:

$$Z := \sum_{y' \in \mathcal{Y}, y'' \in \mathcal{Y}} \exp\left(\sum_{m=1}^{M}\sum_{j} w_j \cdot f_j(x_m, y'_{m-1}, y''_m)\right).$$

Since this normalizing term appears in the denominator and couples together all feature weights that need to be optimized, it becomes prohibitively expensive to find the best solution to Equation 4.2 analytically, with a single shot. A possible remedy to this problem is to resort to other optimization techniques, such as gradient descent, where feature weights are successively changed in the direction of their gradient until they reach the minimum of the loss function.

From Equation 4.3, we can see that the partial derivative of log-likelihood w.r.t. a single feature weight w_j is:

$$\frac{\partial}{\partial w_j}\ell = \sum_{n=1}^{N}\sum_{m=1}^{M} f_j(x_m, y_{m-1}, y_m) - \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{y' \in \mathcal{Y}, y'' \in \mathcal{Y}} f_j(x_m, y'_{m-1}, y''_m) P(y', y''|\mathbf{x}^{(n)}),$$

which, after dividing both parts of the equation by the constant term N (the size of the corpus) can be transformed into:

$$\frac{1}{N}\frac{\partial}{\partial w_j}\ell = \mathbb{E}[f_j(\mathbf{x}, \mathbf{y})] - \mathbb{E}_{\mathbf{w}}[f_j(\mathbf{x}, \mathbf{y})],$$

where the first term ($\mathbb{E}[f_j(\mathbf{x}, \mathbf{y})]$) is the expectation of feature f_j under empirical distribution, and the second term ($\mathbb{E}_{\mathbf{w}}[f_j(\mathbf{x}, \mathbf{y})]$) is the same expectation under model's parameters $\mathbf{w}$. In other words, the optimal solution to the log-likelihood objective in Equation 4.3 is achieved when model's expectation of features matches their (true) empirical expectation on the corpus.

The marginal probabilities of these features, which are required for computing their expectations, can be estimated dynamically using the forward-backward (FB) algorithm (Rabiner, 1990), which is a particular case of the more general belief-propagation method (see Barber, 2012, p. 81).

The only modification that one usually makes to Equation 4.2 in practice, before applying it the the provided training set, is the addition of so-called *regularization terms* (L1 and L2),

which penalize excessively high feature weights, thus preventing the model from overfitting the training data, *i.e.*, we no longer seek feature weights that simply maximize the probability of observed data, but we also want these weights to be as small as possible:

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{n=1}^{N} \ell\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}\right) - \lambda_1\|\mathbf{w}\|_1 - \lambda_2\|\mathbf{w}\|_2, \tag{4.4}$$

where λ_1 and λ_2 are manually set hyper-parameters, which control the amount of penalty that we want impose on the L1 and L2 norms of the weights.

In our experiments, we also adopted this enhanced objective, picking hyper-parameter values that yielded the best results on the held-out development set. Furthermore, in order to reduce the noise that is typically introduced by rare, sporadic features, we only optimized the weights of features that occurred two or more times in the training corpus, ignoring all singleton attributes from these data.

Inference. Once optimal feature weights have been learned, one can unproblematically compute the most likely label assignment for a new instance by using the Viterbi algorithm (Viterbi, 1967), which effectively corresponds to the forward pass of the FB method with the summation over the alternative preceding states replaced by the maximum operator (hence the other name for this algorithm, "max-product").

Features. A crucial component that accounts for a huge part of the success (or failure) of CRFs is features that are provided to this classifier as input.

Traditionally, feature functions in CRFs are divided into transition- and state-based ones. Transition features represent real- or binary-valued functions $f(\mathbf{x}, y'', y') \rightarrow \mathbb{R}$ associated with some data predicate $\phi(\mathbf{x}) \rightarrow \mathbb{R}$ and two labels y'' (typically the label of the previous token) and y' (usually the label of the current word). The value of this function at position m in sequence $\mathbf{x}$ is then defined as:

$$f(\mathbf{x}_m, y'', y') = \begin{cases} \phi(\mathbf{x}_m), & \text{if } \mathbf{y}_{m-1} = y'' \text{ and } \mathbf{y}_m = y' \\ 0, & \text{otherwise;} \end{cases}$$

where predicate ϕ usually represents a simple unit function: $\phi(\mathbf{x}_m) \mapsto 1$, $\forall \mathbf{x}_m$.

In contrast to ternary transition features, state attributes are typically associated with binary predicates, whose output depends on the input data at the given position and label y' at the respective state:

$$f(\mathbf{x}_m, y') = \begin{cases} \phi(\mathbf{x}_m), & \text{if } \mathbf{y}_m = y' \\ 0, & \text{otherwise.} \end{cases}$$

This time, predicate ϕ is usually much more sophisticated and reflects various properties of the input, such as whether the current token is capitalized or whether it begins with a specific prefix or ends with a certain suffix. This type of features commonly accounts for the overwhelming majority of all attributes in CRFs.

As state attributes in our experiments, we used the following features, which, for simplicity, are listed in groups:

- *formal*, which included the initial three characters of each token (*e.g.*, $\phi_{abc}(\mathbf{x}_m) = 1$ if $\mathbf{x}_m \sim$ /^abc/ else 0), its last three characters, and the spelling class of that word (*e.g.*, alphanumeric, digit, or punctuation);
- *morphological*, which encompassed part-of-speech tags of analyzed tokens, grammatical case and gender of inflectable PoS types, degree of comparison for adjectives, as well as mood, tense, and person forms for verbs;
- *lexical*, which comprised the actual lemma and form of the analyzed token (using one-hot encoding), its polarity class (positive, negative, or neutral), which we obtained from the Zurich Polarity Lexicon (Clematide and Klenner, 2010);
- and, finally, *syntactic* features, which reflected the dependency relation via which token x_m was connected to its parent. In addition to this, we also used two binary attributes that showed whether the previous token in the sentence was the parent (first feature) or a child (second feature) of the current word. Apart from that, we devised two more features, one of which encoded the dependency relation of the previous token in the sentence to its parent + the dependency relation of the current token to its ancestor; another feature reflected the dependency link of the next token + the dependency relation of the current token to its parent.

Besides the above attributes, we also introduced a set of complex *lexico-syntactic* features, which simultaneously reflected several semantic and syntactic traits. These were:

- the lemma of the syntactic parent;
- the part-of-speech tag and polarity class of the grandparent in the syntactic tree;
- the lemma of the child node + the dependency relation between the current token and its child;
- the PoS tag of the child node + its dependency relation + the PoS tag of the current token;

Data Set	Sentiment			Source			Target			Macro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1
Training Set	0.949	0.908	0.928	0.903	0.87	0.886	0.933	0.865	0.898	0.904
Test Set	0.37	0.28	0.319	0.305	0.244	0.271	0.304	0.244	0.271	0.287

Table 4.1: Results of fine-grained sentiment analysis with the first-order linear-chain CRFs

- the lemma of the child node + its dependency relation + the lemma of the current token;
- the overall polarity of syntactic children, which was computed by summing up the polarity scores of all immediate dependents, and checking whether the resulting value was greater, less than, or equal to zero.[4]

Results. The results of our experiments are shown in Table 4.1. As we can see from the table, with the given set of features, CRF can perfectly well fit the training data, achieving a macro-averaged F_1-score of 0.904. This model, however, can only partially generalize to unseen messages, where its macro-F_1 reaches merely 0.287, despite the fact that the size of the training corpus is almost 3.5 times bigger than the size of the test set (5,616 versus 1,584 tweets).

4.3.1 Feature Analysis

To estimate the effect of different features on the net results of the CRF system, we performed an ablation test, removing one group of state attributes at a time and rechecking the performance of the model on the development data.

Element	Original F_1-Score	F_1-Score after Feature Removal				
		Formal	Morphological	Lexical	Syntactic	Complex
Sentiment	0.346	$0.343^{-0.003}$	$0.344^{-0.002}$	$0.326^{-0.02}$	$0.345^{-0.001}$	$0.324^{-0.022}$
Source	0.309	$0.321^{+0.012}$	$0.313^{+0.004}$	$0.265^{-0.044}$	$0.359^{+0.05}$	$0.271^{-0.038}$
Target	0.26	$0.282^{+0.022}$	$0.252^{-0.008}$	$0.263^{+0.003}$	$0.233^{-0.027}$	$0.263^{+0.003}$

Table 4.2: Results of the feature ablation tests for the CRF model (negative changes w.r.t. the original scores on the development set are shown in red; positive changes are depicted in green superscripts)[5]

As we can see from the results in Table 4.2, all feature groups are useful for predicting

[4] We again used the Zurich Polarity Lexicon of Clematide and Klenner (2010) for computing these scores.

[5] Negative changes indicate good features in this context, since their removal leads to a degradation of the results.

`sentiment`s, as their removal leads to a degradation of its scores. This quality drop, however, is usually quite small, suggesting that other features can easily make up for the removed attributes. A different situation is observed with `source`s and `target`s though. In the former case, removing formal, morphological, and syntactic features shows a strong positive effect, improving the F_1-scores for `source`s by up to five percent. Removing lexical and lexico-syntactic features, on the contrary, worsens these results, tearing the F_1-measure down by 4.4%. Except for the formal group, all these attributes behave completely differently when applied to `target`s, which benefit from morphological and syntactic features, but apparently get confused by lexical and complex attributes.

Rank	**State Features**		**Transition Features**	
	Feature	Score	Feature	Score
1	prntLemma=meiste → TRG	18.68	NON → TRG	-7.01
2	prntLemma=rettungsschirme → TRG	18.3	NON → SRC	-6.85
3	initChar=sty → NON	-16.04	NON → SNT	-5.39
4	form=meisten → NON	15.99	TRG → SRC	-2.99
5	prntLemma=urlauberin → SNT	14.74	NON → NON	2.69
6	lemma=anfechten → SNT	14.07	SRC → NON	-2.59
7	form=thomasoppermann → TRG	13.44	SNT → SNT	2.54
8	form=bezeichnete → SNT	13.25	TRG → TRG	2.31
9	deprel[0]|deprel[1]=NK|AMS → NON	12.92	SRC → SRC	2.19
10	trailChar=te. → NON	12.77	SRC → TRG	-2.07

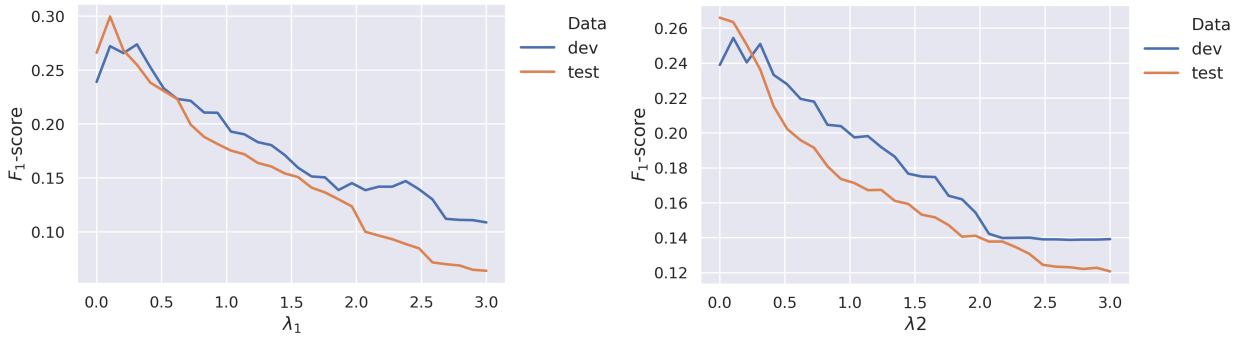
Table 4.3: Top-10 state and transition features learned by the CRF model (sorted by the absolute values of their weights)

In order to get a better insight into the learned model's parameters, we additionally extracted top-ten state and transition features, ranked by the absolute values of their weights. As we can see from the statistics in Table 4.3, three of five top-ranked state attributes ("meiste" [*most*], "rettungsschirme" [*bailout*], and "urlauberin") are complex features that reflect the lemma of the syntactic parent. Another common group of features is lemma and form of the current token: here, we again encounter the word "meisten" (*most*), which, however, indicates the absence of any sentiments this time, and we also can see two other attributes ("anfechten" [*doubt*] and "bezeichnete" [*called*]) that represent the so-called *direct speech events* and correlate with `sentiment`s. The remaining feature ("thomasopperman") is a person name, which frequently appears as sentiment's `target` in our corpus.

An interesting pattern can be observed with transition features: As we can see from the results, top three of these attributes indicate a strong belief in that an objective token is very unlikely to be followed by a `target`, `source`, or `sentiment` tag (hence, the high negative weights of transitions emanating from NON). It is, however, quite common that a NON tag will precede another NON (as we can see from line 5 of the table). Other transitions also mainly reflect plausible regularities: It is, for instance, uncommon that a `target` of an

opinion will appear immediately before a source (TRG→SRC = −2.99); in the same vein, it is fairly improbable that an SRC tag will precede a `TRG` element (SRC→TRG = −2.07); nonetheless, is is perfectly acceptable that the same tag will continue over multiple words (*e.g.*, SNT→SNT = 2.54, TRG→TRG = 2.31).

In order to better understand the reason for the observed overfitting of the weights to the training data, we also compared all features that appeared in the training set with the attributes that occurred in the test part of the corpus. As it turned out, more than two thirds of all unique test features (34,186 out of 49,626) have never been observed during the training and consequently had no meaningful model weights.



(a) λ_1 *(b)* λ_2

Figure 4.2: Results of the linear-chain CRFs with different values of regularization parameters

Another factor that could significantly affect the generalization of the CRF system was the regularization parameters λ_1 and λ_2, which controlled the amount of penalty imposed on too big learned feature weights (see Equation 4.4). Because we chose these parameters based on the model's results on the held-out development data, a possible reason for rather low scores on the test set could be a considerable difference between the distribution of `sentiments`, `sources`, and `targets` in the development and test parts of the corpus. To see whether it indeed was the case, we recomputed the F_1-scores on the development and test data, using different λ values, and present the results of this computation in Figure 4.2. As is evident from the figure, model's F_1-measure on the development set largely correlates with its performance on the test corpus, and almost monotonically decreases with larger λs.

4.3.2 Error Analysis

Besides looking into model's parameters, we also decided to analyze some errors made by the CRF system in order to understand the reasons for its misclassifications.

Example 4.3.2 (An Error Made by the CRF System)
Gold Labels: Überall/TRG NPD/TRG Plakate/TRG %NegSmiley/SNT
Everywhere/TRG NPD/TRG posters/TRG %NegSmiley/SNT

Predicted Labels: Überall/NON NPD/NON Plakate/NON %NegSmiley/NON
Everywhere/NON NPD/NON posters/NON %NegSmiley/NON

One such error is shown in Example 4.3.2. In this case, the classifier has erroneously overlooked a negative emoticon, which expresses author's attitude to election posters of the National Democratic Party of Germany (NPD), and assigned the NON (none) tags to all tokens of the tweet. As it turns out, despite this incorrect assignment, the state potentials of the smiley still achieve their highest scores with the correct SNT (`sentiment`) tag. Moreover, the state scores of the word "Plakate" (*posters*) also reach their maximum value (0.13 in the logarithmic domain) with the correct TRG (`target`) label. Unfortunately, these good guesses of single tags are overruled by the extremely high score of the NON label (6.515) that is assigned to the first word of this message ("überall" [*everywhere*]) and is reinforced by the transition features, which prefer contiguous runs of NONs.

This kind of mistakes is by far the most common type of errors that we have observed on the development set, followed by spans with different boundaries and invalid label sequences similar to the one shown Example 4.3.3, where the classifier assigned only SNT tags to all input tokens, although a `sentiment` in our original corpus annotation could only appear in the presence of a `target` element.

Example 4.3.3 (An Error Made by the CRF System)
Gold Labels: So/SNT muss/SNT das/SNT sein/SNT %PosSmiley/SNT piraten+/TRG
That/SNT 's/SNT the/SNT way/SNT how/SNT it/SNT 's/SNT supposed/SNT to/SNT be/SNT %PosSmiley/SNT piraten+/TRG

Predicted Labels: So/SNT muss/SNT das/SNT sein/SNT %PosSmiley/SNT piraten+/SNT
That/SNT 's/SNT the/SNT way/SNT how/SNT it/SNT 's/SNT supposed/SNT to/SNT be/SNT %PosSmiley/SNT piraten+/TRG

4.4 Recurrent Neural Networks

A competitive alternative to CRFs is deep recurrent neural networks (RNNs). Introduced in the mid-nineties (Hochreiter and Schmidhuber, 1997), RNNs have become one of the most popular trends in the raging tsunami of deep learning applications, demonstrating superior results on many important NLP tasks including part-of-speech tagging (Wang et al., 2015a), dependency parsing (Kiperwasser and Goldberg, 2016), and machine translation (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014; Sutskever et al., 2014). Key factors that account for this success are

1) *the ability of RNN systems to learn optimal feature representations automatically*, which favorably sets them apart from traditional supervised machine-learning frameworks, such as SVMs or CRFs, where all features need to be defined by the user; and

2) *the ability to deal with arbitrary sequence lengths*, which advantageously distinguishes these methods from other NN architectures, such as plain feed-forward networks or convolutional systems without pooling, where the size of the input layer has to be constant.

The main component that underlies any modern RNN approach is a fixed-size hidden vector $\vec{h}$, which is recurrently updated during the analysis of an input sequence $\mathbf{x}$ and is meant to encode the meaning of that sequence. The general form of this vector at input state t is usually defined as:

$$\vec{h}^{(t)} = f(\vec{h}^{(t-1)}, \mathbf{x}^{(t)});$$

where f represents some non-linear transformation function, $\vec{h}^{(t-1)}$ denotes the state of the hidden vector at the previous time step, and $\mathbf{x}^{(t)}$ is the input vector at position t.

LSTM. A fundamental problem that arises from the above definition is that the gradients of model's parameters rapidly vanish to zero or explode to infinity (depending on whether the absolute values of $\vec{h}$ are less or greater than one) as the length of the input sequence increases. In order to solve this issue, Hochreiter and Schmidhuber (1997) proposed the long short-term memory mechanism (LSTM), in which they explicitly incorporated the goal of keeping the gradients within an appropriate range. In particular, given an input sequence $\mathbf{x}$, they introduced a special *activation unit* $\vec{i}^{(t)}$:

$$\vec{i}^{(t)} = \sigma\left(W_i \cdot \mathbf{x}^{(t)} + U_i \cdot \vec{h}^{(t-1)} + \vec{b_i}\right);$$

where σ denotes the sigmoid function; W_i, U_i, and $\vec{b_i}$ represent model's parameters; $\mathbf{x}^{(t)}$ stands for the input state; and $\vec{h}^{(t-1)}$ means the previous hidden state. In addition to the

activation unit, the authors also estimated a dedicated *forget gate* $\vec{f}^{(t)}$:

$$\vec{f}^{(t)} = \sigma\left(W_f \cdot \mathbf{x}^{(t)} + U_f \cdot \vec{h}^{(t-1)} + \vec{b}_f\right),$$

which is used to erase parts of the previous input that appear to be irrelevant.

After computing an *intermediate update state* $\widetilde{c}^{(t)}$ for the current time step t:

$$\widetilde{c}^{(t)} = \tanh\left(W_c \cdot \mathbf{x}^{(t)} + U_c \cdot \vec{h}^{(t-1)} + \vec{b}_c\right),$$

they estimated the *final update* $\vec{c}^{(t)}$ by taking a weighted sum of the candidate update vector $\widetilde{c}^{(t)}$ and the previous update value $\vec{c}^{(t-1)}$:

$$\vec{c}^{(t)} = \vec{i}^{(t)} \odot \widetilde{c}^{(t)} + \vec{f}^{(t)} \odot \vec{c}^{(t-1)};$$

from which, they finally computed the output vector $\vec{o}^{(t)}$ and the new value of the hidden state $\vec{h}^{(t)}$:

$$\begin{aligned}
\vec{o}^{(t)} &= \sigma\left(W_o \cdot \mathbf{x}^{(t)} + U_o \cdot \vec{h}^{(t-1)} + V_o \cdot \vec{c}^{(t)} + \vec{b}_o\right), \\
\vec{h}^{(t)} &= \vec{o}^{(t)} \odot \tanh(\vec{c}^{(t)}).
\end{aligned}$$

GRU. Despite their enormous popularity (*e.g.*, Filippova et al., 2015; Ghosh et al., 2016; Rao et al., 2016), LSTMs have been criticized for the high complexity of their recurrent unit. In order to overcome this deficiency, while still keeping the gradients within an acceptable range, Cho et al. (2014) proposed an alternative architecture called Gated Recurrent Units (GRU). In this framework, the authors also used activation and forget gates ($\vec{i}^{(t)}$ and $\vec{f}^{(t)}$) similar to the ones defined by Hochreiter and Schmidhuber (1997):

$$\begin{aligned}
\vec{i}^{(t)} &= \sigma\left(W_i \cdot \mathbf{x}^{(t)} + U_i \cdot \vec{h}^{(t-1)} + \vec{b}_i\right), \\
\vec{f}^{(t)} &= \sigma\left(W_f \cdot \mathbf{x}^{(t)} + U_f \cdot \vec{h}^{(t-1)} + \vec{b}_f\right).
\end{aligned}$$

With the help of these gates, they estimated the candidate activation $\widetilde{c}^{(t)}$ as:

$$\widetilde{c}^{(t)} = tanh\left(W_c \cdot \mathbf{x}^{(t)} + U_c \cdot \left(\vec{f}^{(t)} \odot \vec{h}^{(t-1)}\right) + \vec{b}_c\right),$$

and computed the hidden state $\vec{h}^{(t)}$ as:

$$\vec{h}^{(t)} = \vec{i}^{(t)} \odot \vec{h}^{(t-1)} + \left(\vec{1} - \vec{i}^{(t)}\right) \odot \widetilde{c}^{(t)}.$$

Final Layer. Because the output vectors of these recurrences ($\vec{o}^{(t)}$ in the LSTM case, and $\vec{h}^{(t)}$ in the case of GRU) do not strictly represent label probabilities (since elements of these vectors can also be negative and typically do not sum to one), and, moreover, because the size of our tagset (four tags: SNT, SRC, TRG, and NON) was obviously too small for the

size of the hidden unit, we set the dimensionality of the intermediate RNN vectors to 100, and apply a linear transformation matrix $O \in \mathbb{R}^{4\times100}$ to the final output of the recursion loop, computing the softmax of their dot product:

$$\vec{p}^{(t)} = softmax\left(O \cdot \vec{o}^{(t)}\right).$$

and considering the greatest value in the resulting vector as the probability of the most likely tag.

Training. A neat property of LSTM and GRU is that the final equation, which is obtained after unrolling the recurrence loop, is differentiable with respect to all of its parameters, and can therefore be optimized with standard gradient update techniques. Since most of these parameters, however, represent high-dimensional matrices or vectors, finding an optimal learning rate (*i.e.*, the size of the update step taken in the direction of the gradient) might pose considerable difficulties, leading either to prohibitively large training times (if the steps are too small) or complete divergence of the trained model (if the steps are too large).

Several algorithms have been proposed for solving this problem, including the method of momentum (Rumelhart et al., 1988), AdaGrad (Duchi et al., 2011), AdaDelta (Zeiler, 2012), RMSProp (Tieleman and Hinton, 2012), etc. In our experiments, we used the last of these options—RMSProp (Tieleman and Hinton, 2012)—as this algorithm showed both a faster convergence and better classification results.

Another important factor that could significantly influence the results were initial values of models' parameters. As shown by He et al. (2015), an inappropriate initialization of neural network might lead to a complete stalling of the whole learning process. Following recommended practices (Saxe et al., 2013), we used orthogonal initialization for all linear transformation matrices, and applied uniform He sampling (He et al., 2015) for setting the initial values of bias vectors.

Finally, due to a high imbalance of the target classes in the training set (where most of the instances represent objective statements without any sentiment tags), we "upsampled" sentiment tweets (*i.e.*, we randomly repeated microblogs containing `sentiments` until we reached an equal proportion of subjective and objective messages), and chose the *hinge-loss* as the optimized objective function L:[6]

$$L = \sum_{i}^{N} \sum_{t=0}^{|\mathbf{x}_i|} \max\left(0, c + \max_{y'\neq y} \vec{p}_{t,y'} - \vec{p}_{t,y}\right) + \alpha \left\|O\right\|_2^2, \tag{4.5}$$

[6]Since most of the tokens in the over-sampled training set still have the NON tag, the easiest way for a classifier to minimize the objective function is to always predict this tag with a very high confidence. We hoped to mitigate this effect by using the hinge-loss, since this function only penalizes incorrectly predicted labels or correct tags whose probability is insufficiently high (less than c), but does not reward any over-confident decisions.

Data Set	Sentiment			Source			Target			Macro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1
LSTM										
Training Set	$0.49^{\pm 0.16}$	$0.75^{\pm 0.01}$	$0.58^{\pm 0.13}$	$0.45^{\pm 0.05}$	$0.63^{\pm 0.12}$	$0.52^{\pm 0.08}$	$0.41^{\pm 0.11}$	$0.73^{\pm 0.06}$	$0.52^{\pm 0.11}$	$0.54^{\pm 0.11}$
Test Set	$0.29^{\pm 0.03}$	$\mathbf{0.31}^{\pm 0.11}$	$\mathbf{0.29}^{\pm 0.03}$	$\mathbf{0.25}^{\pm 0.02}$	$\mathbf{0.31}^{\pm 0.0}$	$\mathbf{0.27}^{\pm 0.01}$	$\mathbf{0.23}^{\pm 0.02}$	$\mathbf{0.25}^{\pm 0.05}$	$\mathbf{0.24}^{\pm 0.01}$	$\mathbf{0.27}^{\pm 0.02}$
GRU										
Training Set	$0.51^{\pm 0.08}$	$0.66^{\pm 0.05}$	$0.57^{\pm 0.03}$	$0.42^{\pm 0.03}$	$0.62^{\pm 0.05}$	$0.5^{\pm 0.03}$	$0.47^{\pm 0.11}$	$0.63^{\pm 0.11}$	$0.52^{\pm 0.04}$	$0.53^{\pm 0.03}$
Test Set	$\mathbf{0.3}^{\pm 0.01}$	$0.26^{\pm 0.06}$	$0.28^{\pm 0.03}$	$0.22^{\pm 0.03}$	$0.28^{\pm 0.02}$	$0.24^{\pm 0.02}$	$0.24^{\pm 0.03}$	$0.21^{\pm 0.07}$	$0.22^{\pm 0.03}$	$0.25^{\pm 0.01}$

Table 4.4: Results of fine-grained sentiment analysis with recurrent neural networks

where $\vec{p}_{t,y'}$ stands for the probability of the most likely wrong tag y' at position t in the training instance $\mathbf{x}_i$, $\vec{p}_{t,y}$ represents the probability of the gold label, and $\|O\|_2^2$ stands for the $L2$-norm of the O matrix.

We optimized the scalar hyper-parameters c and α on the development set, and trained the final model for 256 epochs, choosing parameter values that maximized the macro-averaged F_1-score on the development set.

Inference. Since each of the above approaches (LSTM and GRU) explicitly defines an output unit, the inference of the most likely label assignment for an input instance $\mathbf{x}$ is straightforward and amounts to finding the argmax value of the output vector at each time step of the recurrence:

$$\hat{\mathbf{y}} = \operatorname{argmax} \vec{p}^{(1)}, \operatorname{argmax} \vec{p}^{(2)}, \ldots, \operatorname{argmax} \vec{p}^{(|\mathbf{x}|)}.$$

Results. To account for the random factors in the initialization, we repeated each training experiment three times, and show the mean and the standard deviation of these results in Table 4.4.

As we can see from the table, the LSTM model shows generally better scores than the GRU system on both training and test sets. The only aspect at which it yields slightly worse results than the latter approach is precision of `sentiments`, which, however, is more than compensated for by a much higher recall. Moreover, the overfitting effect is significantly less pronounced than in the CRF case (where the F_1-scores on the training and test data differed by a factor of three). Nonetheless, both RNN systems achieve lower results than the linear-chain CRFs, which indicates the fact that the learned features still cannot capture the full extent of information that a human expert can encode with manually defined attributes.

RNN	Sentiment			Source			Target			Macro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1
Task-Specific Embeddings										
LSTM	0.283	0.288	0.278	**0.293**	0.372	0.328	**0.254**	0.27	**0.259**	0.288
GRU	0.287	0.246	0.263	0.287	0.405	**0.335**	0.252	0.205	0.216	0.271
Least-Squares Embeddings										
LSTM	0.268	**0.37**	0.307	0.261	**0.414**	0.314	0.223	**0.275**	0.245	**0.289**
GRU	0.256	0.341	0.291	0.267	0.395	0.318	0.229	0.262	0.245	0.285
word2vec Embeddings										
LSTM	**0.291**	0.329	**0.309**	0.2	0.311	0.244	0.221	0.219	0.22	0.257
GRU	0.273	0.355	0.301	0.207	0.353	0.257	0.213	0.26	0.233	0.264

Table 4.5: Results of fine-grained sentiment analysis with different word embeddings

4.4.1 Word Embeddings

To see whether using different embeddings would improve the results of the tested methods, we reran our experiments with two alternative embedding types:

- *word2vec vectors* (Mikolov et al., 2013), which had been pretrained on the German Twitter snapshot (Scheffler, 2014) and were kept fixed during the RNN optimization;
- and *least-squares embeddings*, which were previously described in Chapter 3;

subsequently evaluating all systems on the development set.

The results of this evaluation are shown in Table 4.5. As we can see from the scores, least-squares representations significantly improve the recall of all classes, which, in turn, leads to much higher macro-averaged F_1-measures in comparison with other embeddings. The task-specific variant shows second-best results, mainly due to a higher precision of `targets` and `sources`. Finally, word2vec vectors also improve the prediction of `sentiment` spans, but otherwise cause a notable degradation of literally every other aspect.

4.4.2 Error Analysis

As in the previous case, we also decided to have a closer look at some sample errors, which were committed by the tested systems. As it turned out, the most common type of mistakes made by both classifiers was confusion of NON labels with other tags, which we also can see in Examples 4.4.1 and 4.4.2.

Example 4.4.1 (An Error Made by the LSTM System)
Gold Labels: Meine/NON Mama/NON liest/NON bei/NON Twitter/NON mit/NON

My/NON mom/NON is/NON reading/NON Twitter/NON together/NON with/NON me/NON

Predicted Labels: Meine/TRG Mama/NON liest/NON bei/NON Twitter/NON mit/NON
My/TRG mom/NON is/NON reading/NON Twitter/NON together/NON with/NON me/NON

Example 4.4.2 (An Error Made by the GRU System)
Gold Labels: Ich/NON habe/NON das/NON "noch"/NON vergessen/NON
I/NON have/NON forgotten/NON the/NON "still"/NON

Predicted Labels: Ich/SRC habe/NON das/TRG "noch"/NON vergessen/NON
I/SRC have/NON forgotten/NON the/TRG "still"/NON

The obvious reason for these wrong predictions was the upsampling of sentiment tweets that we used to balance the class distribution in the training data. Unfortunately, switching this component off caused all classifiers to always predict only the NON tag and significantly worsened the scores of these approaches in comparison with our initial experiments.

4.5 Evaluation

After estimating the results of popular FGSA approaches with their (mostly) standard settings, evaluating their specific components (features and word embeddings), and looking at their sample errors, we also decided to investigate the impact of common factors, such as annotation scheme, graph structure, and text normalization on the net results of these methods. For this purpose, we reran the evaluation, changing one aspect of the training procedure at a time, and re-estimated the scores of these systems on the development set. The results of these experiments are presented below.

4.5.1 Annotation Scheme

As the first factor that could affect the quality of automatic FGSA methods, we considered the annotation scheme that we used to create the corpus. As described in Section 2.2, we initially asked our experts to assign the `sentiment` label to complete syntactic or discourse-level

units that included both the target of an opinion and its immediate evaluative expression. Even though this decision was linguistically plausible and helpful for determining the boundaries of `sentiments` and their relevant components, it also posed considerable difficulties for sequence labeling techniques, since `sentiment` tags were assigned not only to the immediate polar terms but also to neutral words that occurred within the same syntactic constituent as the polar item and its target. Since none of the tested methods could explicitly incorporate this logic, we decided to check whether an alternative interpretation of the annotation scheme could alleviate their inference.

In particular, instead of unconditionally labeling all words belonging to a `sentiment` span in the original annotation with the SNT tag as we did previously (which we call a *broad* interpretation of the annotation scheme), we only assigned this label to the polar terms found in the corpus (which we call a *narrow* interpretation). The difference between these two takes is shown in Examples 4.5.1 and 4.5.2.

Example 4.5.1 (Broad Sentiment Interpretation)

[[*Francis*]$_{target}$ *makes a* [*very*]$_{intensifier}$ [*good*]$_{polar\text{-}term}$ *impression on* [*me*]$_{source}$*!* [*:)*]$_{polar\text{-}term}$]$_{sentiment}$

→

Francis/TRG makes/SNT a/SNT very/SNT good/SNT impression/SNT on/SNT
me/SRC !/SNT :)/SNT

Example 4.5.2 (Narrow Sentiment Interpretation)

[[*Francis*]$_{target}$ *makes a* [*very*]$_{intensifier}$ [*good*]$_{polar\text{-}term}$ *impression on* [*me*]$_{source}$*!* [*:)*]$_{polar\text{-}term}$]$_{sentiment}$

→

Francis/TRG makes/NON a/NON very/NON good/SNT impression/NON on/NON
me/SRC !/NON :)/SNT

In the former (broad) case, we labeled the whole subjective sentence with the SNT tag except for the words that denoted the target and source of the opinion. In the latter (narrow) case, we only assigned the SNT tag to the polar term "good" and the emoticon ":)," which, however, were expressive enough to convey the main evaluative sense of the whole subjective statement.

The results of the automatic systems with these two approaches are given in Table 4.6. As we can see from the table, the broad interpretation generally leads to notably lower

Method	Sentiment			Source			Target			Macro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1
Broad Interpretation										
CRF	0.38	0.32	0.34	**0.3**	0.33	0.31	**0.29**	0.23	**0.26**	0.31
LSTM	0.28	0.29	0.28	0.29	**0.37**	**0.33**	0.25	**0.27**	**0.26**	0.29
GRU	0.29	0.25	0.26	0.29	0.4	0.34	0.25	0.21	0.22	0.27
Narrow Interpretation										
CRF	0.59	0.64	0.62	0.26	0.23	0.24	0.22	0.20	0.21	0.36
LSTM	**0.62**	**0.65**	**0.63**	**0.3**	0.35	0.32	0.26	0.14	0.18	**0.38**
GRU	**0.62**	0.63	0.62	0.28	0.33	0.3	0.23	0.24	0.23	**0.38**

Table 4.6: Results of fine-grained analysis with broad and narrow sentiment interpretations

scores for `sentiment` spans, but yields much better results for their `source`s and `target`s. An opposite situation is observed with the narrow scheme: even though the F_1-values for `sentiment`s are twice as high as in the broad case, the scores for the remaining elements are up to seven percent lower.

An obvious explanation for these results is the expected better amenability of the narrow scheme to the prediction of `sentiment` labels: since `sentiment` tags are only assigned to obvious polar terms, it becomes easier for the models to infer this class using their state features, especially morphological or lexical ones, or word embeddings. But, on the other hand, such short spans lead to disrupted label chains for other opinion-related elements, setting `sentiment` tags far apart from the spans of their respective `source`s and `target`s. As a consequence, these classes suffer from the lack of context and become heavily dependent on the state attributes as well. But, this time, the effect of state features is rather negative, because in contrast to `polar term`s, being a source or a target of an opinion is not an inherent property of the lexical term, but arises solely from the context which this term appears in.

Consider, for instance, the name "Silvio Berlusconi" in Example 4.5.3, where it appears as the target of a sentiment in the first sentence (which expresses author's hope that Silvio Berlusconi will not be the new Pope), but serves as a normal subject of an objective clause in the second case. The decision about the role of this name depends primarily on the sense of the whole statement rather than the name itself. Consequently, state attributes might only increase our prior belief that certain words would rather appear in a subjective context, but cannot tell for sure whether they actually do so or not.[7] As a consequence, prediction of `source`s and `target`s becomes much harder when they do not have enough context information.

[7]The negative effect of state features on prediction of `source`s and `target`s was actually observed in our corpus, where one of the most frequently made mistakes was the unconditional assignment of the TRG tag to the word "Nordkorea" (*North Korea*) regardless of its surrounding context.

Example 4.5.3 (Contextual Dependence of Target Elements)
Hoffentlich ist es nicht [*Silvio Berlusconi*]$_{target}$. *#Papst*

Hopefully, this won't be [*Silvio Berlusconi*]$_{target}$. *#Pope*

Silvio Berlusconi ist ein italienischer Medienmagnat und Politiker.

Silvio Berlusconi is an Italian media tycoon and politician.

4.5.2 Graph Structure

Since the lack of contextual links played an important role for prediction of `sources` and `targets`, we decided to investigate whether redefining the way these links were established in the models would improve the results. For this purpose, we implemented three possible extensions to the traditional first-order linear-chain CRFs, which are shown in Figure 4.3:

- higher-order linear-chain CRFs,
- first- and higher-order semi-Markov models, and
- tree-structured CRFs.[8]

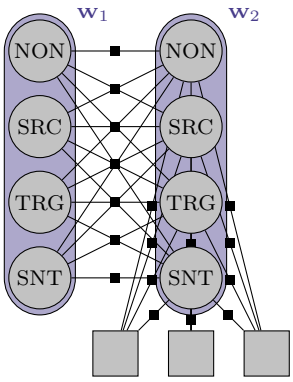
The results of these systems on the training and development sets are shown in Table 4.7.

Element	**Structure**								
	lcCRF^1	lcCRF^2	lcCRF^3	lcCRF^4	smCRF^1	smCRF^2	smCRF^3	smCRF^4	trCRF^1
Training Set									
Sentiment	0.928	0.919	0.922	0.925	0.931	0.931	0.933	0.931	0.906
Source	0.887	0.876	0.89	0.901	0.869	0.886	0.874	0.878	0.881
Target	0.898	0.811	0.816	0.827	0.813	0.827	0.815	0.817	0.876
Development Set									
Sentiment	0.345	0.334	0.332	0.335	**0.395**	0.385	0.389	0.378	0.331
Source	0.313	**0.32**	0.272	0.304	0.298	0.282	0.287	0.291	0.223
Target	0.258	0.235	0.24	0.229	0.287	**0.309**	0.301	0.292	0.243

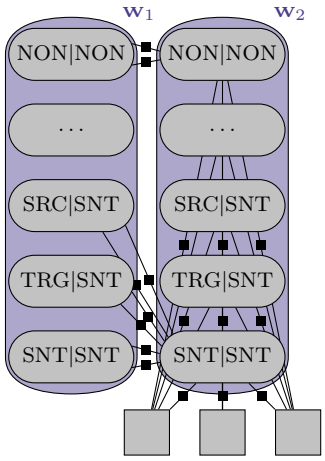
Table 4.7: Results of fine-grained sentiment analysis with different CRF topologies lcCRF—linear-chain CRFs, smCRF—semi-Markov CRFs, trCRF—tree-structured CRFs; 1, 2, 3, and 4 in the superscripts denote the order

As we can see from the scores, semi-Markov CRFs achieve better results at predicting `sentiments` and `targets`, but show a degradation when classifying `sources` of sentiments.

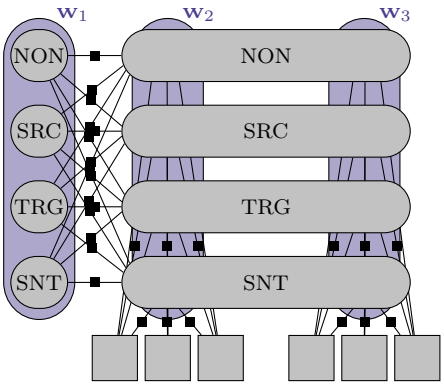
[8]The training and inference algorithms of these CRF variants are described in Appendix C.



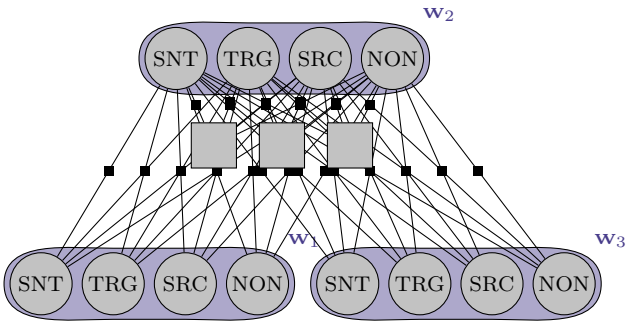
(a) First-order linear-chain CRF



(b) Second-order linear-chain CRF



(c) Semi-Markov CRF



(d) Tree-structured CRF

Figure 4.3: Factor graphs of different CRF structures
(circles represent random variables; gray boxes denote observed input; factors [i.e., feature functions] are shown as tiny black squares)

Furthermore, second-order semi-Markov and linear-chain structures outperform the first-order models at classifying `targets` and `sources`, but further increasing the order of these structures does not bring about any improvements. Somewhat surprisingly, tree-structured CRFs show even worse scores than their linear counterparts.

In order to see whether the same tendencies would hold for deep-learning methods, we also implemented higher-order and tree-structured extensions of LSTM and GRU. In the former case, we passed a concatenation of n preceding $\vec{h}$ vectors (where n is the order of the model) as input to the recurrence loop. In the tree-structure modification, we followed the

approach of Tai et al. (2015) and defined the LSTM unit as follows:

$$
\begin{aligned}
\tilde{h}^{(t)} &= \sum_{k \in C(t)} \vec{h}^{(k)}, \\
\vec{i}^{(t)} &= \sigma\left(W_i \cdot \vec{x}^{(t)} + U_i \cdot \tilde{h}^{(t)} + \vec{b}_i\right), \\
\vec{o}^{(t)} &= \sigma\left(W_o \cdot \vec{x}^{(t)} + U_o \cdot \tilde{h}^{(t)} + \vec{b}_o\right), \\
\vec{u}^{(t)} &= \sigma\left(W_o \cdot \vec{x}^{(t)} + U_o \cdot \tilde{h}^{(t)} + \vec{b}_u\right), \\
\vec{f}^{(t,k)} &= \sigma\left(W_f \cdot \vec{x}^{(t)} + U_f \cdot \vec{h}^{(k)} + \vec{b}_f\right), \\
\vec{c}^{(t)} &= \vec{i}^{(t)} \odot \vec{u}^{(t)} + \sum_{k \in C(t)} f^{(t,k)} \odot c^{(k)}, \\
\vec{h}^{(t)} &= \vec{o}^{(t)} \odot tanh\left(\vec{c}^{(t)}\right);
\end{aligned}
$$

where $C(t)$ stands for the indices of all child nodes of the token t.

In a similar way, we also redefined the GRU unit to the following solutions:

$$
\begin{aligned}
\tilde{h}^{(t)} &= \sum_{k \in C(t)} \vec{h}^{(k)}, \\
\vec{i}^{(t)} &= \sigma\left(W_i \cdot \mathbf{x}^{(t)} + U_i \cdot \tilde{h}^{(t)}\right), \\
\vec{f}^{(t,k)} &= \sigma\left(W_f \cdot \mathbf{x}^{(t)} + U_f \cdot \vec{h}^{(t,k)}\right), \\
\widetilde{c}^{(t)} &= tanh\left(W_c \cdot \mathbf{x}^{(t)} + U_c \cdot \sum_{k \in C(t)} \left(\vec{f}^{(t,k)} \odot \vec{h}^{(k)}\right)\right), \\
\vec{h}^{(t)} &= \vec{i}^{(t)} \odot \tilde{h}^{(t)} + \left(\vec{1} - \vec{i}^{(t)}\right) \odot \widetilde{c}^{(t)}.
\end{aligned}
$$

The results of these modifications are shown in Table 4.8, from which we can see that first-order LSTM still outperforms all higher-order LSTM and GRU variants at predicting `targets` and `sources` of opinions. Furthermore, first-order GRU also achieves the best scores on predicting `sentiment` spans among all compared models. This time, again, none of the tree-structured extensions can outperform the linear-chain systems, which might be partially explained by the errors produced by the parser, whose original target domain is standard-language news texts.

4.5.3 Text Normalization

Another question that remained open in the previous experiments was whether the input passed to the models actually had to be normalized or not. As mentioned in Section 4.2, when preparing the data, we preprocessed all corpus tweets using the rule-based normalization procedure of Sidarenka et al. (2013).

Element	Structure							
	lcLSTM^1	lcLSTM^2	lcLSTM^3	lcGRU^1	lcGRU^2	lcGRU^3	trLSTM^1	trGRU^1
Training Set								
Sentiment	0.584	0.559	0.54	0.57	0.587	0.606	0.43	0.518
Source	0.525	0.458	0.424	0.503	0.546	0.548	0.317	0.372
Target	0.521	0.513	0.501	0.519	0.544	0.605	0.305	0.425
Development Set								
Sentiment	0.278	0.285	0.281	**0.335**	0.252	0.253	0.314	0.292
Source	**0.328**	0.314	0.303	0.263	0.298	0.306	0.256	0.262
Target	**0.259**	0.218	0.222	0.216	0.219	0.188	0.205	0.193

Table 4.8: Results of fine-grained sentiment analysis with different neural network topologies lcLSTM—linear-chain LSTM, lcGRU—linear-chain GRU, trLSTM—tree-structured LSTM, trGRU—tree-structured GRU;
1, 2, and 3 in the superscripts denote the order

Even though these transformations were supposed to improve the grammaticality of sentences, an opposite consequence of this normalization was the loss of (potentially valuable) surface features. In order to check which of these effects had a stronger influence on the FGSA results, we repeated the evaluation once again, turning the preprocessing pipeline off this time.

Data Set	Sentiment			Source			Target			Macro F_1
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	
w Normalization										
CRF	**0.376**	**0.319**	**0.345**	**0.298**	0.33	0.313	**0.293**	0.231	0.258	**0.305**
LSTM	0.283	0.288	0.278	0.293	0.372	0.328	0.254	**0.27**	**0.259**	0.288
GRU	0.287	0.246	0.263	0.287	**0.405**	**0.335**	0.252	0.205	0.216	0.271
w/o Normalization										
CRF	0.301	0.278	0.289	0.276	0.3	0.287	0.255	0.23	0.242	0.273
LSTM	0.274	0.252	0.261	0.284	0.367	0.32	0.237	0.241	0.237	0.273
GRU	0.266	0.245	0.252	0.296	0.369	0.328	0.232	0.268	0.245	0.275

Table 4.9: Results of fine-grained sentiment analysis with (w) and without (w/o) text normalization

As we can see from the results in Table 4.9, text preprocessing clearly helps sentiment classification, as all of the best observed results are achieved exclusively with normalized text. The only aspect that benefits from keeping the input unchanged is precision of `target` classification with GRU, which, in turn, leads to a slightly higher (+0.004) macro-averaged F_1-score for this system. Apart from that, all other aspects and classifiers show a notable degradation when the preprocessing module is switched off.

4.6 Summary and Conclusions

Summarizing these findings, we would like to remind the reader that in this chapter we have evaluated two most common approaches to fine-grained sentiment analysis: conditional random fields and recurrent neural networks. Our experiments showed that CRFs with manually defined features outperform both recurrent neural networks (LSTM and GRU), reaching a macro-averaged F_1-score of 0.287 on predicting `sentiment`s, `source`s, and `target`s.

Furthermore, a closer look at these systems revealed that:

- CRFs can learn meaningful weights for state- and transition-features, although different features types might have different effects on classification of opinion elements: whereas `sentiment`s benefited from all features used in our experiments, `source`s profited most from lexical and complex attributes, and `target`s were positively influenced by morphological and syntactic features only;
- Apart from that, we analyzed the effect of different embedding types on the net results of RNN systems, finding that least-squares embeddings yield the best overall scores for these methods;
- Furthermore, even higher prediction scores for `sentiment`s can be achieved by narrowing the spans of these elements to polar terms. This, however, might negatively affect the classification of `source`s and `target`s;
- Even though context seems to play an important role, redefining models' structures by increasing the order of their dependencies or performing inference over trees instead of linear chains does not bring much improvement. We could, however, still outperform the results of traditional first-order linear-chain CRFs with their first- and second-order semi-Markov modifications;
- In the final step, we estimated the effect of text normalization by rerunning all experiments with original (unnormalized) tweets. This test showed that preprocessing is an extremely helpful procedure, which might improve the results of FGSA methods by up to 3%.

Chapter 5

Message-Level Sentiment Analysis

Having familiarized ourselves with the peculiarities of the creation of a sentiment corpus, the different ways to automatically induce new polarity lists, and the difficulties of fine-grained opinion mining, we now move on to the presumably most popular sentiment analysis task—message-level sentiment analysis or MLSA, in which we need to determine the overall polarity of a message.

Traditionally, this objective is addressed with either of the three popular method groups:

- lexicon-based approaches,
- machine-learning–based (ML) techniques,
- and deep-learning–based (DL) systems.

In this chapter, we are going to scrutinize the most successful representatives of each of these paradigms, propose our own solution, and also analyze errors, the utility of single components, and the effect of additional training factors on the net results of these methods.

We begin our comparison by first presenting two metrics that we will use in our subsequent evaluation. After briefly describing the data preparation step, we proceed to the actual estimation of popular lexicon-, ML-, and DL-based approaches, explaining and evaluating them in Sections 5.3, 5.4, and 5.5. Finally, we conclude with an extensive evaluation of different hyperparameters and settings (including the impact of additional noisily labeled training data, various types of sentiment lexicons, and text normalization), summarizing our results and recapping our findings at the end of this part.

5.1 Evaluation Metrics

To estimate the quality of compared systems, we will rely on two established evaluation metrics that are commonly used to measure MLSA results: The first of these metrics is the *macro-averaged F_1-score* over two main polarity classes (positive and negative):

$$F_1 = \frac{F_{pos} + F_{neg}}{2}.$$

This measure was first introduced by the organizers of the SemEval competition (Nakov et al., 2013; Rosenthal et al., 2014, 2015) and has become a de facto standard not only for the SemEval dataset but virtually for all related message-level sentiment corpora and tasks. This score is supposed to emphasize the ability of a classifier to distinguish between opposite semantic orientations. Although it seemingly ignores the neutral class, this type of misclassifications is indirectly taken into account as well, because confusing the neutral label with another polarity will automatically pull down the values of F_{pos} or F_{neg}.

The second metric, *micro-averaged F_1-score*, explicitly considers all three semantic orientations (positive, negative, and neutral) and essentially corresponds to the prediction accuracy on the complete dataset (see Manning and Schütze, 1999, p. 577). This measure both predates and supersedes the SemEval evaluation as it had already been used in the very first works on sentence-level opinion mining (Wiebe et al., 1999; Das and Chen, 2001; Read, 2005; Kennedy and Inkpen, 2006; Go et al., 2009) and was reintroduced again at the GermEval shared task in 2017 (Wojatzki et al., 2017).

Besides these two metrics, we will also give a detailed information about precision, recall, and F_1-scores for each particular polarity class.

5.2 Data Preparation

As in the previous experiments, we preprocessed all tweets labeled by the second annotator with the text normalization system of Sidarenka et al. (2013), tokenized them using the same adjusted version of Potts' tokenizer,[1] lemmatized and assigned part-of-speech tags to these tokens with the TreeTagger of Schmid (1995), and obtained morphological features and syntactic analyses with the `Mate` dependency parser (Bohnet et al., 2013).

We again divided our corpus into training, development, and test sets, using 70% of the tweets for learning, 10% for tuning and picking optimal model parameters, and the remaining 20% for evaluating the results. Drawing on the work of Wiebe and Riloff (2005), we inferred the polarity of these microblogs, which we will consider as gold labels in our experiments,

[1] `http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py`

using a simple heuristic rule in which we assigned the positive (negative) class to the messages that had exclusively positive (negative) annotated `sentiments`, skipping all microblogs that simultaneously contained multiple labeled opinions with different semantic orientations (178 tweets). In the cases when there was no `sentiment`, we recoursed to a fallback strategy by considering all tweets that contained exclusively positive (negative) annotated `polar terms` as positive (negative), and ignoring all messages that featured polar elements from both polarity classes (335 messages).[2] Finally, all microblogs without any `sentiments` or `polar terms` were regarded as neutral.

A few examples of such heuristically inferred labels are provided below:

Example 5.2.1 (Message-Level Sentiment Annotations)
Tweet: [Ich finde den Papst [putzig]$_{\textit{polar-term:polarity=positive}}$
[☺]$_{\textit{polar-term:polarity=positive}}$]$_{\textit{sentiment:polarity=positive}}$
[*I find the Pope* [*cute*]$_{\textit{polar-term:polarity=positive}}$
[☺]$_{\textit{polar-term:polarity=positive}}$ ·]$_{\textit{sentiment:polarity=positive}}$
Label: positive

Tweet: [typisch]$_{\textit{polar-term:polarity=negative}}$ Bayern kaum ist der neue Papst da und schon haben sie ihn [in der Tasche]$_{\textit{polar-term:polarity=negative}}$ ···
[*Typical*]$_{\textit{polar-term:polarity=negative}}$ *Bavaria The new Pope is hardly there, as they already have him* [*in their pocket*]$_{\textit{polar-term:polarity=negative}}$
Label: negative

As we can see from the examples, our simple rule makes fairly reasonable decisions, assigning the positive class to the first tweet, which also expresses a positive sentiment, and labeling the second message as negative, since it contains two negative polar terms ("typisch" [*typical*] and "in der Tasche haben" [*to have sb. in one's pocket*]).

But because our approach is still an approximation and consequently prone to errors (especially in the cases where the polarity of the whole microblog differs from the semantic orientation of its polar terms, as in the first tweet in Example 5.2.2, or when it is expressed without any explicit polar terms at all, as in the second microblog of this example), we decided to evaluate all MLSA methods also on another German Twitter corpus, SB10k (Cieliebak et al., 2017), which was introduced when we already started working on this chapter and which had been explicitly annotated with message-level polarities of the tweets.

[2]Note that we inferred all message-level labels based on *annotated* `sentiments` and `polar terms` and did not rely on the mere occurrence of positive or negative smileys, which not necessarily implied an expression of polarity.

Example 5.2.2 (Erroneous Sentiment Annotations)
Tweet: Unser Park, unser Geld, unsere Stadt! -NICHT unser Finanzminister! [☺]$_{polar\text{-}term:polarity=positive}$ #schmid #spd #s21 #btw13
Our park, our money, our city! -NOT our Finance Minister!
[☺]$_{polar\text{-}term:polarity=positive}$ *#schmid #spd #s21 #btw13*
Label: **positive***

Tweet: Auf die Lobby-FDP von heute kann Deutschland verzichten ...
Germany can go without today's lobby FDP
Label: **neutral***

The SB10k dataset comprises a total of 9,738 microblogs, which were sampled from a larger snapshot of 5M German tweets gathered between August and November 2013. To ensure lexical diversity and proportional polarity distribution in this corpus, the authors first grouped all posts of this snapshot into 2,500 clusters using the k-means algorithm with unigram features. Afterwards, from each of these groups, they selected tweets that contained at least one positive or one negative term from the German Polarity Clues lexicon (Waltinger, 2010). Each message was subsequently annotated by at least three human experts from a pool of 34 different annotators. The resulting inter-rater reliability (IRR) of this annotation run up to 0.39 Krippendorff's α (Krippendorff, 2007). Unfortunately, due to the restrictions of Twitter's terms of use, which only allow to distribute the ids of the microblogs and their labels, we could only retrieve 7,476 tweets of this collection, which, however, still represents a substantial part of the original dataset.

In addition to the aforementioned two corpora (PotTS and SB10k), we also automatically annotated all microblogs of the German Twitter Snapshot (Scheffler, 2014) by following the procedure of Read (2005) and Go et al. (2009) and assigning the positive (negative) class to the tweets that contained respective emoticons, regarding the rest of the microblogs as neutral. In contrast to the previous two datasets, whose labels were inferred or directly obtained from manual annotations, we will not use this automatically tagged corpus for evaluation, but will only harness it for training in our later weak-supervision experiments.

The resulting statistics on the number of messages and polarity class distribution in these data are shown in Table 5.1.

As we can see, each dataset has its own unique composition of polar tweets: The PotTS corpus, for example, shows a conspicuous bias towards the positive class, with 42% of its microblogs belonging to this polarity. We can partially explain this skewness by the selection criteria that we used to compile the initial data for this collection: Because a big part of this dataset was composed from tweets that contained smileys, and most of these emoticons were

Dataset	Polarity Class				Label Agreement	
	Positive	Negative	Neutral	Mixed*	α	κ
PotTS	3,380	1,541	2,558	513	0.66	0.4
SB10k	1,717	1,130	4,629	0	0.39	NA
GTS	3,326,829	350,775	19,453,669	73,776	NA	NA

Table 5.1: Polarity class distribution in PotTS, SB10k, and the German Twitter Snapshot (GTS) (* — the *mixed* polarity was excluded from our experiments)

positive, which is evident from the statistics of the German Twitter snapshot, the selected microblogs also got biased towards this semantic orientation.

The second most frequent group in the PotTS corpus are neutral tweets, which account for 32% of the data. Negative messages, vice versa, represent a clear minority in this collection (only 19%), which, however, is less surprising as the same tendency can be observed for SB10k and the German Twitter Snapshot too.

Regarding the last two corpora, we can observe a more uniform (though not identical) behavior, where both datasets are dominated by neutral posts, which constitute 62% of SB10k and 84% of all snapshot tweets. The positive class, again, makes up a big part of these data (23% of the former corpus and 14% of the latter dataset), but its influence this time is much less pronounced than in the PotTS case. Finally, negative tweets are again the least represented semantic orientation. The only group that has even less instances than this class is the MIXED polarity. We, however, will skip the mixed orientation in our experiments for the sake of simplicity and uniformity of evaluation.

5.3 Lexicon-Based Methods

The first group of approaches that we are going to explore in this chapter using the aforementioned data are lexicon-based (LB) systems. Just like sentiment lexicons themselves, LB methods for message-level opinion mining have attracted a lot of attention from the very inception of the sentiment analysis field. Starting from the work of Hatzivassiloglou and Wiebe (2000), who gave a statistical proof that the mere occurrence of a subjective adjective from an automatically compiled polarity list was a sufficiently reliable indicator that the whole sentence was subjective, more and more researchers started using lexicons in order to estimate the overall polarity of a text.

One of the first notable steps in this direction was made by Das and Chen (2001), who proposed an ensemble of five classifiers (two of which were purely lexicon-based and the other three heavily relied on lexicon features) to predict the polarity of stock messages, achieving

an accuracy of 62% on a corpus of several hundreds stock board messages. A much simpler method for a related task was suggested by Turney (2002), who determined the *semantic orientation* (SO) of reviews by averaging the PMI scores of their terms, getting these scores from an automatically generated sentiment lexicon. With this approach, the author could reach an accuracy of 74% on a corpus of 410 manually labeled Epinions comments. In the same vein, Hu and Liu (2004) computed the overall polarity of a sentence by comparing the numbers of its positive and negative terms, reversing their orientation if they appeared in a negated context.

In 2006, Polanyi and Zaenen presented an extensive overview and analysis of common lexicon-based sentiment methods that existed at that time, arguing that besides considering the lexical valence (*i.e.*, semantic orientation) of polar expressions, it was also important to incorporate syntactic, discourse-level, and extra-linguistic factors such as negations, intensifiers, modal operators (*e.g.*, *could* or *might*), presuppositional items (*e.g.*, *barely* or *failure*), irony, reported speech, discourse connectors, genre, attitude assessment, reported speech, and multi-entity evaluation. This theoretical hypothesis was also proven empirically by Kennedy and Inkpen (2006), who investigated two ways to determine the polarity of a customer review: In the first approach, the authors simply compared the numbers of positive and negative terms in the text, assigning the review to the class with the greater number of items. In the second attempt, they enhanced the original system with an additional information about contextual valence shifters, increasing or decreasing the sentiment score of a term if it was preceded by an intensifier or downtoner, and changing the polarity sign of this score to the opposite in case of a negation.

Finally, a seminal work on lexicon-based techniques was presented by Taboada et al. (2011), who introduced a manually compiled polarity list[3] and used this resource to estimate the overall semantic orientation of texts. Drawing on the ideas of Polanyi and Zaenen (2006), the authors incorporated a set of additional heuristic rules into their computation by changing the prior SO values of negated, intensified, and downtoned terms, ignoring irrealis and interrogative sentences, and adjusting the weights of specific document sections. An extensive evaluation of this approach showed that the manual lexicon performed much better than automatically generated polarity lists, such as Subjectivity Dictionary (Wilson et al., 2005), Maryland Polarity Set (Mohammad et al., 2009), and SentiWordNet of Esuli and Sebastiani (2006a). Moreover, the authors also demonstrated that their method could be successfully applied to other topics and genres, hypothesizing that lexicon-based approaches were in general more amenable to domain shifts than traditional supervised machine-learning techniques.

[3]The authors hand-annotated all occurrences of adjectives, nouns, and verbs found in a corpus of 400 Epinions reviews with ordinal categories ranging from -5 to 5 that reflected the semantic orientation of a term (positive vs. negative) and its polar strength (weak vs. strong).

It is therefore not surprising that lexicon-based systems have also quickly found their way into the sentiment analysis of social media: For example, one such approach, explicitly tailored to Twitter specifics, was proposed by Musto et al. (2014), who examined four different ways to compute the overall polarity scores of microblogs: *basic*, *normalized*, *emphasized*, and *normalized-emphasized*. In each of these methods, the authors first split the input message into a list of *micro-phrases* based on the occurrence of punctuation marks and conjunctions. Afterwards, they calculated the polarity score for each of these segments and finally estimated the overall polarity of the whole tweet by uniting the scores of its micro-phrases. Musto et al. obtained their best results (58.99% accuracy on the SemEval-2013 dataset) with the normalized-emphasized approach, in which they averaged the polarity scores of segments' tokens, boosting these values by 50% for adjectives, adverbs, nouns, and verbs; and computed the final overall polarity of the microblog by taking the sum of all micro-phrase scores.

Another Twitter-aware system was presented by Jurek et al. (2015), who computed the negative and positive polarity of a message (F_p and F_n respectively) as:

$$\begin{aligned} F_P &= \min\left(\frac{A_P}{2 - \log(3.5 \times W_P + I_P)}, 100\right), \\ F_N &= \max\left(\frac{A_N}{2 - \log(3.5 \times W_N + I_N)}, -100\right); \end{aligned} \tag{5.1}$$

where A_P and A_N represent the average scores of positive and negative lexicon terms found in the tweet; W_P and W_N stand for the raw counts of polar tokens; and I_P and I_N denote the number of intensifiers preceding these words. In addition to that, before estimating the average values, the authors modified the polarity scores s_w of all negated words w using the following rule:

$$neg(s_w) = \begin{cases} \min\left(\frac{s_w - 100}{2}, -10\right) & \text{if } s_w > 0, \\ \max\left(\frac{s_w + 100}{2}, 10\right), & \text{if } s_w < 0. \end{cases}$$

Furthermore, besides computing the polarity scores F_p and F_n, Jurek et al. also determined the subjectivity degree of the message by replacing the A_P and A_N terms in Equation 5.1 with the average of conditional probabilities of the tweet being subjective given the occurrences of the respective polar terms.[4] The authors considered a microblog as neutral if its absolute polarity was less than 25, and the subjectivity value was not greater than 0.5. Otherwise, they assigned a positive or negative label to this message depending on the sign of the polarity score. With this approach, Jurek et al. achieved an accuracy of 77.3% on the manually annotated subset of the Go et al.'s corpus and reached 74.2% on the IMDB review dataset (Maas et al., 2011).

Finally, Kolchyna et al. (2015) also explored two different ways of computing the overall polarity of a microblog: (i) by simply averaging the scores of all lexicon terms found in the

[4]These probabilities were calculated automatically on the noisily labeled data set of Go et al. (2009).

message and (ii) by taking a signed logarithm of this average:

$$\text{Score}_{\text{log}} = \begin{cases} \text{sign}(\text{Score}_{\text{AVG}}) \log_{10}(|\text{Score}_{\text{AVG}}|) & \text{if } |\text{Score}_{\text{AVG}}| > 0.1, \\ 0, & \text{otherwise;} \end{cases}$$

The authors determined the final polarity of a tweet by using k-means clustering, which utilized both of the above polarity values as features. They showed that the logarithmic strategy performed better than the simple average solution, yielding an accuracy of 61.74% on the SemEval-2013 corpus (Nakov et al., 2013).

As it was unclear how each of these methods would perform on PotTS and SB10k, we reimplemented the approaches of Hu and Liu (2004) (as a relatively simple baseline), Taboada et al. (2011), Musto et al. (2014), Jurek et al. (2015), and Kolchyna et al. (2015), and applied these systems to the test sets of these corpora.

Based on our comparison in Chapter 3, we chose the Zurich Polarity List (Clematide and Klenner, 2010) as the primary sentiment lexicon for the tested methods. However, a significant drawback of this resource is that most of its entries have uniform weights, with their polarity scores being either 0.7 or 1. We decided to keep the original values as is, and only multiplied the scores of negative terms by -1, since all of the tested approaches presupposed different signs for the terms with opposite semantic orientations.[5] Moreover, because some analyzers (*e.g.*, Taboada et al. [2011] and Musto et al. [2014]) required part-of-speech tags of lexicon entries, we automatically tagged all terms in this polarity list with the TreeTagger (Schmid, 1995), choosing the most probable part-of-speech tag for each entry and also using the tag sequences whose probabilities were at least two times lower than the likelihood of the best assignment, duplicating the lexicon entries in the second case.

Furthermore, since all of the systems except for that of Kolchyna et al. (2015) by default returned continuous real values, but our evaluation required discrete polarity labels (*positive*, *negative*, or *neutral*), we discretized the results of these approaches using the following simple procedure: We first determined the optimal threshold values for each particular polar class on the training and development sets,[6] and then derived polarity labels for the test messages by comparing their predicted SO scores with these thresholds. To achieve the former goal (*i.e.*, to find the optimal thresholds), we exhaustively searched through all unique polarity values assigned to the training and development instances and checked whether using these values as a boundary between two adjacent polarity classes (sorted in ascending order of their positivity) would increase the overall macro-F_1 on the training and development sets.

The final results of this evaluation are shown in Table 5.2.

[5] We will investigate the impact of other lexicons with presumably better scoring later in Section 5.6.2.

[6] Since none of the methods required training or involved any sophisticated hyper-parameters, we used both training and development data to optimize the threshold scores.

Method	Positive			Negative			Neutral			Macro	Micro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
HL	0.75	**0.76**	**0.76**	0.53	0.43	0.47	0.67	0.73	0.69	**0.615**	**0.685**
TBD	**0.77**	0.71	0.74	**0.54**	0.39	0.45	0.63	0.77	0.69	0.597	0.674
MST	0.75	0.72	0.74	0.48	**0.47**	**0.48**	**0.68**	0.72	0.7	0.606	0.675
JRK	0.6	0.31	0.41	0.42	0.2	0.27	0.43	0.8	0.56	0.339	0.467
KLCH	0.71	0.72	0.71	0.34	0.17	0.22	0.66	**0.82**	**0.73**	0.468	0.651
SB10k											
HL	**0.49**	**0.62**	**0.55**	0.27	0.33	0.3	**0.73**	0.62	0.67	**0.421**	0.577
TBD	0.48	0.6	0.53	0.24	0.27	0.25	0.72	0.63	0.67	0.393	0.57
MST	0.45	0.49	0.47	0.29	**0.35**	**0.32**	0.7	0.64	0.67	0.395	0.568
JRK	0.41	0.39	0.4	**0.36**	0.26	0.3	0.69	0.75	0.72	0.351	0.592
KLCH	0.39	0.22	0.28	0.34	0.13	0.19	0.66	**0.86**	**0.75**	0.235	**0.606**

Table 5.2: Results of lexicon-based MLSA methods
HL – Hu and Liu (2004), TBD – Taboada et al. (2011), MST – Musto et al. (2014), JRK – Jurek et al. (2015), KLCH – Kolchyna et al. (2015)

As we can see, the performance of the tested methods significantly varies across different polarity classes, but follows more or less the same pattern on both datasets: For example, the most simple approach of Hu and Liu (2004) achieves surprisingly good quality at predicting positive tweets, showing the highest recall and F_1-measure on the PotTS corpus and yielding the best overall scores for this polarity class on the SB10k set. Moreover, on the latter data, it also outperforms all other systems in terms of the precision of neutral microblogs. Combined with its generally good results on other metrics, this classifier attains the highest macro-averaged F_1-result for all classes and sets up a new benchmark for the micro-F_1 on the PotTS test set.

The approach of Taboada et al. (2011), which can be viewed as an extension of the previous method, only surpasses the HL classifier w.r.t. the precision of positive and negative messages, but still loses more than 0.02 macro-F_1 due to a lower recall of the neutral class. A better performance in this regard is shown by the analyzer of Musto et al. (2014), which shows a fairly strong recall of negative tweets, which in turn leads to the best F_1-score for this polarity. Unfortunately, since this semantic orientation is the most underrepresented one in both corpora, this success is not reflected in the overall statistics: Although this methods ranks second in terms of the macro-averaged F_1, it lags behind its competitors with regard to the micro-averaged value on the SB10k corpus.

Finally, the system of Kolchyna et al. (2015) shows very strong recall and F_1-scores for the neutral class on both sets and also achieves the best accuracy (0.606) on the SB10k data, but its quality for the remaining two polarities is fairly suboptimal, with the F_1-scores for these semantic orientations ranking last or second to last in both cases.

5.3.1 Polarity-Changing Factors

Since the analysis of context factors is commonly considered to be one of the most important components of any lexicon-based MLSA system, and because the method with the simplest approach to this task achieved surprisingly good results, outperforming other more sophisticated competitors, we decided to recheck the utility of this module for all classifiers. In order to do so, we successively deactivated, one by one, parts of the classifiers that analyzed the surrounding context of polar terms and recomputed the F_1-scores of all systems after these changes.

Polarity-Changing Factors	**System Scores**									
	HL		**TBD**		**MST**		**JRK**		**KLCH**	
	Macro $F_1{}^{+/-}$	Micro F_1	Macro $F_1{}^{+/-}$	Micro F_1	Macro $F_1{}^{+/-}$	Micro F_1	Macro $F_1{}^{+/-}$	Micro F_1	Macro $F_1{}^{+/-}$	Micro F_1
PotTS										
All	0.615	0.685	0.593	0.671	0.606	0.675	0.339	0.467	0.468	0.651
–Negation	0.622	**0.691**	0.596	0.672	**0.641**	0.7	0.357	0.473	0.298	0.463
–Intensification	NA	NA	0.595	0.672	NA	NA	0.339	0.467	NA	NA
–Other Modifiers	NA	NA	0.613	0.684	NA	NA	NA	NA	NA	NA
SB10k										
All	**0.421**	0.577	0.392	0.569	0.395	0.568	0.351	0.592	0.235	0.606
–Negation	0.415	0.576	0.395	0.572	0.381	0.559	0.316	0.586	0.218	**0.609**
–Intensification	NA	NA	0.4	0.576	NA	NA	0.352	0.59	NA	NA
–Other Modifiers	NA	NA	0.406	0.566	NA	NA	NA	NA	NA	NA

Table 5.3: Effect of polarity-changing factors on lexicon-based MLSA methods

As we can see from the results in Table 5.3, various methods respond in different ways to this ablation: For example, the scores of the Hu and Liu system improve on the PotTS corpus, but degrade on the SB10k dataset after switching off the negation handling. The same situation can also be observed with the analyzers of Musto et al. (2014) and Jurek et al. (2015). The classifier of Taboada et al. (2011), however, benefits from this deactivation in both cases, and the approach of Kolchyna et al. (2015) vice versa shows a performance drop on either dataset with the only exception being the micro-averaged F_1 on the SB10k data, which unexpectedly improves from 0.606 to 0.609.

As to the intensification handling, we can see that only two approaches (TBD and JRK) have this component at all. As in the previous case, the Taboada system profits from its deactivation, with the macro- and micro-averaged F_1-scores going up by 0.002 on PotTS and 0.008 on the SB10k corpus. A more varied situation is observed with the analyzer of Jurek et al. (2015), whose PotTS results are virtually unaffected by these changes, but the macro-averaged F_1 slightly increases and the micro-averaged score slightly decreases on the Cieliebak et al.'s dataset.

Finally, "other modifiers" (such as irrealis and interrogative clauses) only play a role as a polarity-changing factor in the system of Taboada et al. (2011) and, as we can see from the figures, do there rather more harm than good: deactivating this part boosts the macro-averaged F_1-scores on PotTS and SB10k by 0.02 and 0.014 respectively. At the same time, the micro-averaged result of this system climbs up from 0.671 to 0.684 on the former dataset, but drops from 0.569 to 0.566 on the latter corpus.

5.3.2 Error Analysis

In order to get a better intuition about the strengths and weaknesses of each particular classifier, we additionally collected a set of errors that were specific to only one of above the systems and will discuss some of these cases here in detail.

The first such error, which was made by the system of Taboada et al. (2011), is shown in Example 5.3.1. Here, a strongly positive tweet describing one's excitement about a technical report was erroneously classified as neutral despite the presence of the prototypical positive term "gut" (*good*) in its superlative form "beste" (*best*). Unfortunately, it is the degree of comparison which becomes fatal in this case: According to the implementation of Taboada et al. (2011), any superlative adjective has to be preceded by the definite article and a verb in order to be considered as a polar term for the final SO computation. Although the adjective "beste" (*best*) can fulfill the first criterion (it immediately follows the determiner "der" [*the*]), the lack of the preceding verb nullifies its effect.

Example 5.3.1 (An Error Made by the System of Taboada et al.)
Tweet: Der beste Microsoft Knowledgebase-Artikel, den ich je gelesen habe.
The best Microsoft-Knowledgebase article I've ever read.

Gold Label: positive
Predicted Label: neutral*

Another error of this method is shown in Example 5.3.2. This time, the presence of the colloquial term "verarschen" (*to hoax*) suggests that the tweet at hand is negative. Alas, the occurrence of another verb ("wollt" [*wanna*]) is interpreted as an irrealis clue, which prevents further SO computation and leads to a zero score to the whole message.[7]

Example 5.3.2 (An Error Made by the System of Taboada et al.)
Tweet: Die Konklave wählt den Papst und dann sagen sie Gott war es — Wollt ihr mich verarschen ?!

[7]Please note that the occurrence of the question mark does not affect the sentiment score because "?!" is not included in the list of valid punctuation marks in the original implementation.

The conclave elects the Pope and then they say it was God — do you wanna hoax me ?!

Gold Label: **negative**
Predicted Label: **neutral***

At this point, we already can see that the main flaws of the TBD approach apparently stem from its overly coarse rules, which, in addition, are not always valid in German, whose word order is significantly laxer than English syntax.[8]

Returning back to our error analysis, let us look at another erroneous case shown in Example 5.3.3. This time, the system of Musto et al. (2014) incorrectly assigned the neutral label to a positive tweet even though the positive term "gut" (*good*) again appears in this message. As it turns out, the occurrence of this word is still insufficient for the classifier to predict the positive class although this term has the highest possible positive score in the lexicon (1.0), which is additionally boosted by a factor of 1.5, since this word is an adjective. But the crushing factor in this case is the length of the tweet: since this approach relies on the average SO-score for all words in a sentence, the value 1.5 of the only positive term is divided by 7 (the length of the sentence) and drops down to 0.214, which is below the threshold for the positive class (0.267).

Example 5.3.3 (An Error Made by the System of Musto et al.)
Tweet: Mensch Meier, Mensch Meier! Das sieht gut aus für die %User:
Gosh Meier, Gosh Meier! It looks good for the %User:

Gold Label: **positive**
Predicted Label: **neutral***

As it turns out, this kind of mistakes is by far the most common type of errors characteristic to the MST system. Further examples of such incorrect decisions are provided in Example 5.3.4:

Example 5.3.4 (Errors Made by the System of Musto et al.)
Tweet: Der %User tut echt geile musik machen. Nichts mit Boyband hier.
The %User is making really great music. Nothing with Boyband here.

Gold Label: **positive**
Predicted Label: **neutral***

Tweet: Diese S5E5 Episode mit den Zugüberfall war wieder genial! BreakingBad

[8] In order to check this claim, we tried to temporarily deactivate the above two heuristics (predicate check for superlative adjectives and irrealis blocking by model verbs) and recomputed the scores of this system, getting in both cases an improvement by almost one percent on either corpus.

This S5E5 episode with train robbery was brilliant again! BreakingBad

Gold Label: positive
Predicted Label: neutral*

A different kind of problems is experienced by the approach of Jurek et al. (2015), which apparently has difficulties with correctly predicting the positive class. A deeper analysis of its misclassifications revealed that the reason for it is relatively simple: Because this classifier uses conditional probabilities of polar terms instead of their original lexicon scores, and we have estimated these probabilities on the noisily labeled German Twitter Snapshot, which was extremely biased towards the positive class (see Table 5.1), all positive lexicon entries received extremely high scores. As a consequence, even a single occurrence of a positive term in a message outweighed the effect of any negative expressions, even if they were more frequent in that tweet. This is, for instance, the case in Example 5.3.5 where the score of the (questionable) positive expression "Normal" (*normally*) is greater than the absolute sum of two negative values for the terms "sich beschweren" (*to complain*) and "ekelhaft" (*disgusting*).

Example 5.3.5 (An Error Made by the System of Jurek et al.)
Tweet: Normal bin ich ja nicht der mensch dwer sich beschwert wegen dem essen aber diese Pizza von Joeys... boah wie ekelhaft
Normally I'm not a person who complains about food but this pizza from Joeys... Boah it's so disgusting

Gold Label: negative
Predicted Label: positive*

The same problem also afflicts the system of Kolchyna et al., whose error example is given in 5.3.6. In contrast to the previous approaches, which mainly rely on manually designed heuristic rules, this method makes its decisions using a trained k-NN classifier. Nevertheless, its prediction in the provided case is still incorrect as it evidently confuses the positive class with the neutral polarity.

Example 5.3.6 (An Error Made by the System of Kolchyna et al.)
Tweet: das Hört sich echt Super an! %PosSmiley macht sami nicht auch so ein Video? Noah süsse beste Freunde! ♡ %User isilie saminator
It sounds really fantastic! %PosSmiley won't sami also make such a video? Noah's sweet best friends! ♡ %User isilie saminator

Gold Label: positive
Predicted Label: neutral*

In order to understand the reason for this misclassification, we first looked at the initial SO scores computed by the Kolchyna analyzer. As it turned out, both values that were used by the internal k-NN predictor of this system as features (the average SO score of all polar terms found in the message and the logarithm of this average) were relatively high, amounting to 33.42 and 2.52 respectively. But a closer look at the selected nearest neighbors revealed that even despite such high SO values, top three of the closest neighbors of this microblog were indeed neutral, as we can see from the list below:

1. **Tweet:** "Not in my backyard" -Mentalität dt. Politik: "Nächster Castor geht wohl doch nach Gorleben... -%Link antiatom"
   "Not in my backyard" -Mentality of German politics: "Next Castor will probably still got to Gorleben... -%Link antiatom"
   Label: neutral
   Distance: $6.83e^{-03}$;

2. **Tweet:** Kanzlerin im Google-Hangout: "Die Technik soll sich mal bemühen"
   Chancellor in Google-Hangout: "The technology should make an effort"
   Label: neutral
   Distance: $1.6e^{-02}$;

3. **Tweet:** Kanzlerin im Google-Hangout: "Die Technik soll sich mal bemühen"
   Chancellor in Google-Hangout: "The technology should make an effort"
   Label: neutral
   Distance: $1.6e^{-02}$;[9]

4. **Tweet:** Wünsche mir ein Format wie zdflogin auch für das %User. Viele Themen, klare Aussagen. Schönes Special %User zur Landtagswahl! %PosSmiley
   Wish %User had a format like zdflogin. Many topics, clear statements. Nice Special %User zur Landtagswahl! %PosSmiley
   Label: positive
   Distance: $2.1e^{-02}$;

5. **Tweet:** Ich bin ja so gespannt ob die FDP im September erst den Zahnärzten und dann den Apothekern mit Geschenken dankt, oder anders rum...
   I'm so curious whether FDP will first give gifts to dentists and then to pharmacists in September, or whether it'll be vice versa
   Label: positive
   Distance: $4.12e^{-02}$;

[9] Please note that this tweet is not a duplicate of the previous microblog, but a different message (with its distinct message id), which, however, has the same wording.

Even more surprisingly, the SO scores of the neighboring neutral instances were indeed also relatively high: In the first microblog, for example, the system recognized two polar terms: the English word "Not", which was confused with the German term "Not" (*distress*), and "nächster" (*next*). Another polar expression ("sich bemühen" [*to make an effort*]) was found in messages 2 and 3. Although two of these terms ("Not" and "sich bemühen") had a negative label in the sentiment lexicon, their conditional probability of being associated with the positive class was more than ten times bigger than the chance to appear in a negative microblog (according to the computed statistics). As a consequence of this positive probability bias, many neutral tweets from the training set ended up in close vicinity to actual positive examples.

As we can see, lexicon-based methods experience various kinds of problems with predicting the polarity of short casually written microblogs: Some of these systems apply rules that are too specific to a particular language and domain, so that they do not generalize well to German tweets; others rely on noisy statistics, which might be extraordinarily skewed towards just one polarity. Now, we should check whether other approaches to the message-level sentiment analysis (which rely on completely different principles and paradigms) will also be susceptible to these kinds of errors.

5.4 Machine-Learning Methods

Despite their immense popularity, linguistic plausibility, and simplicity to implement, lexicon-based approaches often have been criticized for the rigidness of their classification[10] and the inability to incorporate additional, non-lexical attributes into their final decisions. Moreover, as noted by Pang et al. (2002) and also confirmed empirically by Riloff et al. (2003) and Gamon (2004), many linguistic expressions that actually correlate with the subjectivity and polarity of a sentence (*e.g.*, exclamation marks or spelling variations) are very unlikely to be included into a sentiment lexicon even by a human expert. As a consequence of this, with the emergence of manually annotated corpora, lexicon-based systems have been gradually superseded by supervised machine-learning techniques.

One of the first steps in this direction was taken by Wiebe et al. (1999), who used a Naïve Bayes classifier to differentiate between subjective and objective statements. Using binary features that reflected the presence of a pronoun, an adjective, a cardinal number, or a modal verb in the analyzed sentence, the authors achieved an accuracy of 72.17% on the two-class prediction task (differentiating between positive and negative classes), outperforming the

[10]Since these systems only rely on the precomputed weights of lexicon entries, considering these coefficients as constant, their decision boundaries frequently appear to be suboptimal as many terms might have different polarity and intensity values depending on the domain (see Eisenstein, 2017; Yang and Eisenstein, 2017).

majority class baseline by more than 20%. An even better result (81.5%) could be reached when the dataset was restricted only to the examples with the most confident annotation.

Inspired by this success, Yu and Hatzivassiloglou (2003) presented a more elaborated system in which they first distinguished between subjective and objective documents, then differentiated between polar and neutral sentences, and, finally, determined the polarity of that clauses. As in the previous case, the authors used a Naïve Bayes predictor for the document-level task, reaching a remarkable F_1-score of 0.96 on this objective; and applied an ensemble of NB systems to predict the subjectivity of single sentences. To determine the semantic orientation of subjective clauses, Yu and Hatzivassiloglou averaged the polarity scores of their tokens, obtaining these scores from an automatically constructed sentiment lexicon (Hatzivassiloglou and McKeown, 1997). This way, they attained an accuracy of 91% on a set of 38 sentences that had a perfect inter-annotator agreement.

In order to check the effectiveness of the Naïve Bayes approach, Pang et al. (2002) compared the results of NB, MaxEnt, and SVM systems on the movie review classification task, trying to predict whether a review was perceived as thumbs up or thumbs down. In contrast to the previous works, they found the SVM classifier working best for this objective, yielding 82.9% accuracy when used with unigram features only. This conclusion paved the way for the following triumph of the support-vector approach, which was dominating the whole sentiment research field for almost a decade ever since. For example, Gamon (2004) also trained an SVM predictor using a set of linguistic and surface-level features (including part-of-speech trigrams, context-free phrase-structure patterns, and part-of-speech information coupled with syntactic relations) to distinguish between positive and negative customer feedback, achieving 77.5% accuracy and $\approx$0.77 F_1 by using only top 2,000 attributes that had the highest log-likelihood ratio with the target class. Furthermore, Pang and Lee (2005) addressed the problem of multi-class rating, attempting to predict the number of stars assigned to a review. For this purpose, they compared three different SVM types: (i) one-versus-all SVM (OVA-SVM), (ii) SVM regression, (iii) and OVA-SVM with metric labeling; getting their best results ($\approx$52% accuracy) with the last option. Finally, Ng et al. (2006) proposed a multi-stage SVM system, in which they first classified whether the given text was a review or not and then tried to predict its polarity. Due to a better usage of higher-order n-grams (where, instead of naïvely considering all token sequences up to length n as new features, the authors only took 5,000 most useful ones), Ng et al. (2006) even improved the state of the art on the Pang and Lee's corpus, boosting the classification accuracy from 87.1 to 90.5%.

But a real game change in the MLSA research field happened with the introduction of the SemEval shared task on sentiment analysis in Twitter (Nakov et al., 2013). Starting from its inaugural run in 2013, this competition has rapidly caught the attention of the broader NLP community and has been rerun five times, attracting more than 40 active participants every year.

It is not surprising that the first winning systems in this task closely followed in the footsteps of the advances in the general opinion mining at that time. For example, the two top-scoring submissions in the initial iteration (Mohammad et al., 2013; Günther and Furrer, 2013) both relied on the SVM algorithm: The first of these approaches, an analyzer developed by Mohammad et al. (2013), was the absolute winner of SemEval 2013, scoring impressive 0.69 macro-averaged two-class F_1 on the provided Twitter corpus. The key to the success of this method was an extensive set of linguistic features devised by the authors, which included character and token n-grams, Brown clusters (Brown et al., 1992), statistics on part-of-speech tags, punctuation marks, elongated words etc. But the most useful type of attributes according to the feature ablation test turned out to be the features that reflected information from various sentiment lexicons. In particular, depending on the type of the polarity list from which such information was extracted, Mohammad et al. introduced two types of lexicon attributes: *manual* and *automatic* ones. The former group was computed with the help of the NRC emotion lexicon (Mohammad and Turney, 2010), MPQA polarity list (Wilson et al., 2005), and Bing Liu's manually compiled polarity set (Hu and Liu, 2004). For each of these resources and for each of the non-neutral polarity classes (positive and negative), the authors estimated the total sum of the lexicon scores for all message tokens and also separately calculated these statistics for each particular part-of-speech tag, considering them as additional attributes. Automatic features were obtained using the Sentiment140 and Hashtag Sentiment Base polarity lists (Kiritchenko et al., 2014). Again, for each of these lexicons, for each of the two polarity classes, the authors produced four features representing the number of tokens with non-zero scores, the sum and the maximum of all respective lexicon values for all words, and the score of the last term in the tweet. These two feature groups (manual and automatic lexicon attributes) improved the macro-averaged ${F_1}^{+/-}$-score by almost five percent, outperforming in this regard all other traits.

Another notable submission, the system of Günther and Furrer (2013), also relied on a linear SVM predictor with a rich set of features. Like Mohammad et al. (2013), the authors used original and lemmatized unigrams, word clusters, and lexicon features. But in contrast to the previous approach, this application utilized only one polarity list—that of Esuli and Sebastiani (2005). Partially due to this fact, Günther and Furrer found the word clusters working best among all features. This method also yielded competitive results (0.653 ${F_1}^{+/-}$) on the message-level polarity task, attaining second place in that year.

Later on, Günther et al. (2014) further improved their results (from 0.653 to 0.691 two-class F_1) by extending the original system with a Twitter-aware tokenizer (Owoputi et al., 2013), spelling normalization module, and a significantly increased set of lexicon-based features. In particular, instead of simply relying on SentiWordNet (Esuli and Sebastiani, 2005), Günther et al. applied a whole ensemble of various polarity lists including Liu's opinion lexicon, MPQA subjectivity list, and TwittrAttr polarity resource. As mentioned by the

authors, the last change was of particular use to the classification accuracy, improving the macro-$F_1^{+/-}$ by almost four percent.

An even better score on this task, could be attained with the approach of Miura et al. (2014), who also utilized a supervised ML classifier with character and word n-grams, word clusters, disambiguated senses, and lexicon scores of message tokens as features. Similarly to the systems of Mohammad et al. (2013) and Günther et al. (2014), the authors made heavy use of various kinds of polarity lists including AFINN-111 (Nielsen, 2011), Liu's Opinion Lexicon (Hu and Liu, 2004), General Inquirer (Stone et al., 1966), MPQA Polarity List (Wiebe and Riloff, 2005), NRC Hashtag and Sentiment140 Lexicon (Mohammad et al., 2013), as well as SentiWordNet (Esuli and Sebastiani, 2006b), additionally applying a whole set of preprocessing steps such as spelling correction, part-of-speech tagging with lemmatization, and a special weighting scheme for underrepresented classes. Due to these enhancements, combined with a carefully tuned LogLinear classifier, Miura et al. (2014) were able to boost the sentiment classification results on the SemEval 2014 test set to 0.71 $F_1^{+/-}$.

In order to see how this family of methods would perform on our Twitter corpora, we have reimplemented the approaches of Gamon (2004), Mohammad et al. (2013), and Günther et al. (2014) with the following modifications: In the system of Gamon (2004), we used the available dependency analyses from the `MateParser` (Bohnet, 2009) instead of constituency trees, considering each node of the dependency tree as a syntactic constituent and regarding the two-tuple (`dependency-link-to-the-parent`, `node's-PoS-tag`) as the name of that constituent (for example, a finite verb at the root of the tree was mapped to the constituent (`-`, `VVFIN`), where `-` is the name of the root relation). Furthermore, because the Brown clusters were not available for German, we had to remove this attribute altogether from the feature sets of Mohammad et al.'s and Günther et al.'s methods. Moreover, because the former system relied on two types of lexicon attributes—manual and automatic ones, we used two polarity lists for these approaches: the Zurich Sentiment Lexicon of Clematide and Klenner (2010) as a manual resource and our Linear Projection Lexicon, which was introduced in Chapter 3, as an automatically generated polarity list. All remaining attributes and training specifics were kept maximally close to their original descriptions.

The results of our reimplementations are shown in Table 5.4. As we can see from the scores, the system of Mohammad et al. (2013) clearly dominates its competitors on both corpora. This holds for all presented metrics except for the recall of positive tweets on the PotTS dataset and neutral messages on the SB10k data, where it is outperformed by the analyzers of Günther et al. (2014) and Gamon (2004) respectively. In any other respect, however, the results of the MHM classifier are notably higher than those of the GNT method, sometimes surpassing it by up to 12% (this is, for instance, the case for the recall of negative microblogs on the SB10k corpus). This margin becomes even larger if we compare the scores of Mohammad's system with the performance of Gamon's predictor, which is by far the

Method	Positive			Negative			Neutral			Macro	Micro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
GMN	0.67	0.73	0.7	0.35	0.15	0.21	0.6	0.72	0.66	0.453	0.617
MHM	**0.79**	0.77	**0.78**	**0.58**	**0.56**	**0.57**	**0.73**	**0.76**	**0.74**	**0.674**	**0.727**
GNT	0.71	**0.8**	0.75	0.55	0.45	0.5	0.68	0.63	0.65	0.624	0.673
SB10k											
GMN	0.65	0.45	0.53	0.38	0.08	0.13	0.72	**0.93**	0.81	0.329	0.699
MHM	**0.71**	**0.65**	**0.68**	**0.51**	**0.4**	**0.45**	**0.8**	0.87	**0.84**	**0.564**	**0.752**
GNT	0.67	0.62	0.64	0.44	0.28	0.34	0.78	0.87	0.82	0.491	0.724

Table 5.4: Results of machine-learning–based MLSA methods
GMN – Gamon (2004), MHM – Mohammad et al. (2013), GNT – Günther et al. (2014)

weakest ML method in this survey. This weakness, however, is less surprising regarding the fact that Gamon's approach is purely grammar-based and relies only on information about part-of-speech tags and constituency parses without any lexicon traits or even plain n-gram features. Partially due to these limited input attributes, the results of this analyzer are even worse than the average scores of lexicon-based methods.

5.4.1 Feature Analysis

Because input features appeared to play a crucial role for the success of ML-based systems, we decided to investigate the impact of this factor in more detail and performed an ablation test for each of the tested classifiers, removing one of their feature groups at a time and recomputing their scores.

As we can see from the results in Table 5.5, the approach of Gamon (2004) typically achieves its best performance when all of the input attributes (PoS tags and syntactic constituents) are active. This is for example the case for the micro- and macro-averaged F_1 on the PotTS corpus, and also holds for the two-class macro-F_1 on the SB10k data. The only exception to this tendency is the micro-averaged F_1-score on the latter dataset, which shows a slight improvement (from 0.699 to 0.7) after the removal of part-of-speech features.

Similarly, the analyzer of Mohammad et al. (2013) seems to rather suffer than benefit from the part-of-speech attributes, which decrease its micro-averaged scores by almost 0.07 points on PotTS and 0.05 F_1 on SB10k. One possible explanation for this degradation could be the differences in the utilized PoS taggers and tagsets: Whereas the original Mohammad et al.'s classifier relied on a special Twitter-aware tagger (Owoputi et al., 2013), whose tags were explicitly adjusted to the peculiarities of social media texts (including special labels for the @-mentions and #hashtags), we instead used the output of the standard TreeTagger (Schmid, 1995), which, apart from lacking any Twitter-specific information,

Features	System Scores					
	GMN		MHM		GNT	
	Macro $F_1^{+/-}$	Micro F_1	Macro $F_1^{+/-}$	Micro F_1	Macro $F_1^{+/-}$	Micro F_1
PotTS						
All	**0.453**	**0.617**	**0.674**	0.727	**0.624**	0.673
–Constituents	0.388	0.545	NA	NA	NA	NA
–PoS Tags	0.417	0.607	0.669	0.721	NA	NA
–Character Features	NA	NA	0.671	**0.734**	NA	NA
–Token Features	NA	NA	0.659	0.704	0.0	0.366
–Automatic Lexicons	NA	NA	0.667	0.717	0.613	0.666
–Manual Lexicons	NA	NA	0.665	0.715	0.617	**0.675**
SB10k						
All	**0.329**	0.699	0.564	0.752	0.491	0.724
–Constituents	0.127	0.646	NA	NA	NA	NA
–PoS Tags	0.301	**0.7**	**0.57**	**0.757**	NA	NA
–Character Features	NA	NA	0.546	0.753	NA	NA
–Token Features	NA	NA	0.559	0.741	0.046	0.62
–Automatic Lexicons	NA	NA	0.54	0.753	**0.517**	0.735
–Manual Lexicons	NA	NA	0.553	0.751	0.51	**0.739**

Table 5.5: Results of the feature-ablation test for ML-based MLSA methods

was also trained on a completely different text genre (newspaper articles) and therefore a priori produced unreliable output. As a consequence, the effect of part-of-speech information is rather harmful, and the only aspect where it comes in handy is the macro-averaged F_1 on the PotTS corpus, which improves by 0.003 when these features are used. A better alternative in this regard could be the Twitter-specific tagger for German developed by Rehbein (2013), we could not, however, find this tagger in the public domain, and, moreover, its usage would preclude the following MATE analysis due to the difference in the tagsets.

An even more controversial situation is observed with the classifier of Günther et al. (2014). Although this system lacks any part-of-speech attributes, its reaction to the deletion of other features (first of all token and lexicon traits) is quite unexpected. For example, the macro-averaged F_1-scores on both corpora drop almost to zero when the information about tokens is excluded. On the other hand, the deactivation of manual lexicons surprisingly improves the micro-averaged results on both datasets and also increases the macro-$F_1^{+/-}$ on the SB10k data. We also notice a similar (though less pronounced) trend with automatic lexicons: the ablation of these features lowers the scores on PotTS, but improves both results on SB10k. We can partially explain this negative effect of polarity lists by the coarseness of lexicon features: This classifier uses only binary attributes, which reflect whether the given tweet has more positive or more negative lexicon items, but it does not distinguish between the scores or intensities of these terms.

Besides analyzing the utility of each particular feature group, we also decided to have

a look at the top-10 most relevant attributes learned by each system. The summarized overview in Table 5.6 partially confirms our previous findings: For example, the most useful traits for the analyzer of Gamon (2004) are attributes reflecting the information about both constituents and part-of-speech tags, with five of its ten entries featuring the interjection tag, which appears to be especially important for predicting the positive class. On the other hand, the system of Mohammad et al. (2013) seems to rely more on token and character n-grams, as nine out of ten attributes belong to either of these two categories. The only outlier in this respect is the `Last%QMarkCnt` attribute (line 2), which denotes the presence of a question mark and is apparently a good clue of neutral microblogs. Finally, the classifier of Günther et al. (2014) almost exclusively prefers lexical n-grams, as it has nine unigrams and one bigram among its top-ten entries.

Rank	**GMN**			**MHM**			**GNT**		
	Feature	Label	Weight	Feature	Label	Weight	Feature	Label	Weight
1	NK-ITJ|	POS	0.457	*	NEUT	0.131	hate	NEG	1.86
2	DM-ITJ|	POS	0.334	Last-%QMark-Cnt	NEUT	0.088	sick	NEG	1.7
3	V-DM-I	POS	0.244	s-c	NEG	0.079	kahretsinn	NEG	1.69
4	N-NK-I	POS	0.24	*-%possmiley	POS	0.067	dasisaberschade	NEG	1.69
5	MO-ITJ|	POS	0.211	c-h-e-i-s	NEG	0.064	Anziehen	POS	1.67
6	A-DM-I	POS	0.196	h-a-h	POS	0.064	\x016434	POS	1.65
7	A-MO-I	POS	0.191	t-‿-.	NEG	0.064	pärchenabend	POS	1.65
8	NK-ITJ	POS	0.165	geil	POS	0.062	derien♡♡	POS	1.65
9	NK-$.	NEUT	0.16	*-?	NEUT	0.062	schön-nicht	POS	1.56
10	DM-ITJ	POS	0.157	?	NEUT	0.061	applause	POS	1.5

Table 5.6: Top-10 features learned by ML-based MLSA methods (sorted by the absolute values of their weights)

5.4.2 Classifiers

Another important factor that could significantly affect the quality of ML-based approaches was the underlying classification method, which was used to optimize the feature weights and make the final predictions. Although most of the previous studies agree on the superior performance of support vector machines for this task (see Pang et al., 2002; Gamon, 2004; Mohammad et al., 2013), we decided to question these conclusions as well and reran our experiments, replacing the linear SVC predictor with the Naïve Bayes and Logistic Regression algorithms.

Somewhat surprisingly, these changes indeed resulted in an improvement, especially in the case of the logistic classifier, which yielded the best macro- and micro-averaged scores

for the systems of Mohammad et al. (2013) and Günther et al. (2014) on the PotTS corpus (see Table 5.7) and also produced the highest micro-F_1 results for these two approaches on the SB10k dataset. Nevertheless, the SVM algorithm still remains a competitive option, in particular for the feature-sparse method of Gamon (2004), but also with respect to the macro-F_1 of Mohammmad's and Günther's analyzers.

Classifier	**System Scores**					
	GMN		**MHM**		**GNT**	
	Macro $F_1^{+/-}$	Micro F_1	Macro $F_1^{+/-}$	Micro F_1	Macro $F_1^{+/-}$	Micro F_1
PotTS						
SVM	**0.453**	**0.617**	0.674	0.727	**0.624**	0.673
Naïve Bayes	0.432	0.577	0.635	0.675	0.567	0.59
Logistic Regression	0.431	0.612	**0.677**	**0.741**	**0.624**	**0.688**
SB10k						
SVM	0.329	**0.699**	**0.564**	0.752	0.491	0.724
Naïve Bayes	**0.351**	0.637	0.516	0.755	0.453	0.675
Logistic Regression	0.309	0.693	0.553	**0.772**	**0.512**	**0.75**

Table 5.7: Results of ML-based MLSA methods with different classifiers

Even though our results contradict previous claims in the literature, we would advise against premature conclusions at this point and stress the fact that different classifiers might have fairly varying results on different datasets. Therefore, higher scores of the logistic regression on our corpora do not preclude better SVM results on the official SemEval data.

5.4.3 Error Analysis

As in our previous experiments, we also decided to have a closer look at errors produced by each tested system. For this purpose, we again collected misclassifications that were unique to only one of the classifiers, and provide some examples of these errors below.

The first wrong result shown in Example 5.4.1 was produced by the system of Gamon (2004).

Example 5.4.1 (An Error Made by the System of Gamon)
Tweet: Das ist das zynische. Über Themen labern, Leute schlecht machen. Wenn nicht der Papst damit Thema wäre, kein Wort. Ich hasse das.
It's cynical. To babble about topics, to talk people down. If the topic wouldn't be the Pope, no word. I hate this.

Gold Label: negative
Predicted Label: positive*

In this case, the classifier incorrectly assigned the positive label to a clearly negative microblog despite the presence of multiple negatively connoted terms ("zynische" [*cynical*], "labern" [*to babble*], "schlecht machen" [*to talk down*], and "hasse" [*to hate*]). The reason for this decision is quite simple: As we already noted in the foregoing description, this method is completely unlexicalized and relies only on grammatical information while making its predictions. In particular, for this microblog, the top-5 most important features (ranked by the absolute values of their coefficients) are:

1. `PD-ADJA` (neutral): -0.62896911412,
2. `--VVINF` (negative): 0.517300341184,
3. `PD-ADJA` (positive): 0.505413668274,
4. `--VVINF` (positive): -0.346990702756,
5. `CJ-VVINF` (positive): 0.303311030403.

As we can see, none of these attributes reflects any information about the lexical terms appearing in the message, and the system simply prefers the positive class based on the presence of a predicate adjective (`PD-ADJA`) and coordinately conjoined infinitive (`CJ-VVINF`).

Another error shown in Example 5.4.2 was made by the system of Mohammad et al. (2013). This time, a positive tweet was misclassified as neutral. But the reason for this erroneous decision is completely different. As we can see from the list of the highest ranked features given below:

1. `*` (neutral): 0.131225868029,
2. `*` (negative): -0.0840804221845,
3. `%PoS-CARD` (neutral): 0.0833658576233,
4. `%PoS-ADJD` (neutral): -0.069745190018,
5. `t-␣-n` (positive): 0.0556721202587;

this analyzer makes its decision based on rather general, but extremely heavy-weighted features, such as placeholder token `*` or the PoS-tag features (`%PoS-CARD` and `%PoS-ADJD`). As a result, its prediction succumbs to the neutral bias of these general attributes.

Example 5.4.2 (An Error Made by the System of Mohammad et al.)
Tweet: das klingt richtig gut! Was für eine hast du denn? (uvu) %PosSmiley3
It sounds really great. Which one do you have? (uvu) %PosSmiley3

Gold Label: positive
Predicted Label: neutral*

Finally, the last example (5.4.3) shows another wrong decision, where a negative microblog was incorrectly analyzed as positive by the method of Günther et al. (2014), even though the polar term "borniert" (*narrow-minded*) was present in both utilized sentiment lexicons (ZPL and Linear Projection) as a negative item. This again can be explained by the prevalence of general features (*e.g.*, `8`, `nicht-nur_NEG`, `nur_NEG`, etc.) and their strong bias towards the majority class in the PotTS dataset.

Example 5.4.3 (An Error Made by the System of Günther et al.)
Tweet: Den CDU-Wählern traue ich durchaus zu der FDP 8 bis 9% zu bescheren! Die sind so borniert, nicht nur in Niedersachsen!
I don't put giving 8 to 9% to the FDP past the CDU-voters! They are so narrow-minded, not only in Lower Saxony!

Gold Label: negative
Predicted Label: positive

5.5 Deep-Learning Methods

Even though traditional ML-based approaches still show competitive results and play an important role in the sentiment analysis of social media, they are gradually giving place to allegedly more powerful and in a certain sense more intuitive deep learning (DL) methods. As we already mentioned in the previous chapter, in contrast to the standard supervised techniques with human-engineered features, DL systems induce the best feature representation completely automatically, and in some cases might produce even better features than the ones devised by human experts. Another important advantage of this paradigm is its more straightforward way to implement the "compositionality" of language (Frege, 1892): Whereas conventional classifiers usually consider each instance as a *bag of features* and predict its label based on the sum of these features' values multiplied with their respective weights, DL approaches try to *combine* the representation of each part of that instance (be it tokens or sentences) into a single whole and then deduce the final class from this joint embedding.

Among the first who explicitly incorporated the compositionality principle into a DL-based sentiment application were Yessenalina and Cardie (2011). In their proposed matrix-space approach, the authors represented each word w of an input phrase $\mathbf{x}^i = w^i_1, w^i_2, \ldots, w^i_{|\mathbf{x}^i|}$ as a matrix $W_w \in \mathbb{R}^{m \times m}$ and computed the sentiment score ξ^i of this phrase as the product

of its token matrices, multiplying the final result with two auxiliary model parameters $\vec{u}$ and $\vec{v} \in \mathbb{R}^m$ to get a scalar value:

$$\xi^i = \vec{u}^\top \left(\prod_{j=1}^{|x^i|} W_{w_j^i} \right) \vec{v}.$$

After computing this term, they predicted the intensity and polarity of the phrase on a five-level sentiment scale (ranging from very negative to very positive) by comparing ξ^i with automatically derived thresholds. With this system, Yessenalina and Cardie attained a ranking loss of 0.6375 on the MPQA corpus (Wiebe et al., 2005), outperforming the traditional PRank algorithm (Crammer and Singer, 2001) and bag-of-words ordered logistic regression.

Almost simultaneously with this work, Socher et al. (2011) introduced a deep recursive autoencoder (RAE), in which they obtained a fixed-width vector representation for a complex phrases $\vec{w}_p$ by recursively merging the vectors of its tokens over a binarized dependency tree, first multiplying these vectors with a compositional matrix W and then applying a non-linear function ($softmax$) to the resulting product:

$$\vec{w}_p = softmax \left(W \begin{bmatrix} \vec{w}_l \\ \vec{w}_r \end{bmatrix} \right), \tag{5.2}$$

where $\vec{w}_l$ and $\vec{w}_r$ represent the embeddings of the left and right dependents respectively. By applying a max-margin classifier to the final phrase vector, the authors could improve the state of the art on predicting sentence-level polarity of user's blog posts (Potts, 2010) and also outperformed the system of Nasukawa and Yi (2003) on the MPQA dataset (Wiebe et al., 2005), achieving 86.4% accuracy on predicting contextual polarity of opinionated expressions.

Later on, Socher et al. (2012) further improved this approach by associating an additional matrix W_w with each vocabulary word w and performing the inference simultaneously over both vector and matrix representations:

$$\vec{w}_p = \tanh \left(W_v \begin{bmatrix} W_r \vec{w}_l \\ W_l \vec{w}_r \end{bmatrix} \right),$$

$$W_p = W_m \begin{bmatrix} W_l; \\ W_r \end{bmatrix};$$

where $\vec{w}_p \in \mathbb{R}^n$ stands for the embedding of the parent node, $\vec{w}_l$ and $\vec{w}_r$ represent the embeddings of its left and right dependents, and $W_p, W_l, W_r \in \mathbb{R}^{n \times n}$ denote the respective matrices associated with these vertices. The compositionality matrices $W_v \in \mathbb{R}^{n \times 2n}$ and $W_m \in \mathbb{R}^{n \times 2n}$ were shared across all instances and learned along with the vector embeddings. This model, called Matrix-Vector Recursive Neural Network (MVRNN), surpassed the RAE

system on the IMDB movie review dataset (Pang and Lee, 2005), attaining 0.91 Kullback-Leibler divergence between the assigned scores and probabilities of correct labels.

Yet another improvement, a Recursive Neural Tensor Network (RNTN), was presented by Socher et al. (2013). In this system, the authors again opted for a vector representation of words, but enhanced the original matrix-vector product from Equation 5.2 with an additional tensor multiplication:

$$\vec{w}_p = softmax\left(\begin{bmatrix}\vec{w}_l\\ \vec{w}_r\end{bmatrix}^\top V^{[1:d]}\begin{bmatrix}\vec{w}_l\\ \vec{w}_r\end{bmatrix} + W\begin{bmatrix}\vec{w}_l\\ \vec{w}_r\end{bmatrix}\right),$$

where $\vec{w}_p, \vec{w}_l, \vec{w}_r \in \mathbb{R}^n$, and $W \in \mathbb{R}^{n\times 2n}$ are defined as before; and V represents a $2n \times 2n \times n$-dimensional tensor. By increasing this way the number of parameters in comparison with the RAE approach, but significantly reducing it with respect to the MVRNN method, the authors gained a significant improvement of the results, boosting the classification accuracy on their own Stanford Sentiment Treebank from 82.9 to 85.4%.

A real breakthrough in the use of deep learning methods for sentiment analysis of Twitter happened with the work Severyn and Moschitti (2015b), whose proposed feed-forward DL system ranked first in SemEval-2015 Subtask 10 A (phrase-level polarity prediction) (Rosenthal et al., 2015) and achieved second place (0.6459 ${F_1}^{+/-}$) in Subtask 10 B (message-level classification) of this competition. Drawing on the ideas of Kalchbrenner et al. (2014), the authors devised a simple convolutional network in which they multiplied pretrained word embeddings with 300 distinct convolutional kernels each of width 5, pooled the maximum value of this multiplication for each kernel, and then passed the results of this pooling to a piecewise linear ReLU filter with a densely connected softmax layer. An important aspect of this approach, which accounted for a huge part of its success, was a special multi-stage training scheme that was used to optimize the parameters: In the initial stage of this scheme, Severyn and Moschitti first computed Twitter-specific word embeddings by applying the word2vec algorithm to a large Twitter corpus. Afterwards, they pretrained the complete system including the word vectors, convolutional filters, and inter-layer matrices on a big set of noisily labeled microblogs from this collection, and, finally, fine-tuned the parameters of the model on the official SemEval dataset.

Later on, this system was further improved by Deriu et al. (2016), who increased the number of convolutional layers (applying two layers instead of one) and simultaneously trained two such models (using word2vec vectors as input for the first one and passing GloVe embeddings to the second), joining their output at the end and achieving this way 0.671 ${F_1}^{+/-}$ on the SemEval-2015 test set. A similar enhancement was also proposed by Rouvier and Favre (2016), who used three different types of embeddings (word2vec, word2vec specific to particular parts of speech, and sentiment-tailored vectors), training separate sets of convolutions for each of these types.

Although convolutional approaches still show competitive scores and are hard to outperform in practice, in recent time, they are gradually being superseded by recurrent neural networks (Xu et al., 2016; Wang et al., 2015b). One of the most prominent such systems has been recently proposed by Baziotis et al. (2017). In their submission to SemEval 2017 (Rosenthal et al., 2017), the authors used two successive bidirectional LSTM units (BiLSTMs). In each of these units, they concatenated the results of the left-to-right recurrence ($\vec{h}^{(l)}_{i_{\rightarrow}} \in \mathbb{R}^{150}$) with the respective outputs of the right-to-left loop ($\vec{h}^{(l)}_{i_{\leftarrow}} \in \mathbb{R}^{150}$) and then passed the result of this concatenation ($\vec{h}^{(l)}_{i} = [\vec{h}^{(l)}_{i_{\rightarrow}}, \vec{h}^{(l)}_{i_{\leftarrow}}] \in \mathbb{R}^{300}$) to the next layer of the network. After getting the output of the second BiLSTM, they united the states of this unit from all time steps i into a single vector $\vec{a}$ with the help of a special attention mechanism, in which they first multiplied each BiLSTM state $\vec{h}_i$ with the respective globally normalized attention score a_i and then took the sum of these weighted vectors over all i positions:

$$\begin{aligned} \vec{a} &= \sum_{i=1}^{|\mathbf{x}|} a_i \vec{h}^{(2)}_i, \\ \text{where } a_i &= \frac{\exp(e_i)}{\sum_{j=1}^{|\mathbf{x}|} \exp(e_j)}, \\ \text{s.t. } e_i &= \tanh\left(\vec{\alpha}\vec{h}^{(2)}_i + \beta_i\right). \end{aligned} \tag{5.3}$$

The $\vec{\alpha}$ and β terms in the above equations denote the attention parameters (score and bias), which are optimized during the training process. To make the final prediction, Baziotis et al. multiplied the attention vector $\vec{a}$ with matrix W and computed element-wise softmax of this product, getting probability scores for each of the three polarity classes and choosing the label with the maximum score:

$$\hat{y} = \text{argmax}\left(softmax(W^{\top}\vec{a})\right). \tag{5.4}$$

With this approach, the authors attained the first first place in Task 4 of SemEval-2017 (0.675 $F_1^{+/-}$), being on a par with the system of Cliche (2017) and even outperforming the method of Rouvier (2017) despite the fact that both of these competitors used ensembles of LSTMs and convolutional networks.

5.5.1 Lexicon-Based Attention

Even though the approach of Baziotis et al. (2017) represents the current state of the art in sentiment analysis of Twitter and yields extraordinarily good results, in our opinion, this method has yet some potential for improvements. This, first of all, concerns the way how attention coefficients are computed. As we can see from Equation 5.5, the magnitude of the attention score a_i primarily depends on the absolute value of the BiLSTM outputs and the bias term at the i-th position. Albeit this strategy is definitely plausible, assuming the fact

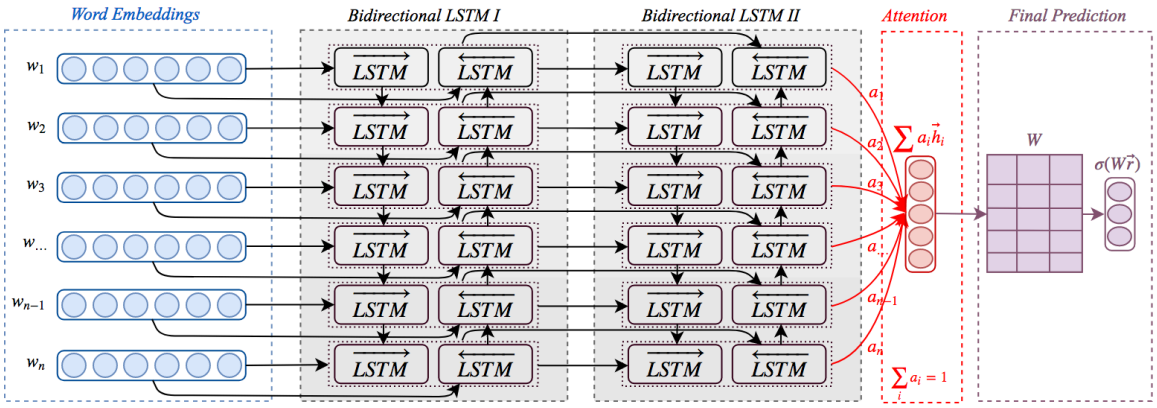


Figure 5.1: Architecture of the neural network proposed by Baziotis et al. (2017)

that LSTMs shall produce higher scores for polar tokens and presupposing that polar terms near the end of the message will usually have a greater influence on the net polarity of the tweet than subjective words at its beginning, a crucial prerequisite for this strategy to work is (i) that the LSTM layer can already provide sufficiently reliable results and (ii) that the bias terms do not overly boost the importance of irrelevant tokens that just accidentally appeared at favored positions. Unfortunately, both of these prerequisites are rarely fulfilled in practice.

In order to overcome these deficiencies, we augmented the original architecture of Baziotis et al. (2017) shown in Figure 5.1 with two additional types of attention: *lexicon-* and *context-based* one. In the former type, we estimated the importance weight b_i for position i as the polarity score of the word w_i, obtaining this value from our Linear Projection lexicon and normalizing it by the sum of polarity scores for all tweet tokens:

$$\vec{b} = \sum_{i=1}^{|\mathbf{x}|} b_i \vec{h}_i,$$

$$b_i = \frac{\exp(f_i)}{\sum_{j=1}^{|\mathbf{x}|} \exp(f_j)},$$

$$\text{s.t. } f_i = \begin{cases} \tanh(abs(V[w_i]) + \epsilon) & \text{if } w_i \in V \\ \tanh(\epsilon) & \text{otherwise.} \end{cases}$$

This way, we hoped to force the network to pay more attention to the BiLSTM outputs that were produced at the positions of polar terms rather than favoring arbitrary words in the message.

Another important factor that could notably affect the polarity of a microblog were the so-called *valence shifters* (Polanyi and Zaenen, 2006)—words and phrases such as "kaum" (*hardly*) or "nicht" (*not*) that could significantly change (or even reverse) the semantic orientation of polar terms. To account for these phenomena, we added another type of attention—a *context-based* one, whose goal was to identify such shifters in the message and give them

bigger weights in the recursion. To discern these elements, we introduced a linear classifier that had to predict the modifying power of a token w_i, given its original word embedding $\vec{w}_i$ and the LSTM output of its parent in the dependency tree times the lexicon-based attention score of that parent ($\vec{b}_p := b_p\vec{h}_p$). To keep the resulting attention scores within an appropriate range, we again used the same tanh transformation and global normalization over all positions as we did in the previous two types:

$$\begin{aligned}\vec{c} &= \sum_{i=1}^{|\mathbf{x}|} c_i \vec{h}_i, \\ c_i &= \frac{\exp(g_i)}{\sum_{j=1}^{|\mathbf{x}|}\exp(g_j)}, \\ g_i &= \tanh\left(C[\vec{w}_i, \vec{b}_p]^\top\right).\end{aligned}$$

The C term in the above equation represents a context-based attention matrix $\mathbb{R}^{200\times100}$; the $\vec{w}_i$ variable denotes the word embedding of the i-th token; and the $\vec{b}_p$ term stands for the value of vector $\vec{b}$ (the result of lexicon-based attention from Equation 5.5.1) at position p (the index of syntactic parent of w_i). With this classifier, we hoped to amplify the importance of shifting words in the cases when the immediate syntactic ancestors of these tokens were highly subjective expressions (*e.g.*, "Er hat die Prüfung kaum bestanden" [*He hardly passed the exam*] or "Ich mag den neuen Bundesminister nicht" [*I do not like the new federal minister*]), but ignore them when they did not relate to any subjective term.

At last, to make the final prediction, we concatenated the outputs of the three attention layers into a single matrix $A \in \mathbb{R}^{3\times100}$ and multiplied it with a vector $\vec{w} \in \mathbb{R}^{1\times100}$, applying softmax normalization at the end:

$$\begin{aligned}\vec{o} &= softmax\left(A\vec{w}^\top\right), \text{ where} \\ A &= \begin{bmatrix}\vec{a} \\ \vec{b} \\ \vec{c}\end{bmatrix}.\end{aligned}$$

Since introducing additional attention types increased the number of model parameters, we removed one of the intermediate Bi-LSTM layers in the network to counterbalance this effect and report our results for both settings: using one and two Bi-LSTM units (denoted as LBA$^{(1)}$ and LBA$^{(2)}$, respectively). The final architecture of our approach is shown in Figure 5.2.

To evaluate the performance of the previously presented methods and to compare our lexicon-based attention system with these solutions, we reimplemented the approaches of Yessenalina and Cardie (2011), Socher et al. (2011, 2012, 2013), Severyn and Moschitti (2015b), and Baziotis et al. (2017). For the sake of uniformity and simplicity, we used task-specific

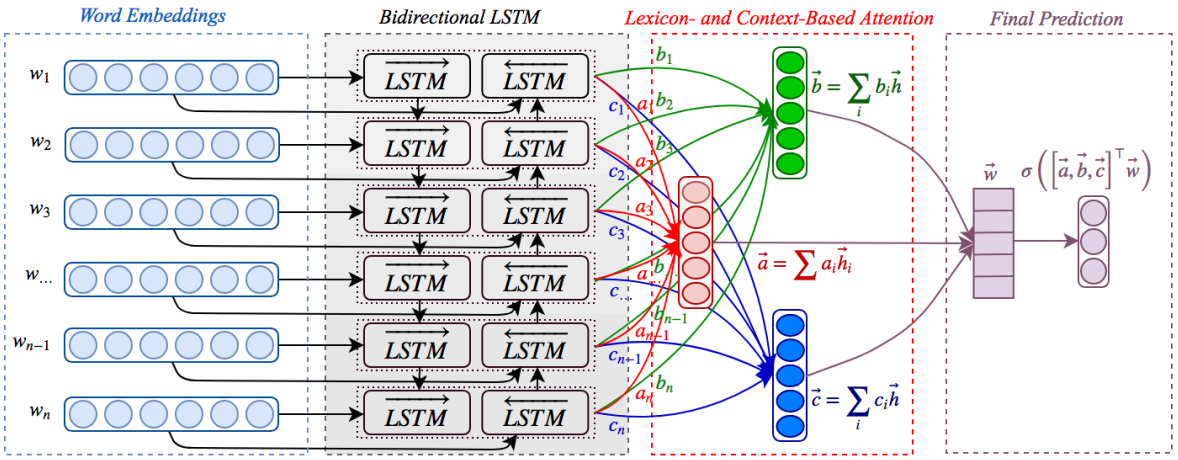


Figure 5.2: Architecture of the neural network with lexicon- and context-based attention

word embeddings of size $\mathbb{R}^{100}$ in all systems, optimizing these vectors along with other network parameters during the training. Moreover, we also unified the final activation parts and cost functions of all networks, using a densely connected softmax layer as the last component of each classifier and optimizing their weights w.r.t. the categorical hinge loss on the training data, picking the values that yielded the highest accuracy on the development set.

The results of this evaluation are shown in Table 5.8. As we can see from the figures, the LBA method performs fairly well, especially on the positive and neutral classes where it achieves the best F_1-benchmarks on both datasets and also attains the highest overall micro-averaged F_1-scores on all test samples (0.662 on PotTS and 0.737 on SB10k). Even though our approach also yields the best macro-averaged result on the SB10k set (0.321 F_1), it seems to face a major difficulty with the extreme label skewness of this corpus, failing to predict any negative tweet in the test set. This problem, in general, appears to be an insurmountable hurdle for almost all other compared systems, especially the matrix-space, MVRNN, and convolutional approaches, which eventually end up predicting only the most common neutral label for all messages in this dataset. A single notable exception to this tendency is the recursive neural tensor approach of Socher et al. (2013), which succeeds in classifying some of the negative instances and also predicts positive and neutral labels, but whose precision and recall are still far below an acceptable level.

A similar, though less severe situation is also observed on the PotTS corpus. This time, the Y&C, MVRNN, BAZ, and LBA$^{(2)}$ methods lapse into always predicting only the most frequent positive class. Other systems, however, perform much better, especially the approach of Severyn and Moschitti (2015b), which does an extraordinarily good job at classifying

Method	Positive			Negative			Neutral			Macro $F_1^{+/-}$	Micro F_1
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1		
PotTS											
Y&C	0.45	**1.0**	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.308	0.446
RAE	0.64	0.78	0.7	0.38	0.04	0.08	0.57	0.68	0.62	0.389	0.605
MVRNN	0.45	**1.0**	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.308	0.446
RNTN	0.45	0.87	0.59	0.19	0.02	0.03	0.32	0.1	0.15	0.312	0.428
SEV	0.73	0.79	0.76	**0.41**	**0.52**	**0.46**	**0.72**	0.55	0.62	**0.608**	0.651
BAZ	0.45	**1.0**	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.308	0.446
LBA$^{(1)}$	**0.82**	0.73	**0.77**	0.0	0.0	0.0	0.56	**0.92**	**0.69**	0.387	**0.662**
LBA$^{(2)}$	0.45	**1.0**	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.308	0.446
SB10k											
Y&C	0.0	0.0	0.0	0.0	0.0	0.0	0.62	**1.0**	0.77	0.0	0.622
RAE	0.63	0.57	0.6	0.0	0.0	0.0	**0.75**	0.94	0.83	0.299	0.721
MVRNN	0.0	0.0	0.0	0.0	0.0	0.0	0.62	**1.0**	0.77	0.0	0.622
RNTN	0.2	0.03	0.05	0.07	0.01	0.02	0.62	0.94	0.75	0.033	0.594
SEV	0.0	0.0	0.0	0.0	0.0	0.0	0.62	**1.0**	0.77	0.0	0.622
BAZ	0.75	0.47	0.58	0.0	0.0	0.0	0.71	0.98	0.83	0.291	0.72
LBA$^{(1)}$	0.72	**0.58**	**0.64**	0.0	0.0	0.0	0.74	0.97	**0.84**	**0.321**	**0.737**
LBA$^{(2)}$	**0.76**	0.49	0.6	0.0	0.0	0.0	0.72	0.98	0.83	0.298	0.723

Table 5.8: Results of deep-learning–based MLSA methods
Y&C – Yessenalina and Cardie (2011), RAE – Recursive Auto-Encoder (Socher et al., 2011), MVRNN – Matrix-Vector RNN (Socher et al., 2012), RNTN – Recursive Neural-Tensor Network (Socher et al., 2013), SEV – Severyn and Moschitti (2015b), BAZ – Baziotis et al. (2017), LBA$^{(1)}$ – lexicon-based attention with one Bi-LSTM layer, LBA$^{(2)}$ – lexicon-based attention with two Bi-LSTM layers

negative messages, reaching remarkable 0.46 F_1 on this subset and also attaining the best macro-average score (0.608) on all tweets due to its competitive performance on positive and neutral microblogs. Nevertheless, even the best-performing DL systems (SEV and LBA) lag far behind the traditional supervised machine-learning method of Mohammad et al. (2013), and barely outperform the lexicon-based approach of Hu and Liu (2004) in terms of the micro-averaged F_1 on SB10k. Two possible explanations for these mediocre scores could be a bad starting point of the parameters, which prevented the optimizers from finding the optimal solution to the optimization objective, or an insufficient amount of training data, which caused an extreme overfitting of the training set, but poor generalization to unseen examples. We will now investigate both of these factors in detail.

5.5.2 Word Embeddings

As in the previous chapters, we decided to replace randomly initialized word vectors in the very first layer of vector-based neural networks with pretrained word2vec embeddings, keeping this parameter fixed during the optimization. As we can see from the figures in

Table 5.9, this operation leads to a significant improvement of the results for almost all classifiers except for the recursive auto-encoder and convolutional approach of Severyn and Moschitti (2015b), where it slightly lowers the micro-averaged F_1-score in the former case (from 0.605 to 0.55) and considerably worsens the macro-averaged F_1 (from 0.608 to 0.36 F_1) of the latter system. Nonetheless, even despite these exceptional setbacks, the best observed macro-score increases from 0.608 to 0.64 on the PotTS dataset and almost doubles from 0.321 to 0.53 on the SB10k data. A similar situation is observed with the micro-averaged F_1, which rises from 0.662 to 0.69 on PotTS and also improves from 0.737 to 0.75 on the SB10k corpus. Unfortunately, these improvements usually come at the expense of a lower recall of the majority classes (positive and neutral respectively), but the gains in the overall metrics are generally much higher and, first of all, more important than the losses in these single aspects.

Method	**Positive**			**Negative**			**Neutral**			**Macro**	**Micro**
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
RAE	$0.58^{-0.06}$	$0.74^{-0.04}$	$0.65^{-0.05}$	$0.34^{-0.04}$	$0.26^{+0.22}$	$0.29^{+0.21}$	$0.59^{+0.02}$	$0.46^{-0.22}$	$0.52^{-0.1}$	$0.47^{+0.08}$	$0.55^{-0.06}$
RNTN	$0.48^{+0.03}$	$0.77^{-0.1}$	0.59	$0.33^{+0.14}$	$0.03^{+0.01}$	$0.06^{+0.03}$	$0.46^{+0.14}$	$0.33^{+0.23}$	$0.38^{+0.01}$	$0.33^{+0.02}$	$0.47^{+0.04}$
SEV	$0.69^{-0.04}$	$0.74^{-0.05}$	$0.72^{-0.04}$	$0.0^{-0.41}$	$0.0^{-0.52}$	$0.0^{-0.46}$	$0.58^{-0.14}$	$0.84^{+0.29}$	$0.69^{+0.07}$	$0.36^{-0.25}$	$0.64^{-0.01}$
BAZ	$0.85^{+0.4}$	$0.61^{-0.39}$	$0.71^{+0.09}$	$0.57^{+0.57}$	$0.32^{+0.32}$	$0.41^{+0.41}$	$0.55^{+0.55}$	$0.87^{+0.87}$	$0.68^{+0.68}$	$0.56^{+0.25}$	$0.65^{+0.2}$
LBA$^{(1)}$	$0.86^{+0.04}$	$0.6^{-0.13}$	$0.71^{-0.06}$	$0.61^{+0.61}$	$0.46^{+0.46}$	$0.53^{+0.53}$	$0.6^{+0.04}$	$0.89^{-0.03}$	$0.72^{+0.03}$	$0.62^{+0.23}$	$0.68^{+0.02}$
LBA$^{(2)}$	$0.84^{+0.39}$	$0.65^{-0.35}$	$0.73^{+0.11}$	$0.57^{+0.57}$	$0.54^{+0.54}$	$0.55^{+0.55}$	$0.63^{+0.63}$	$0.82^{+0.82}$	$0.72^{+0.72}$	$0.64^{+0.33}$	$0.69^{+0.24}$
SB10k											
RAE	$0.61^{-0.02}$	$0.56^{-0.01}$	$0.58^{-0.02}$	$0.29^{+0.29}$	$0.01^{+0.01}$	$0.02^{+0.02}$	$0.74^{-0.01}$	$0.92^{-0.02}$	$0.82^{-0.01}$	0.3	$0.71^{-0.01}$
RNTN	$0.54^{+0.34}$	$0.02^{-0.01}$	$0.04^{-0.01}$	$0.0^{-0.07}$	$0.0^{-0.01}$	$0.0^{-0.02}$	$0.63^{+0.01}$	$1.0^{+0.06}$	$0.77^{+0.02}$	$0.02^{-0.01}$	$0.62^{+0.03}$
SEV	$0.72^{+0.72}$	$0.5^{+0.5}$	$0.59^{+0.59}$	$0.49^{+0.49}$	$0.27^{+0.27}$	$0.35^{+0.35}$	$0.75^{-0.13}$	$0.92^{-0.08}$	$0.82^{+0.05}$	$0.47^{+0.47}$	$0.73^{+0.11}$
BAZ	$0.78^{+0.03}$	$0.51^{+0.04}$	$0.61^{+0.03}$	$0.49^{+0.49}$	$0.42^{+0.42}$	$0.45^{+0.45}$	$0.78^{+0.07}$	$0.91^{-0.07}$	$0.84^{+0.01}$	$0.53^{+0.24}$	$0.75^{+0.03}$
LBA$^{(1)}$	$0.84^{+0.12}$	$0.42^{-0.16}$	$0.56^{-0.08}$	$0.5^{+0.5}$	$0.28^{+0.28}$	$0.36^{+0.36}$	0.74	$0.96^{-0.01}$	0.84	$0.46^{+0.14}$	$0.73^{+0.01}$
LBA$^{(2)}$	$0.79^{+0.03}$	$0.45^{-0.04}$	$0.57^{-0.03}$	$0.57^{+0.57}$	$0.23^{+0.23}$	$0.33^{+0.33}$	$0.74^{+0.02}$	$0.96^{-0.02}$	$0.84^{+0.01}$	$0.45^{+0.15}$	$0.74^{+0.02}$

Table 5.9: Results of deep-learning–based MLSA methods with pretrained word2vec vectors

In order to see whether these changes would be different if we optimized word representations as well, we reran our experiments once again, initializing word vectors with word2vec embeddings as before, but allowing them to be updated during the training. Moreover, to approximate task-specific representations of words that were missing from the training set, we also computed the optimal transformation matrix for converting the original word2vec vectors into optimized sentiment embeddings using the method of the ordinary least squares, as we did in the previous chapters, and used this matrix to derive task-specific vectors during the testing.

As suggested by the results in Table 5.10, these modifications improve the results even further, setting a new record of the macro-averaged F_1-scores on the PotTS corpus (0.69 F_1), and pushing our LBA$^{(1)}$ system even above its most challenging competitors. A similar

effect is also observed with other systems, first of all BAZ and LBA$^{(2)}$, which yield similarly good results for all polarities. Nevertheless, like in the previous case, these improvements usually cause a drop in the recall for the most frequent class of the respective dataset, which is especially severe for the system of Baziotis et al. on the PotTS data (-0.28 on positive messages) and the LBA$^{(1)}$ approach on the SB10k test set (-0.17 on neutral tweets). Furthermore, the convolutional system of Severyn and Moschitti and recursive neural tensor approach of Socher et al. (2013) fail to predict any negative tweet on PotTS and SB10k, respectively, which also leads to a notable drop of their overall macro-F_1–values. These drops, however, are rather exceptional, as the same system of Severyn and Moschitti shows an extraordinary big boost of the results on the SB10k corpus (+0.45 macro-F_1 and +0.1 micro-F_1–score), and the macro-averaged F_1-values of all recurrent methods also become twice as high as in the case of randomly initialized word vectors.

Method	**Positive**			**Negative**			**Neutral**			**Macro**	**Micro**
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
RAE	$0.61^{-0.03}$	$0.61^{-0.17}$	$0.61^{-0.09}$	$0.22^{-0.16}$	$0.01^{-0.03}$	$0.03^{-0.05}$	$0.48^{-0.09}$	$0.72^{-0.04}$	$0.57^{-0.05}$	$0.32^{-0.07}$	$0.54^{-0.07}$
RNTN	0.45	$0.82^{-0.05}$	0.59	$0.24^{+0.05}$	$0.06^{-0.04}$	$0.1^{-0.07}$	$0.43^{+0.09}$	$0.17^{+0.07}$	$0.24^{+0.09}$	$0.34^{+0.03}$	$0.44^{-0.01}$
SEV	0.73	$0.74^{-0.05}$	$0.74^{-0.02}$	$0.0^{-0.41}$	$0.0^{-0.52}$	$0.0^{-0.46}$	$0.56^{-0.16}$	$0.84^{+0.29}$	$0.68^{+0.06}$	$0.37^{-0.24}$	$0.64^{-0.01}$
BAZ	$0.82^{+0.37}$	$0.72^{-0.28}$	$0.77^{+0.15}$	$0.62^{+0.62}$	$0.49^{+0.49}$	$0.55^{+0.55}$	$0.68^{+0.68}$	$0.85^{+0.85}$	$0.76^{+0.76}$	$0.66^{+0.35}$	$0.73^{+0.28}$
LBA$^{(1)}$	$0.76^{-0.06}$	$0.84^{+0.11}$	$0.79^{+0.02}$	$0.6^{+0.6}$	$0.56^{+0.56}$	$0.58^{+0.58}$	$0.75^{+0.19}$	$0.68^{-0.24}$	$0.72^{+0.03}$	$0.69^{+0.3}$	$0.73^{+0.07}$
LBA$^{(2)}$	$0.84^{+0.39}$	$0.73^{-0.27}$	$0.78^{+0.16}$	$0.57^{+0.57}$	$0.48^{+0.48}$	$0.53^{+0.53}$	$0.66^{+0.66}$	$0.82^{+0.82}$	$0.73^{+0.73}$	$0.65^{+0.34}$	$0.72^{+0.27}$
SB10k											
RAE	$0.5^{-0.13}$	$0.73^{+0.16}$	$0.59^{-0.01}$	$0.35^{+0.35}$	$0.06^{+0.06}$	$0.1^{+0.1}$	$0.8^{+0.05}$	$0.8^{-0.14}$	$0.8^{-0.03}$	$0.35^{+0.15}$	$0.68^{-0.04}$
RNTN	$0.0^{-0.02}$	$0.0^{-0.03}$	$0.0^{-0.05}$	$0.0^{-0.07}$	$0.0^{-0.01}$	$0.0^{-0.02}$	0.62	$1.0^{-0.06}$	$0.77^{-0.02}$	$0.0^{-0.03}$	$0.62^{+0.03}$
SEV	$0.64^{+0.64}$	$0.58^{+0.58}$	$0.61^{+0.61}$	$0.51^{+0.51}$	$0.21^{+0.21}$	$0.3^{+0.3}$	$0.76^{+0.14}$	$0.89^{-0.11}$	$0.82^{+0.05}$	$0.45^{+0.45}$	$0.72^{+0.1}$
BAZ	$0.72^{+0.03}$	$0.59^{+0.12}$	$0.65^{+0.07}$	$0.53^{+0.53}$	$0.33^{+0.33}$	$0.41^{+0.41}$	$0.79^{+0.08}$	$0.91^{-0.07}$	$0.84^{+0.01}$	$0.53^{+0.24}$	$0.75^{+0.03}$
LBA$^{(1)}$	$0.6^{-0.12}$	$0.72^{+0.14}$	$0.66^{+0.02}$	$0.47^{+0.47}$	$0.42^{+0.42}$	$0.44^{+0.44}$	$0.84^{+0.1}$	$0.8^{-0.17}$	$0.82^{+0.02}$	$0.55^{+0.23}$	$0.73^{-0.01}$
LBA$^{(2)}$	$0.72^{-0.04}$	$0.57^{+0.08}$	$0.64^{+0.04}$	$0.55^{+0.55}$	$0.39^{+0.39}$	$0.46^{+0.46}$	$0.79^{+0.07}$	$0.9^{-0.08}$	$0.84^{+0.01}$	$0.55^{+0.25}$	$0.75^{+0.03}$

Table 5.10: Results of deep-learning–based MLSA methods with least-squares embeddings

5.5.3 Error Analysis

Before we proceed with the evaluation of the second factor (larger training set), let us first analyze some errors that were specific to each of the classifiers trained with the least-squares embeddings.

Since interpreting and understanding the results of deep learning systems is a complex task due to a big number of model parameters and unobvious correlations between them, we decided to use the LIME package (Ribeiro et al., 2016), a recently proposed model-agnostic interpretation tool, to get a better intuition about the reasons of the classifiers' decisions. To derive an explanation for a particular prediction, LIME randomly removes or perturbs parts

of the input (in our case, tokens), estimating which of these modifications lead to the biggest changes in the output, and assigns corresponding class-specific association scores to each of the changed parts. The higher this score, the more predictive is the given feature for that particular label. For the sake of vividness, we have highlighted all tokens that, according to LIME, were associated with the neutral class as white, marked negative attributes with the blue background, and highlighted positively connoted words in green, reflecting the respective association strength with a higher color brightness.

The first incorrect prediction shown in Example 5.5.1 was made by the RAE system of Socher et al. (2011). As we can see from the visualization, the model correctly recognized the positive term "gefällt" (*to like*), but, unfortunately, this word is the only one which contributes to the right decision, and its learned weight is obviously not enough to outdo the effect of multiple neutral and negative items, such as "Grün" (*green*), "Schwarz" (*black*), and most surprisingly "PosSmiley%" (*PosSmiley%*), which unexpectedly is stronger associated with the negative semantic orientation than with the positive class. As a consequence of this, the classifier erroneously predicts the NEUTRAL label for the whole message, falling against the prevalence of allegedly objective terms.

Example 5.5.1 (An Error Made by the RAE System)

Tweet: Grün - Schwarz in meinem Bundesland. Gefällt mir doch sehr %PosSmiley

Green - Black in my state. Yet , I like it so much %PosSmiley

Gold Label: positive

Predicted Label: neutral*

A similar situation is also observed with the recurrent neural tensor, whose sample error is shown in Example 5.5.2. As we can see from the analysis, the bias towards the neutral class is even more pronounced this time, as virtually all of the terms in the tweet are highlighted in white. The only word which shows a minimal negative connotation is "tumblr," which indeed appeared twice in a negative tweet, two times in neutral messages, and once in a positive microblog in the training corpus. Nonetheless, even for this term the skewness towards the neutral orientation is still ten times bigger than its association with the negative polarity ($1.4e^{-4}$ versus $1.5e^{-5}$), which can be explained by the general prevalence of neutral messages in SB10k.

Example 5.5.2 (An Error Made by the RNTN System)

Tweet: tumblr people sind meine lieblings people %PosSmiley

tumblr people are my favorite people %PosSmiley

Gold Label: positive
Predicted Label: neutral*

A slightly different behavior is shown by the method of Severyn and Moschitti (2015b) on the PotTS corpus. This time, we can see at least two clearly positive words ("ist" [*is*] and "Freund" [*friend*]). However, the former of these terms is an auxiliary copular verb, which can hardly express any polarity, since it usually plays an auxiliary role and lacks any distinct lexical meaning. Nevertheless, the latter word ("Freund" [*friend*]) indeed conveys a positive feeling of its prepositional argument ("Iran") towards the subject of the sentence ("Syrien" [*Syria*]), but this positive effect is nullified by the author's statement that this friendship poses a problem. Unfortunately, the word "Problem" (*problem*) is recognized only as a neutral marker, just like many other terms in this microblog.

Example 5.5.3 (An Error Made by the SEV System)
Tweet: Syrien ist Freund von Iran , das ist das Problem ! annewill
Syria is a friend of Iran . That 's the problem ! annewill

Gold Label: negative
Predicted Label: neutral*

In Example 5.5.4, we can see another error made by the system of Baziotis et al. (2017). This time, again, we observe the prevalence of positive and neutral items, with the only exception being the possessive pronoun "meinen" (*my*), which, according to the classifier, indicates negative polarity. Apart from this term, we also can notice several inaccuracies at recognizing positive and neutral features: For example, the pronominal adverb "darin" (*in it*) is the strongest positive trait, whose predictiveness is even higher than the scores of the words "singen" (*to sing*) and "Liebeslied" (*love song*). This contradicts the fact that pronominal adverbs by themselves do not express any semantic orientation, all the more as in this case the antecedent of the adverb (the noun "Kleiderschrank" [*wardrobe*]) is recognized as a neutral item. On the other hand, the modal verb "wollte" (*wanted*) is considered as an objective term, although it has a slight positive connotation as it expresses a wish of the author.

Example 5.5.4 (An Error Made by the BAZ System)
Tweet: Wollte meinen Kleiderschrank aufräumen . . . sitze nun darin und singe Liebeslieder . . .
Wanted to clean up my wardrobe . . . Now sitting in it and singing love songs . . .

Gold Label: neutral
Predicted Label: positive*

Finally, Example 5.5.5 shows an incorrect prediction of our lexicon-based attention system. In contrast to the previous two methods, the positive information is much more condensed in this case and represented by a single term "super." Surprisingly, this term outweighs a whole bunch of neutral features such as "gerade" (*right now*), "Lust haben" (*to be up to*), "was" (*something*) etc. Admittedly, the first part of this message indeed expresses a positive attitude of the author, but this effect is invalidated by the second clause, which shows the impossibility of that wish.

Example 5.5.5 (An Error Made by the LBA System)

Tweet: Gerade super Lust , mit Carls Haaren was zu machen aber ca 300 km Distanz halten mich davon ab .

Super up to do something with Carl 's hair right now , but ca. 300 km distance keep me off from this.

Gold Label: **neutral**

Predicted Label: **positive***

5.6 Evaluation

Now that we have familiarized ourselves with the peculiarities and results of the most prominent sentiment analysis approaches from all method groups (lexicon-, machine-learning– and deep-learning–based ones), let us have a closer look at how changing different common parameters of these methods might affect their performance. In particular, we would like to see whether increasing the amount of the training data, switching to a different type of sentiment lexicon, or using unnormalized text as input would improve or, vice versa, lower the classification scores.

5.6.1 Weak Supervision

The first avenue that we are going to explore in this evaluation is the effect of weakly supervised data—an additional collection of training tweets that have been automatically labeled with sentiment tags based on the occurrence of some sufficiently reliable formal criteria, such as emoticons or hashtags.

Among the first who proposed the idea of training a sentiment classifier on a larger corpus of automatically annotated messages was Read (2005), who gathered a set of 766,000 Usenet posts containing frownies or smileys, assigned a polarity label to each of these posts, judging by the type of the emoticons, and subsequently used a subset of these documents (22,000 posts) to optimize a Naïve Bayes and SVM system. Even though these classifiers could

achieve a considerable accuracy (up to 70%) on predicting noisy labels of the remaining posts, they could not generalize to texts from other genres (movie reviews and newswire articles) where they hardly outperformed the random-chance baseline. With the onset of the Twitter era, this idea of weak supervision has experienced its renaissance with the works of Go et al. (2009), Pak and Paroubek (2010), and Barbosa and Feng (2010).

In order to check the effect of such noisily annotated data on our tested methods, we also automatically labeled all messages from the German Twitter Snapshot (Scheffler, 2014) based on the occurrences of smileys: In particular, we considered a microblog as positive if its normalized version contained the token `%PositiveSmiley` with no other facial expressions. Likewise, we regarded a message as negative if the only emoticon in this tweet was `%NegativeSmiley`. We skipped all posts that contained both types of smileys, and assigned the rest of the messages to the neutral class. (A detailed breakdown of the final distribution is given in Table 5.1 at the beginning of this chapter.)

Since it was impossible to utilize the whole snapshot for the training due to limited computational resources (only reading the dataset into memory would require 9.3Gb RAM, not to mention the space required for storing the embeddings and features), we confined ourselves to one sixth of these data, which still resulted in 4 M messages. Furthermore, to mitigate the extreme skewness of this corpus, we downsampled positive and neutral tweets to get an equal number of instances for all classes (59,000 microblogs for each polarity) and used these examples in addition to the manually analyzed PotTS and SB10k tweets.

Since lexicon-based approaches were mostly independent of the training set, we decided to rerun our experiments only with ML- and fastest DL-based methods (RNN, SEV, BAZ, and LBA),[11] which still incurred running times up to five days for some systems. The results of this evaluation are shown in Table 5.11.

As we can see from the scores, apart from improved precision of positive tweets and higher recall of negative microblogs, adding noisily labeled messages to the training set has a strong negative effect on the results of all methods, with the biggest drops demonstrated by the approach of Baziotis et al. (-0.5 macro-F_1 and -0.43 micro-F_1 on the PotTS corpus; -0.34 macro-F_1 and -0.51 micro-F_1 on the SB10k dataset) and our own LBA$^{(2)}$ solution (-0.42 macro-F_1-score and -0.5 micro-F_1 on the PotTS test set; -0.43 macro-F_1 and -0.61 micro-F_1 on the SB10k data), which both fail to predict any neutral message on PotTS and always assign the same polarity to all SB10k tweets. Less severe, but still substantial degradation is also observed with the machine-learning systems of Mohammad et al. and Günther and Furrer as well as our DL-based LBA$^{(1)}$ method, whose macro-averaged F_1-scores go down by

[11] In all subsequent evaluation experiments with DL-based systems, we will use pre-trained word2vec vectors (if applicable) with the least-squares fallback, and compare the results of these approaches to the respective scores in Table 5.10.

Method	Positive			Negative			Neutral			Macro	Micro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
GMN	$0.8^{+0.13}$	$0.34^{-0.39}$	$0.48^{-0.22}$	$0.2^{-0.15}$	$0.29^{+0.14}$	$0.24^{-0.03}$	$0.53^{-0.07}$	$0.79^{+0.07}$	$0.63^{-0.03}$	$0.36^{-0.01}$	$0.49^{-0.12}$
MHM	$0.86^{+0.07}$	$0.59^{-0.18}$	$0.7^{-0.08}$	$0.31^{-0.27}$	$0.39^{-0.17}$	$0.35^{-0.22}$	$0.55^{-0.18}$	$0.68^{-0.08}$	$0.61^{-0.13}$	$0.52^{-0.15}$	$0.59^{-0.14}$
GNT	$0.86^{+0.15}$	$0.6^{-0.2}$	$0.71^{-0.04}$	$0.26^{-0.29}$	$0.31^{-0.14}$	$0.28^{-0.22}$	$0.53^{-0.15}$	$0.68^{-0.05}$	$0.59^{-0.06}$	$0.5^{-0.12}$	$0.57^{-0.1}$
RAE	$0.68^{+0.07}$	$0.31^{-0.3}$	$0.43^{-0.18}$	$0.25^{+0.03}$	$0.46^{+0.45}$	$0.32^{+0.29}$	$0.49^{+0.01}$	$0.61^{-0.11}$	$0.54^{-0.03}$	$0.38^{+0.06}$	$0.45^{-0.09}$
SEV	$0.87^{+0.14}$	$0.51^{-0.23}$	$0.64^{-0.1}$	$0.27^{+0.27}$	$0.49^{+0.49}$	$0.35^{+0.35}$	$0.55^{-0.01}$	$0.58^{-0.26}$	$0.56^{-0.12}$	$0.49^{+0.12}$	$0.53^{-0.11}$
BAZ	$0.0^{-0.82}$	$0.0^{-0.72}$	$0.0^{-0.77}$	$0.19^{-0.43}$	$1.0^{+0.51}$	$0.32^{-0.23}$	$0.0^{-0.68}$	$0.0^{-0.85}$	$0.0^{-0.76}$	$0.16^{-0.5}$	$0.19^{-0.43}$
$LBA^{(1)}$	$0.48^{-0.28}$	$0.88^{+0.04}$	$0.62^{-0.17}$	$0.25^{-0.35}$	$0.23^{-0.33}$	$0.24^{-0.34}$	$0.0^{-0.75}$	$0.0^{-0.68}$	$0.0^{-0.72}$	$0.43^{-0.26}$	$0.44^{-0.29}$
$LBA^{(2)}$	$0.91^{+0.07}$	$0.08^{-0.65}$	$0.14^{-0.64}$	$0.19^{-0.38}$	$0.99^{+0.51}$	$0.32^{-0.21}$	$0.0^{-0.66}$	$0.0^{-0.82}$	$0.0^{-0.73}$	$0.23^{-0.42}$	$0.22^{-0.5}$
SB10k											
GMN	$0.71^{+0.06}$	$0.27^{-0.18}$	$0.4^{-0.13}$	$0.24^{-0.14}$	$0.11^{+0.03}$	$0.15^{+0.02}$	$0.71^{-0.01}$	$0.96^{+0.03}$	$0.82^{-0.01}$	$0.27^{-0.06}$	$0.68^{-0.02}$
MHM	$0.77^{+0.06}$	$0.4^{-0.25}$	$0.53^{-0.15}$	$0.61^{-0.1}$	$0.1^{-0.3}$	$0.18^{-0.27}$	$0.71^{-0.09}$	$0.97^{-0.1}$	$0.82^{-0.02}$	$0.35^{-0.21}$	$0.71^{-0.04}$
GNT	$0.77^{+0.1}$	$0.39^{-0.23}$	$0.52^{-0.12}$	$0.25^{-0.19}$	$0.13^{-0.15}$	$0.17^{-0.17}$	$0.71^{-0.07}$	$0.92^{+0.05}$	$0.8^{-0.02}$	$0.34^{-0.15}$	$0.68^{-0.04}$
RAE	$0.44^{-0.06}$	$0.27^{-0.51}$	$0.34^{-0.25}$	$0.24^{-0.11}$	$0.59^{+0.53}$	$0.34^{+0.24}$	$0.78^{-0.02}$	$0.62^{-0.18}$	$0.69^{-0.11}$	$0.34^{-0.01}$	$0.54^{-0.14}$
SEV	0.64	$0.39^{-0.19}$	$0.49^{-0.12}$	$0.34^{-0.17}$	$0.12^{-0.09}$	$0.18^{-0.12}$	$0.7^{-0.06}$	$0.9^{+0.01}$	$0.78^{-0.04}$	$0.33^{-0.12}$	$0.69^{-0.03}$
BAZ	$0.24^{-0.48}$	$1.0^{+0.41}$	$0.38^{-0.27}$	$0.0^{-0.53}$	$0.0^{-0.33}$	$0.0^{-0.41}$	$0.0^{-0.79}$	$0.0^{-0.91}$	$0.0^{-0.84}$	$0.19^{-0.34}$	$0.24^{-0.51}$
$LBA^{(1)}$	$0.64^{+0.04}$	$0.43^{-0.29}$	$0.52^{-0.14}$	$0.59^{+0.12}$	$0.09^{-0.33}$	$0.16^{-0.28}$	$0.71^{-0.13}$	$0.93^{+0.13}$	$0.8^{-0.02}$	$0.34^{-0.21}$	$0.69^{-0.04}$
$LBA^{(2)}$	$0.0^{-0.72}$	$0.0^{-0.57}$	$0.0^{-0.64}$	$0.14^{-0.41}$	$1.0^{+0.61}$	$0.25^{-0.21}$	$0.0^{-0.79}$	$0.0^{-0.9}$	$0.0^{-0.84}$	$0.12^{-0.43}$	$0.14^{-0.61}$

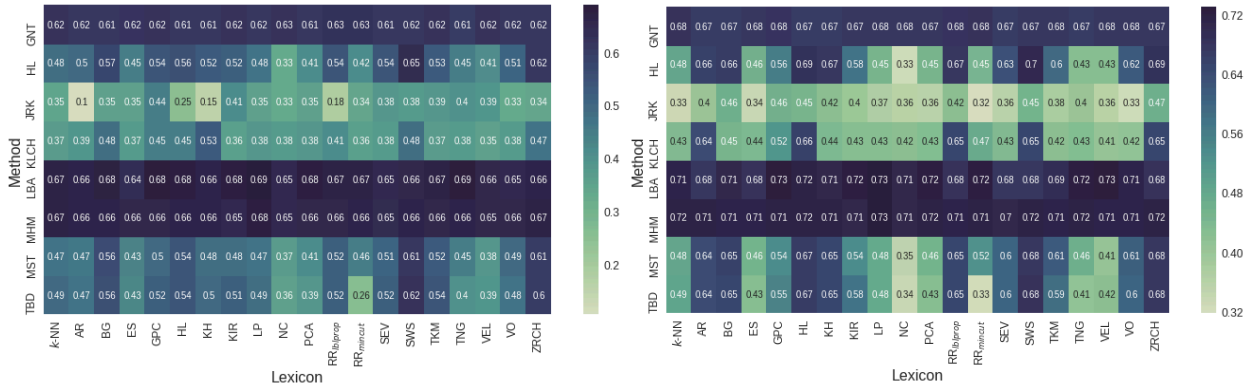
Table 5.11: Results of MLSA methods with weakly supervised data

0.15, 0.12, and 0.26 points on the former corpus and sink by 0.21, 0.15, 0.21, respectively, on the latter dataset. The micro-averaged F_1-results of these methods, however, decrease to a much smaller degree, since the main drops happen on the negative class, which is by far the least represented polarity in both corpora. The micro-averages of the remaining systems seem to be affected even less, but are still worse than the results obtained without the snapshot data. We hypothesize that the main reason for this decrease is a substantial difference between the class distributions in noisily annotated training tweets and manually labeled test sets, which overly bias classifiers' predictions.

5.6.2 Lexicons

Another factor that could significantly affect the results of some systems was the sentiment lexicon that these systems used either directly, for computing the polarity of a message (*e.g.*, lexicon-based approaches), or indirectly, as features or attention scores (*e.g.*, ML- and DL-based techniques). To estimate the effect of this resource, we successively replaced the lexicons that we used in our previous experiments with other polarity lists presented in Chapter 3, and recomputed the scores of the tested systems.

As we can see in Figure 5.3, the system of Mohammad et al. (2013) and our own lexicon-based attention approach clearly outperform all other competitors on the PotTS corpus independent of the lexicon they use. The only method that comes at least close to their results is the ML-based classifier of Günther et al. (2014), which is still almost 5% below



(a) `Macro`-F_1 *(b)* `Micro`-F_1

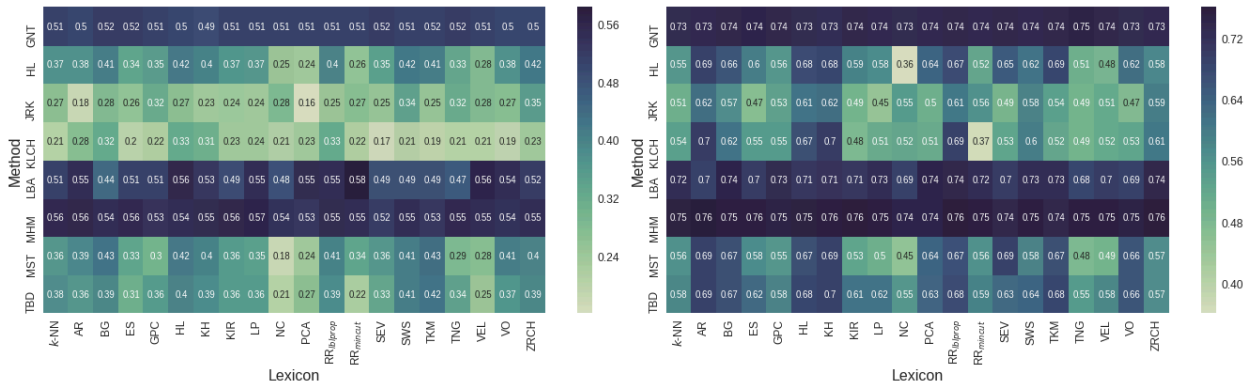
Figure 5.3: Results of MLSA methods with different lexicons on the PotTS corpus

the average macro-F_1 of these two classifiers. The same also applies to the micro-F_1-scores, where the solution of Günther et al. (2014) loses almost 3% on average to the two top performers. Regarding the differences between the MHM and LBA themselves, we can observe a rather mixed relation: The approach of Mohammad et al. (2013) yields better macro-averaged F_1-results with the lexicons of Esuli and Sebastiani (2005), Vo and Zhang (2016), and Clematide and Klenner (2010), but falls against LBA when used with the polarity lists of Blair-Goldensohn et al. (2008), Waltinger (2010), Hu and Liu (2004), Kiritchenko et al. (2014), Rao and Ravichandran (2009), Takamura et al. (2005), Tang et al. (2014b), and Velikovich et al. (2010) as well as the NWE-based LinProj and PCA lexicons. Moreover, when trained with the polarity list of Tang et al. and our LinProj lexicon, the LBA system achieves the best overall macro-F_1 on this corpus.

These results, however, look slightly differently when we consider the micro-averaged scores. This time, the system of Mohammad et al. outperforms our solution in eight out of twenty cases, but performs worse than LBA with four other polarity lists (GPC, KIR, $\text{RR}_{\text{mincut}}$, and VEL). Nevertheless, our approach still reaches the best overall observed score (0.73) with three tested resources (GPC, LinProj, and VEL).

Regarding the performance of single lexicons, we can see that the best results are achieved with the manually curated SentiWS (Remus et al., 2010) and Zurich Polarity List (Clematide and Klenner, 2010), followed by the dictionary-based approaches of Blair-Goldensohn et al. (2008) and Rao and Ravichandran (2009). The method of the nearest centroids vice versa appears to be of the lowest utility for almost all systems, even though it demonstrated quite acceptable scores in our initial intrinsic evaluation.

A similar situation also holds for the SB10k corpus, where the ML-based approaches of Mohammad et al. (2013) and Günther et al. (2014) and our proposed LBA system outperform all other methods in terms of both macro- and micro-averaged F_1-scores. This time, however, the average difference between the macro-results of LBA and GNT is much smaller



(a) `Macro`-F_1 *(b)* `Micro`-F_1

Figure 5.4: Results of MLSA methods with different lexicons on the SB10k corpus

and amounts to only 0.02% in favor of LBA, which again achieves the best overall macro-F_1 (0.58) in combination with the min-cut lexicon of Rao and Ravichandran (2009). Unfortunately, our system clearly falls against the latter classifier with respect to the micro-averaged scores, performing worse than it in 16 out of 20 experiments.

The effect of single lexicons is also less pronounced than in the PotTS case, as all of the tested polarity lists show a more or less similar behavior, especially regarding the macro-averaged F_1-score. In terms of the micro-F_1, however, we can observe that dictionary-based lists, especially those of Awadallah and Radev (2010), Blair-Goldensohn et al. (2008), Hu and Liu (2004), and Kim and Hovy (2004), lead to generally better scores than corpus- and NWE-based resources.

5.6.3 Text Normalization

Finally, the last aspect that we are going to analyze in this evaluation is the effect of the text normalization, which we applied to the input messages before passing them to the classifiers. To verify the utility of this step, we rerun all experiments from the initial sections, using the original Twitter messages instead of their preprocessed forms, and recalculated the results of the tested systems.

As we can see from the figures in Table 5.12, switching off the normalization has a strong negative effect on the scores of almost all approaches except for the methods of Yessenalina et al. (2010) and Socher et al. (2012, 2013), which notoriously keep predicting the majority class in most of the cases in the same way as they did before. Apart from this, we can notice that the lexicon-based systems (HL, JRK, KLCH, MST, and TBD) suffer the greatest loss in terms of both macro- and micro-averaged F_1-scores on the PotTS corpus (up to -0.25 macro- and and -0.22 micro-F_1). A closer look at their errors revealed that this deterioration is mostly due to the increased variety of different emoticons in the dataset (which were typically

unified during the preprocessing) and the absence of these forms in the utilized polarity list. The second biggest quality drop is demonstrated by the DL-based approaches BAZ, LBA$^{(1)}$, and LBA$^{(2)}$, which apparently also got confused by the higher lexical variety of the input and failed to optimize all their internal parameters to properly fit this diversity. The remaining DL- and ML-based classifiers (especially those of MHM, GNT, and RNTN) seem to be more resistant to the introduced changes, but still show a decrease by up to 0.04 macro- and 0.08 micro-F_1. The only exception in this case is the MVRNN system of Socher et al. (2012), which slightly improves on the negative and neutral classes, leaving the majority class pitfall. Unfortunately, this increase appears to be too small to positively influence the overall statistics of this method.

Regarding the breakdown of single polarity classes, we can see that most of the rare improvements affect the recall of positive and neutral messages, with the biggest gains demonstrated by the RAE and RNTN approaches (+0.37 and +0.11, respectively). Other positive changes are fairly sporadic and produced by only few classifiers (first of all, MVRNN). Nevertheless, even in these exceptional cases, the improvements are typically so small that they hardly outweigh the decreased scores on other aspects and have virtually no effect on the net results for all classes.

A similar situation also happens on the SB10k corpus, where we can see even fewer improvements (in 10 out of 176 cases). The biggest increase this time (+0.16 recall) is demonstrated by the approach of Baziotis et al. (2017) on the negative class. The remaining growths, however, are much smaller and typically range between one and seven percent. On the other hand, three of the tested methods (Y&C, MVRNN, and RNTN) have exactly the same results as they did previously with normalized messages, although, most of the time, these classifiers only predict the majority label anyway. As to the rest of the systems, we can see that their scores are notably lower than in our initial experiments, but the decrease is much smaller in comparison with the PotTS corpus. A sad exception in this case is a major drop of the recall of neutral messages (−0.79) demonstrated by our LBA$^{(1)}$ system, which, in turn, results in a significant decrease of its macro- and micro-averaged F_1-scores (−0.14 and −0.46, respectively). Other approaches (including the sibling method LBA$^{(2)}$) behave much more stable in this regard and their average decrease amounts to −0.06 macro- and −0.03 micro-F_1.

Similar to the results on the PotTS data, most of the gains are concentrated at the recall of the neutral class (four out of ten improvements), with the other positive changes being rather sporadic and affecting only a few classifiers. Nevertheless, unlike in the previous case, this time, we can even observe a slight improvement of the macro-averaged F_1-measure for one of the systems (the lexicon-based approach of Jurek et al.), but its micro-averaged metric remains mainly unaffected by this increase. In general, however, the vast majority of macro-

Method	Positive			Negative			Neutral			Macro	Micro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
HL	$0.63^{-0.12}$	$0.3^{-0.46}$	$0.4^{-0.36}$	$0.46^{-0.07}$	$0.29^{-0.14}$	$0.36^{-0.11}$	$0.41^{-0.26}$	$0.77^{+0.04}$	$0.54^{-0.15}$	$0.38^{-0.24}$	$0.464^{-0.22}$
TBD	$0.65^{-0.12}$	$0.24^{-0.47}$	$0.36^{-0.38}$	$0.46^{-0.08}$	$0.27^{-0.12}$	$0.34^{-0.11}$	$0.41^{-0.22}$	$0.83^{+0.06}$	$0.55^{-0.14}$	$0.348^{-0.25}$	$0.457^{-0.22}$
MST	$0.63^{-0.12}$	$0.29^{-0.43}$	$0.4^{-0.34}$	$0.47^{-0.01}$	$0.34^{-0.13}$	$0.39^{-0.09}$	$0.42^{-0.26}$	$0.77^{+0.05}$	$0.54^{-0.16}$	$0.4^{-0.21}$	$0.47^{-0.21}$
JRK	$0.44^{-0.16}$	$0.22^{-0.09}$	$0.29^{-0.12}$	$0.14^{-0.28}$	$0.06^{-0.14}$	$0.08^{-0.19}$	$0.36^{-0.07}$	$0.7^{-0.1}$	$0.47^{-0.09}$	$0.19^{-0.15}$	$0.36^{-0.11}$
KLCH	$0.61^{-0.1}$	$0.23^{-0.49}$	$0.33^{-0.38}$	$0.33^{-0.01}$	$0.21^{+0.04}$	$0.26^{+0.04}$	$0.41^{-0.25}$	0.82	$0.55^{-0.18}$	$0.3^{-0.17}$	$0.44^{-0.21}$
GMN	$0.59^{-0.08}$	$0.77^{+0.04}$	$0.66^{-0.04}$	$0.37^{-0.02}$	$0.14^{-0.01}$	$0.2^{-0.01}$	$0.57^{-0.03}$	$0.55^{-0.17}$	$0.56^{-0.1}$	$0.43^{-0.02}$	$0.57^{-0.05}$
MHM	$0.78^{-0.01}$	$0.76^{-0.01}$	$0.77^{-0.01}$	$0.59^{+0.01}$	$0.54^{-0.02}$	$0.56^{-0.01}$	$0.7^{-0.03}$	$0.74^{-0.02}$	$0.72^{-0.02}$	$0.67^{-0.006}$	$0.71^{-0.007}$
GNT	$0.68^{-0.03}$	0.8	$0.73^{-0.02}$	0.55	$0.43^{-0.02}$	$0.48^{-0.02}$	$0.67^{-0.01}$	$0.59^{-0.04}$	$0.62^{-0.03}$	$0.61^{-0.017}$	$0.65^{-0.02}$
Y&C	0.45	1.0	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.45
RAE	$0.46^{-0.15}$	$0.98^{+0.37}$	$0.62^{+0.01}$	$0.0^{-0.22}$	$0.0^{-0.01}$	$0.0^{-0.03}$	$0.63^{+0.15}$	$0.05^{-0.67}$	$0.09^{-0.48}$	$0.31^{-0.01}$	$0.46^{-0.08}$
MVRNN	0.45	$0.92^{-0.08}$	$0.6^{-0.02}$	$0.08^{+0.08}$	$0.01^{+0.01}$	$0.01^{+0.01}$	$0.26^{+0.26}$	$0.03^{+0.03}$	$0.06^{+0.06}$	0.31	$0.43^{-0.02}$
RNTN	0.45	$0.93^{+0.11}$	$0.61^{+0.02}$	$0.29^{+0.05}$	$0.01^{-0.05}$	$0.01^{-0.09}$	$0.4^{-0.03}$	$0.07^{-0.1}$	$0.12^{-0.12}$	$0.31^{-0.03}$	$0.45^{-0.01}$
SEV	$0.56^{-0.17}$	$0.79^{+0.05}$	$0.66^{-0.08}$	0.0	0.0	0.0	$0.57^{+0.01}$	$0.57^{-0.27}$	$0.57^{-0.11}$	$0.33^{-0.04}$	$0.56^{-0.08}$
BAZ	$0.65^{-0.17}$	$0.59^{-0.13}$	$0.62^{-0.15}$	0.62	$0.22^{-0.27}$	$0.32^{-0.23}$	$0.5^{-0.18}$	$0.74^{-0.11}$	$0.6^{-0.16}$	$0.47^{-0.19}$	$0.57^{-0.16}$
$LBA^{(1)}$	$0.58^{-0.18}$	$0.77^{-0.07}$	$0.66^{-0.13}$	$0.54^{-0.06}$	$0.53^{-0.03}$	$0.54^{-0.04}$	$0.63^{-0.12}$	$0.37^{-0.31}$	$0.46^{-0.26}$	$0.6^{-0.09}$	$0.58^{-0.15}$
$LBA^{(2)}$	$0.67^{-0.17}$	$0.52^{-0.21}$	$0.59^{-0.19}$	$0.51^{-0.06}$	$0.44^{-0.04}$	$0.47^{-0.06}$	$0.52^{-0.14}$	$0.7^{-0.12}$	$0.6^{-0.13}$	$0.53^{-0.12}$	$0.57^{-0.15}$
SB10k											
HL	$0.41^{-0.08}$	$0.42^{-0.2}$	$0.42^{-0.13}$	$0.24^{-0.03}$	$0.28^{-0.06}$	$0.26^{-0.04}$	$0.66^{-0.07}$	$0.63^{-0.01}$	$0.65^{-0.02}$	$0.34^{-0.08}$	$0.53^{-0.05}$
TBD	$0.41^{-0.07}$	$0.37^{-0.23}$	$0.39^{-0.14}$	$0.21^{-0.03}$	$0.24^{-0.03}$	$0.22^{-0.03}$	$0.65^{-0.07}$	$0.66^{+0.03}$	$0.66^{-0.01}$	$0.31^{-0.08}$	$0.53^{-0.04}$
MST	$0.4^{-0.05}$	$0.32^{-0.17}$	$0.35^{-0.12}$	$0.26^{-0.03}$	$0.3^{-0.05}$	$0.28^{-0.04}$	$0.65^{-0.05}$	$0.68^{-0.04}$	0.67	$0.32^{-0.08}$	$0.54^{-0.03}$
JRK	$0.4^{-0.01}$	$0.42^{-0.03}$	$0.41^{-0.01}$	0.36	0.26	0.3	0.69	$0.72^{-0.03}$	$0.71^{-0.01}$	$0.36^{+0.01}$	$0.59^{-0.006}$
KLCH	$0.42^{+0.03}$	$0.21^{-0.01}$	0.28	$0.25^{-0.09}$	0.13	$0.17^{-0.02}$	0.66	0.86	0.75	$0.23^{-0.005}$	$0.6^{-0.002}$
GMN	$0.48^{-0.17}$	$0.31^{-0.14}$	$0.37^{-0.16}$	$0.27^{-0.11}$	$0.07^{-0.01}$	$0.11^{-0.02}$	$0.69^{-0.03}$	$0.9^{-0.03}$	$0.78^{-0.03}$	$0.24^{-0.09}$	$0.64^{-0.06}$
MHM	$0.67^{-0.04}$	$0.62^{-0.03}$	$0.65^{-0.03}$	$0.59^{-0.08}$	$0.42^{-0.02}$	$0.49^{-0.04}$	0.8	$0.88^{-0.01}$	0.84	$0.56^{-0.002}$	$0.75^{-0.001}$
GNT	$0.42^{-0.25}$	$0.21^{-0.41}$	$0.28^{-0.36}$	$0.25^{-0.19}$	$0.13^{-0.15}$	$0.17^{-0.17}$	$0.66^{-0.12}$	$0.86^{-0.01}$	$0.75^{-0.07}$	$0.22^{-0.2}$	$0.604^{-0.12}$
Y&C	0.0	0.0	0.0	0.0	0.0	0.0	0.62	1.0	0.77	0.0	0.62
RAE	$0.46^{-0.04}$	$0.62^{-0.11}$	$0.53^{-0.06}$	$0.18^{-0.17}$	$0.02^{-0.04}$	$0.03^{-0.07}$	$0.77^{-0.03}$	$0.82^{+0.02}$	$0.79^{-0.01}$	$0.28^{-0.07}$	$0.66^{-0.02}$
MVRNN	0.19	0.01	0.03	0.0	0.0	0.0	0.62	0.97	0.76	0.01	0.61
RNTN	0.0	0.0	0.0	0.0	0.0	0.0	0.62	1.0	0.77	0.0	0.62
SEV	$0.58^{-0.06}$	$0.39^{-0.19}$	$0.47^{-0.14}$	$0.23^{-0.28}$	$0.05^{-0.16}$	$0.08^{-0.22}$	$0.7^{-0.06}$	$0.92^{+0.03}$	$0.8^{-0.02}$	$0.27^{-0.18}$	$0.67^{-0.05}$
BAZ	$0.69^{-0.03}$	$0.54^{-0.16}$	$0.6^{-0.05}$	$0.36^{-0.17}$	$0.49^{+0.16}$	0.41	0.79	$0.79^{-0.12}$	$0.79^{-0.05}$	$0.51^{-0.02}$	$0.69^{-0.06}$
$LBA^{(1)}$	$0.24^{-0.36}$	$0.86^{+0.14}$	$0.38^{-0.28}$	$0.45^{-0.02}$	$0.45^{+0.03}$	$0.45^{+0.01}$	$0.69^{-0.15}$	$0.01^{-0.79}$	$0.02^{-0.8}$	$0.41^{-0.14}$	$0.27^{-0.46}$
$LBA^{(2)}$	$0.74^{-0.02}$	$0.42^{-0.15}$	$0.54^{-0.1}$	$0.62^{+0.07}$	$0.25^{-0.14}$	$0.35^{-0.11}$	$0.73^{-0.06}$	$0.95^{+0.05}$	$0.82^{-0.02}$	$0.45^{-0.1}$	$0.72^{-0.03}$

Table 5.12: Results of MLSA methods without text normalization

and micro-F_1-scores show an obvious decline on both datasets, which once again proves the advantage of preprocessing.

5.7 Summary and Conclusions

Now the we have reached the end of the chapter, we would like to remind the reader that in this part of the thesis we have made the following findings and contributions:

- we have compared three major families of message-level sentiment analysis methods: lexicon-, machine-learning– and deep-learning–based ones, finding that the last two

groups significantly outperform lexicon-driven systems;

- surprisingly, among all compared lexicon methods, the most simple one (the classifier of Hu and Liu [2004]) produced the best macro- and micro-averaged F_1-results on the PotTS corpus (0.615 and 0.685, respectively) and also yielded the highest macro F_1-measure on the SB10k dataset (0.421). Other systems, however, could have improved their scores if they better handled the negation of polar terms (after switching off the negation component in the method of Musto et al., its macro-F_1 on the PotTS corpus increased to 0.641, surpassing the benchmark of Hu and Liu);

- as expected, the ML-based system of Mohammad et al. (2013)—the winner of the inaugural run of SemEval task in sentiment analysis of Twitter (Nakov et al., 2013)—also surpassed other ML competitors, achieving highly competitive results: 0.674 macro- and 0.727 micro-F_1 on the PotTS data, and 0.564 macro- and 0.752 micro-averaged F_1-measure on the SB10k test set;

- as in the previous case, however, these results could have been improved if the classifier dispensed with character-level and part-of-speech features and used logistic regression instead of SVM;

- a much more varied situation was observed with deep-learning–based systems, which frequently simply fell into always predicting the majority class for all tweets, but sometimes yielded extraordinarily good results as it was the case with our proposed lexicon-based attention system, which attained 0.69 macro-F_1 on the PotTS corpus and 0.55 macro F_1-score on the SB10k dataset (0.73 and 0.75 micro-F_1, respectively), setting a new state of the art for the former data;

- speaking of word embeddings, we should note that almost all DL-based approaches showed fairly low scores when they used randomly initialized task-specific embeddings, but notably improved their results after switching to pre-trained word2vec vectors, and benefited even more from the least-squares fallback;

- against our expectations, we could not overcome the majority class pitfall of DL-based systems after adding more weakly supervised training data, which, in general, only lowered the scores of both ML- and DL-based methods. Since this result contradicts the findings of other authors, we hypothesize that this degradation is primarily due to the differences in the class distributions between automatically and manually labeled tweets;

- on the other hand, we could see that using more qualitative sentiment lexicons (especially manually curated and dictionary-based ones) resulted in further improvements for the systems that relied on this lexical resource;

- last but not least, we proved the utility of the text normalization step, which brought about significant improvements for all tested methods, as confirmed by our last ablation test.

Chapter 6

Discourse-Aware Sentiment Analysis

Although message-level sentiment analysis methods do a fairly good job at classifying the overall polarity of a message, a crucial limitation of all these systems is that they completely overlook the structural nature of their input by either considering it as a single whole (*e.g.*, bag-of-features approaches) or analyzing it as a monotone sequence of equally important elements (*e.g.*, recurrent neural methods). Unfortunately, both of these solutions violate the hierarchical principle of language (de Saussure and Engler, 1990; Hjelmslev, 1970), which states that complex linguistic units are formed from smaller language elements in the bottom-up way, *e.g.*, words are created by putting together morphemes, sentences are made of several words, and discourses are composed of multiple coherent sentences. Moreover, apart from this inherent structural heterogeneity, even units of the same linguistic level might play a different role and be of unequal importance when joined syntagmatically into the higher-level whole. For example, in words, the root morpheme typically conveys more lexical meaning than the affixes; in sentences, the syntactic head usually dominates its grammatical dependents; and, in discourse, one of the sentences frequently expresses more relevant ideas than the rest of the text.

Exactly the lack of discourse information was one of the main reasons for the misclassifications made by the systems of Severyn and Moschitti (2015b), Baziotis et al. (2017), and our own LBA method in Examples 5.5.3, 5.5.4, and 5.5.5. Since none of these approaches explicitly took discourse structure into account, we decided to check whether making the last of these solutions (the LBA classifier) aware of discourse phenomena would improve its results. But before we present these experiments, we first would like to make a short digression into the theory of discourse and give an overview of the most popular approaches to text-level analysis that exist in the literature nowadays. Afterwards, in Section 6.2, we will describe the way how we inferred discourse information for PotTS and SB10k tweets. Then, in Section 6.3, we will summarize the current state of the art in discourse-aware sentiment analysis (DASA) and also present our own methods, evaluating them on the aforementioned

datasets. After analyzing the effects of various common factors (such as the impact of the underlying sentiment classifier and the amenability of various discourse relation schemes to different DASA approaches), we will recap the results and summarize our findings in the last part of this chapter.

6.1 Discourse Analysis

Since the main focus of our experiments will be on *discourse analysis*, we first need to clarify what discourse analysis actually means and which common ways there are to represent and analyze discourse automatically.

In a nutshell, discourse analysis is an area of research which explores and analyzes language phenomena beyond the sentence level (Stede, 2011). Although the scope of this research can be quite large, ranging from the use of pronouns in a sentence to the logical composition of the whole document, in our work we will primarily concentrate on the coherence structure of a text, *i.e.*, its segmentation into *elementary discourse units* (typically single propositions) and induction of hierarchical *coherence relations* (semantic or pragmatic links) between these EDUs.

Although the idea of splitting the text into smaller meaningful pieces and inferring semantic relationships between these parts is anything but new, dating back to the very origins of general linguistics (Aristotle, 2010) and in particular its structuralism branch (de Saussure and Engler, 1990), an especially big surge of interest in this field happened in the 1970-s with the fundamental works of van Dijk (1972) and van Dijk and Kintsch (1983), who introduced the notion of local and global coherence, defining the former as a set of "rules and conditions for the well-formed concatenation of pairs of sentences in a linearly ordered sequence" and specifying the latter as constraints on the macro-structure of the narrative (see Hoey, 1983). Similar ideas were also proposed by Longacre (1979, 1996), who considered the paragraph as a unit of tagmemic grammar that was composed of multiple sentences according to a predefined set of compositional principles. Almost contemporary with these works, Winter (1977) presented an extensive study of various lexical means that could connect two sentences and grouped these means into two major categories: Matching and Logical Sequence, depending on whether they introduced sentences that were giving more details on the preceding content (Matching) or adding new information to the narrative (Logical Sequence).

The increased interest of traditional linguistics in text-level analysis has rapidly spurred the attention of the broader NLP community. Among the first who stressed the importance of discourse phenomena for automatic generation and understanding of texts was Hobbs (1979), who argued that semantic ties between sentences were one the most important component

for building a coherent discourse. Similarly to Winter, Hobbs also proposed a classification of inter-sentence relations, dividing them into ELABORATION, PARALLEL, and CONTRAST. Albeit this taxonomy was obviously too small to accommodate all possible semantic and pragmatic relationships that could exist between two clauses, this division had laid the foundations for many successful approaches to automatic discourse analysis that appeared in the following decades.

RST. One of the best-known such approaches, *Rhetorical Structure Theory* or *RST*, was presented by Mann and Thompson (1988). Besides revising Hobbs' inventory of discourse relations and expanding it to 23 elements (including new items such as ANTITHESIS, CIRCUMSTANCE, EVIDENCE, and ELABORATION), the authors also grouped all coherence links into nucleus-satellite (hypotactic) and multinuclear (paratactic) ones, depending on whether the arguments of these edges were of different or equal importance to the content of the whole text. Based on this grouping, they formally described each relation as a set of constraints on the *Nucleus* (N), *Satellite* (S), *the N+S combination*, and *the effect* of the whole combination on the reader (R). An excerpt from the original description of the ANTITHESIS relation is given in Example 6.1.1

Example 6.1.1 (Definition of the ANTITHESIS Relation)
Relation Name: ANTITHESIS

Constraints on N: *W has positive regard for the situation presented in N*

Constraints on S: *None*

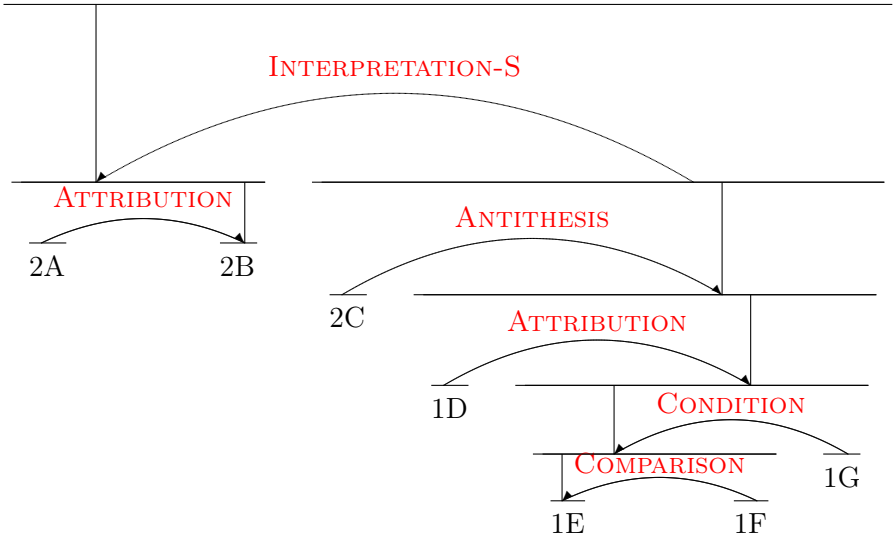
Constraints on the N+S Combination: *the situations presented in N and S are in contrast (i.e., are (a) comprehended as the same in many respects, (b) comprehended as differing in a few respects and (c) compared with respect to one or more of these differences); because of an incompatibility that arises from the contrast, one cannot have positive regard for both the situations presented in N and S; comprehending S and the incompatibility between the situations presented in N and S increases R's positive regard for the situation presented in N*

Effect: *R's positive regard for N is increased*

Locus of the Effect: *N*

The authors then defined the general structure of discourse as a projective (constituency) tree whose nodes were either elementary discourse units or subtrees, which were connected to each other via discourse relations.

You can see an example of such a discourse tree from the original Rhetorical Structure Treebank (Carlson et al., 2001) in Figure 6.1.



[Analysts said,]^{1A} [profit for the dozen or so big drug makers, as a group, is estimated to have climbed between 11% and 14%.]^{1B} [While that's not spectacular,]^{1C} [Neil Sweig, an analyst with Prudential Bache, said]^{1D} [that the rate of growth will "look especially good]^{1E} [as compared to other companies]^{1F} [if the economy turns downward."]^{1G} (WSJ-2341; Carlson et al., 2001)

Figure 6.1: Example of an RST-tree

Despite its immense popularity and practical utility (see Marcu, 1998; Yoshida et al., 2014; Bhatia et al., 2015; Goyal and Eisenstein, 2016), RST has often been criticized for the rigidness of the imposed tree structure (Wolf and Gibson, 2005) and unclear distinction between discourse relations (Nicholas, 1994; Miltsakaki et al., 2004a). As a result of this criticism, two alternative approaches to automatic discourse analysis were proposed in later works.

PDTB. One of these approaches, *PDTB* (named so after the Penn Discourse Treebank [Prasad et al., 2004]), was developed by a research group at University of Pennsylvania (Miltsakaki et al., 2004a,b; Prasad et al., 2008). Instead of fully specifying the hierarchical structure of the whole text and providing an all-embracing set of discourse relations, the authors of this theory mainly focused on the grammatical and lexical means that could connect two sentences (*connectives*) and express a semantic relationship (*sense*) between these predicates. Typical such means are coordinating or subordinating conjunctions (*e.g.*, *and*, *because*, *since*) and discourse adverbials (*e.g.*, *however*, *otherwise*, *as a result*), which can denote a Com-

PARISON, a CONTINGENCY, or some other sense[1] between two sentential arguments (ARG1 and ARG2).

Apart from *explicitly* mentioned connectives, Prasad et al. (2004) also allowed for situations where a connective was missing but could be easily inferred from the text. They called such cases *implicit* discourse relations and demanded the arguments of such structures be determined as well. Furthermore, if there was no connective at all, the authors of PDTB distinguished three different possibilities:

- the coherence relation was either expressed by an alternative lexical means, which made the connective redundant (ALTLEX),
- or it was achieved by referring to the same entities in both arguments (ENTREL),
- or there was no coherence relation at all (NOREL);

and also provided a special ATTRIBUTION label for marking the authors of reported speech.

Example 6.1.2 shows the previous fragment of the Rhetorical Treebank now annotated according to the PDTB scheme.

As we can see from the analysis, PDTB is indeed more flexible than RST, as it allows its discourse units (arguments) to overlap, be disjoint or even embedded into other segments. The assignment of sense relations is also more straightforward and mainly determined by the connectives that link the arguments. But, at the same time, the structure of this annotation is completely flat so that we can neither infer which of the sentences plays a more prominent role nor see the modification scope of other supplementary statements.

Example 6.1.2 (Example of PDTB Analysis)

Analysts said, [*profit for the dozen or so big drug makers, as a group, is estimated to have climbed between 11% and 14%.*]$_{rel1:arg1}$ [IMPLICIT:=*in fact*]$_{rel1:connective}$ [[EXPLICIT:=*While*]$_{rel2:connective}$ [*that's not spectacular*]$_{rel2:arg2}$]$_{rel1:arg2}$; *Neil Sweig, an analyst with Prudential Bache, said* [[[*that the rate of growth will "look especially good as compared to other companies*]$_{rel3:arg1}$ [EXPLICIT: *if*]$_{rel3:connective}$[*the economy turns downward*]$_{rel3:arg2}$]$_{rel2:arg1}$]$_{rel1:arg2}$."

[1] In particular, the authors of PDTB distinguished four major senses (COMPARISON, CONTINGENCY, EXPANSION, and TEMPORAL), and subdivided each of these categories into further subtypes, *e.g.*, COMPARISON included CONCESSION and CONTRAST, whereas CONTINGENCY sense was further divided into CAUSE and CONDITION.

SDRT. Another alternative to RST, *Segmented Discourse Representation Theory* or *SDRT*, was proposed by Lascarides and Asher (2001). Although developed from a completely different angle of view (the authors of SDRT mainly drew their inspiration from predicate logic, dynamic semantics, and anaphora theory), this theory shares many of its features with Rhetorical Structure Theory, as it also assumes a graph-like structure of text and distinguishes between coordinating and subordinating relations. However, unlike RST, Segmented Discourse Representation explicitly allows the text structure to be a multigraph and not only tree (*i.e.*, a discourse node can have multiple parents and can also be connected via multiple links to the same vertex), provided that it does not have crossing dependencies (*i.e.*, does not violate the right-frontier constraint).

We can also notice the relatedness of the two theories by looking at the SDRT analysis of the previous RST fragment in Example 6.2. Although the names of the relations in the presented graph differ from those used in Rhetorical Structure Theory, many of these links have the same (or at least similar) meaning as the respective edges in the first analysis: for example, the SOURCE relation in SDRT almost completely corresponds to the ATTRIBUTION edge in Example 6.1, and the CONTRAST link is similar to the COMPARISON relation defined by Carlson and Marcu (2001).

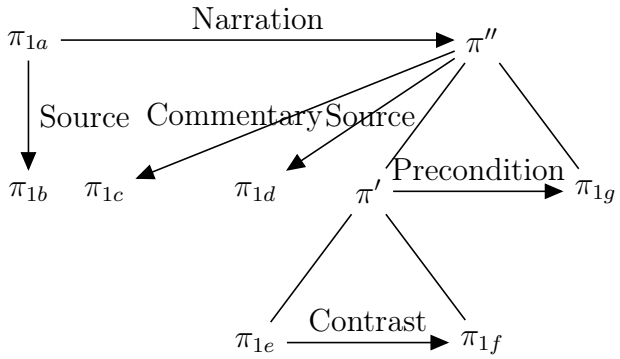


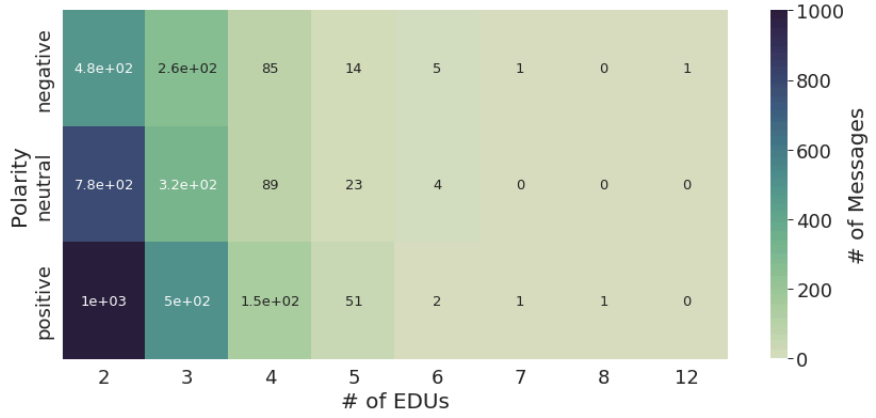
Figure 6.2: Example of an SDRT graph

Final choice. Because it was unclear which of these approaches (RST, PDTB, or SDRT) would be more amenable to our sentiment experiments, we have made our decision by considering the following theoretical and practical aspects: From theoretical perspective, we wanted to have a strictly hierarchical discourse structure for each analyzed tweet so that we could infer the semantic orientation of that message by recursively accumulating polarity scores of its elementary discourse segments. From practical point of view, since there was no discourse parser readily available for German, we wanted to have a maximal assortment of such systems available for English so that we could pick one that would be easiest to retrain on German data. Fortunately, both of these concerns have lead us to the same

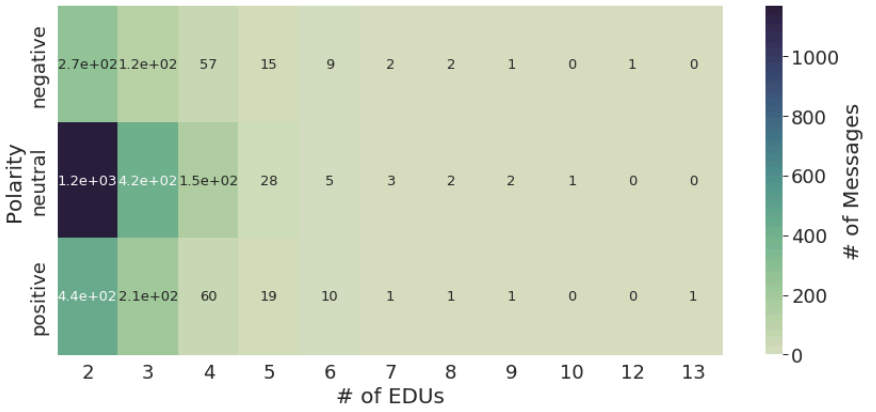
solution—Rhetorical Structure Theory, which was the only formalism that explicitly guaranteed a single root for each analyzed text and also offered a wide variety of open-source parsing systems (*e.g.*, Hernault et al., 2010; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Yoshida et al., 2014; Joty et al., 2015).

6.2 Data Preparation

To prepare the data for our experiments, we split all microblogs from



(a) PotTS



(b) SB10k

Figure 6.3: Distribution of elementary discourse units and polarity classes in the training and development sets of PotTS and SB10k

the PotTS and SB10k corpora into elementary discourse units using the ML-based discourse segmenter of Sidarenka et al. (2015b), which had been previously trained on the Potsdam Commentary Corpus (PCC 2.0; Stede and Neumann, 2014). After filtering out all tweets that had only one EDU,[2] we obtained 4,771 messages (12,137 segments) for PotTS

[2]Since the focus of this chapter is mainly on discourse phenomena, we skip all messages that consist of a single discourse segment, because their overall polarity is unaffected by the discourse structure and can be normally determined with the standard discourse-unaware sentiment techniques.

and 3,763 posts (9,625 segments) for the SB10k corpus. In the next step, we assigned polarity scores to the segments of these microblogs with the help of our lexicon-based attention classifier, analyzing each elementary unit in isolation, independently of the rest of the tweet. We again used the same 70–10–20 split into training, development, and test sets as we did in the previous chapters, considering message-level labels inferred from the annotation of the second expert as gold standard for the PotTS corpus and using provided manual sentiment labels for tweets as reference for the SB10k data.

As we can see from the statistics in Figure 6.3, most tweets that consist of multiple EDUs typically have two or three segments, whereas messages with more than three discourse units are extremely rare. This is also not surprising regarding that the maximum length of a microblog is constrained to 140 characters. Nonetheless, even with this severe length restriction, there still are a few messages that have up to 13 EDUs. Since it was somewhat surprising for us to see that many segments in a single tweet, we decided to have a closer look at these cases. As it turned out, such high number of discourse units typically resulted from spurious punctuation marks, which were carelessly used by Twitter users and evidently confused the segmenter (see Example 6.2.1).

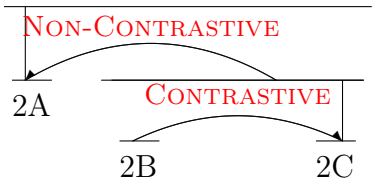
Example 6.2.1 (SB10k Tweet with 13 EDUs)
Tweet: [Guinness on Wheelchairs :]$_1$ [Das .]$_2$ [Ist .]$_3$ [Verdammt .]$_4$ [Noch .]$_5$ [Mal .]$_6$ [Einer .]$_7$ [Der .]$_8$ [Besten .]$_9$ [Werbespots .]$_{10}$ [Des .]$_{11}$ [Jahrzehnts .]$_{12}$ [(Auch ...]$_{13}$
[*Guinness on Wheelchairs :*]$_1$ [*This .*]$_2$ [*Is .*]$_3$ [*Gosh .*]$_4$ [*Darn .*]$_5$ [*It .*]$_6$ [*One .*]$_7$ [*Of .*]$_8$ [*The best .*]$_9$ [*Commercials .*]$_{10}$ [*Of .*]$_{11}$ [*The Decade .*]$_{12}$ [*(Also ...*]$_{13}$

Another noticeable trend that we can see in the data is that the distribution of polar classes in messages with multiple segments largely corresponds to the frequencies of these polarities in the complete datasets: For example, the positive semantic orientation still dominates the PotTS corpus, whereas the neutral polarity constitutes the vast majority of the SB10k set. At the same time, negative microblogs again are the least represented class in both cases and account for only 22% of the former corpus and for 16% of the latter data.

To obtain RST trees for these messages, we retrained the DPLP discourse parser of Ji and Eisenstein (2014) on PCC, after converting all discourse relations to the binary scheme {Contrastive, Non-Contrastive} as suggested by Bhatia et al. (2015).[3] In contrast to the original DPLP implementation though, we did not use Brown clusters (Brown et al., 1992), because this resource was not available for German, nor did we apply the linear projection of the features, because the released parser code was missing this component

[3]See Table 6.3 for more details regarding this mapping.

either. In part due to these modifications, but mostly because of the specifics of the German language (richer morphology, higher lexical variety, and syntactic ambiguity) and a skewed distribution of discourse relation, the results of the retrained model were considerably lower than the figures reported for the English treebank, amounting to 0.777, 0.512, and 0.396 F_1 for span, nuclearity, and relation classification on PCC 2.0 versus corresponding 82.08, 71.13, and 61.63 F_1 on the RST Treebank.[4]



[Mooooiiinn.]^{2A} [Gegen solche Nächte hilft die beste Kur nicht.]^{2B} [Aber Kaffee!]^{2C} (PotTS; Sidarenka, 2016b)
[Hellllooooo!]^{2A} [Even the best cure won't help against such nights.]^{2B} [But coffee!]^{2C}

Figure 6.4: Example of an automatically constructed RST-tree for a Twitter message

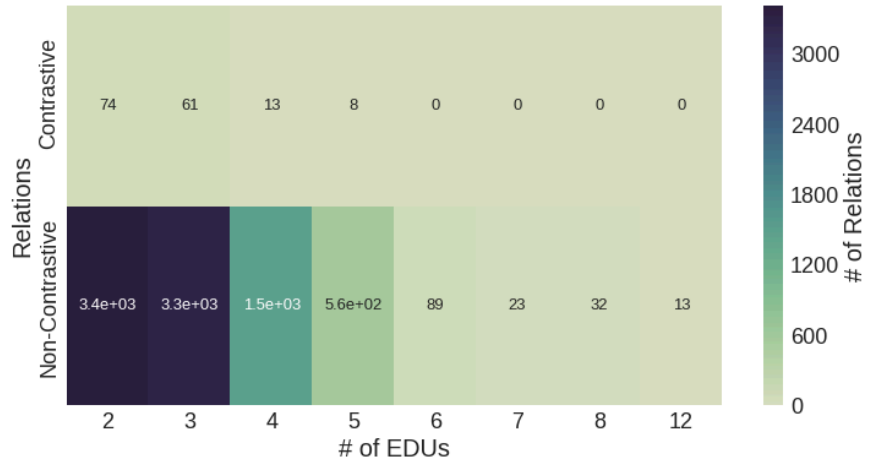
An example of an automatically induced RST tree is shown in Figure 6.4. As we can see from this picture, the adapted parser can correctly distinguish between contrastive and non-contrastive relations in the analyzed tweet (even though it only predicts the former class for two percent of all edges on the PotTS and SB10k data [see Figure 6.5]), but apparently struggles with the disambiguation of the nuclearity status, assigning the highest importance in this example to the initial discourse segment ("Mooooiiinn." [*Hellllooooo!*]), which is merely a greeting, and weighing the second EDU ("Gegen solche Nächte hilft die beste Kur nicht." [*Even the best cure won't help against such nights.*]) less than the third one ("Aber Kaffee!" [*But coffee!*]), although traditional RST would rather consider both units as equally relevant and join them via the multi-nuclear Contrast link.

6.3 Discourse-Aware Sentiment Analysis

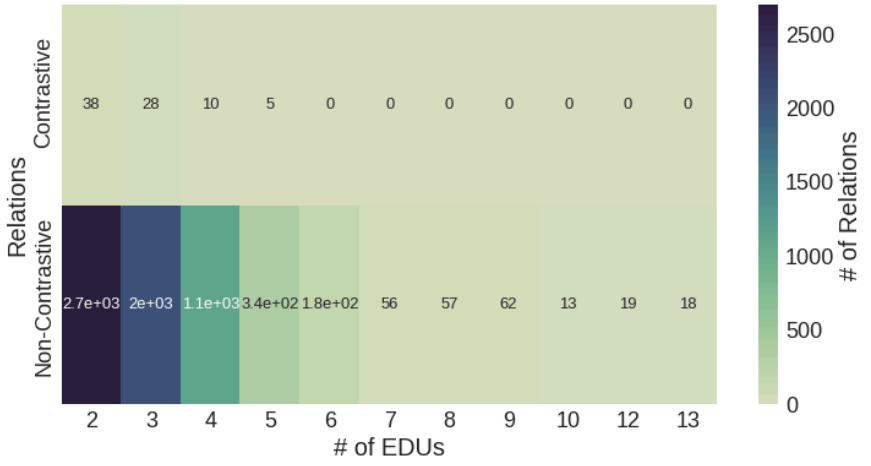
Now before we use these data in our sentiment experiments, let us first revise the most prominent approaches to discourse-aware sentiment analysis that exist in the literature nowadays.

As it turns out, even the very first works on opinion mining already pointed out the importance of discourse phenomena for classification of the overall polarity of a text. For example, in the seminal paper of Pang et al. (2002), where the authors tried to predict the

[4]Following Ji and Eisenstein (2014), we use the span-based evaluation metric of Marcu (2000).



(a) PotTS



(b) SB10k

Figure 6.5: Distribution of discourse relations in the training and development sets of PotTS and SB10k

semantic orientation of movie reviews, they quickly realized the fact that it was insufficient to rely on the mere presence or even the majority of polarity clues in the text, because these clues could any time be reversed by a single counter-argument of the critic (see Example 6.3.1). This observation was also confirmed by Polanyi and Zaenen (2006), who ranked discourse relations among the most important factors that could significantly affect the intensity and polarity of a sentiment. To prove this claim, they gave several convincing examples, where a concessive statement considerably weakened the strength of a polar opinion, and vice versa, an elaboration notably increased its persuasiveness.

Pang and Lee (2004) were also among the first who incorporated a discourse-aware component into a document-level sentiment classifier. For this purpose, they developed a two-stage system in which the first predictor distinguished between subjective and objective statements by constructing a graph of all sentences (linking each sentence to its neighbors and also connecting it to two abstract polarity nodes) and then partitioning this graph into two clusters

(subjective and objective) based on its minimum cut; the second classifier then inferred the overall polarity of the text by only looking at the sentences from the first (subjective) group. With this method, Pang and Lee achieved a statistically significant improvement (86.2% versus 85.2% for the Naïve Bayes system and 86.15% versus 85.45% for SVM) over classifiers that analyzed all text sentences at once, without any filtering.

Example 6.3.1 (Polarity reversal via discourse antithesis)
This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up. (Pang et al., 2002)

Although an oversimplification, the core idea that locally adjacent sentences are likely to share the same subjective orientation (*local coherence*) was dominating the following DASA research for almost a decade. For example, Riloff et al. (2003) also improved the accuracy of their Naïve Bayes predictor of subjective expressions by almost two percent after adding a set of local coherence features. Similarly, Hu and Liu (2004) could better disambiguate users' attitudes to particular product attributes by taking the semantic orientation of previous sentences into account.

At the same time, another line of discourse-aware sentiment research concentrated on the joint classification of all opinions in the text, where in addition to predicting each sentiment in isolation, the authors also sought to maximize the "total happiness" (*global coherence*) of these assignments, ensuring that related subjective statements received agreeing polarity scores. Notable works in this direction were done by Snyder and Barzilay (2007), who proposed the Good Grief algorithm for predicting users' satisfaction with different restaurant aspects, and Somasundaran et al. (2008a,b), who introduced the concept of *opinion frames* (OF), a special data structure for capturing the relations between opinions in discourse. Depending on the type of these opinions (arguing [A] or sentiment [S]), their polarity towards the target (positive [P] or negative [N]), and semantic relationship between these targets (alternative [Alt] or the same [$same$]), the authors distinguished 32 types of possible frames (*SPSPsame*, *SPSNsame*, *APAPalt*, etc.), dividing them into reinforcing and non-reinforcing ones. In later works, Somasundaran et al. (2009b,a) also presented two joint inference frameworks (one based on the iterative classification and another one relying on integer linear programming) for determining the best configuration of all frames in text, achieving 77.72% accuracy on frame prediction in the AMI meeting corpus (Carletta et al., 2005).

An attempt to unite local and global coherence was made by McDonald et al. (2007), who tried to simultaneously predict the polarity of a document and classify semantic orientations of its sentences. For this purpose, the authors devised an undirected probabilistic graphical model based on the structured linear classifier (Collins, 2002). Similarly to Pang and Lee

(2004), they connected the label nodes of each sentence to the labels of its neighboring clauses and also linked these nodes to the overarching vertex representing the polarity of the text. After optimizing this model with the MIRA learning algorithm (Crammer and Singer, 2003), McDonald et al. achieved an accuracy of 82.2% for document-level classification and 62.6% for sentence-level prediction on a corpus of online product reviews, outperforming pure document and sentence classifiers by up to four percent. A crucial limitation of this system though was that its optimization required the gold labels of sentences and documents to be known at the training time, which considerably limited its applicability to other domains with no such data.

Another significant drawback of all previous approaches is that they completely ignored traditional discourse theory and, as a result, severely oversimplified discourse structure. Among the first who tried to overcome this omission were Voll and Taboada (2007), who proposed two discourse-aware enhancements of their lexicon-based sentiment calculator (SO-CAL). In the first method, the authors let the SO-CAL analyze only the topmost nucleus EDU of each sentence, whereas in the second approach, they expanded its input to all clauses that another classifier had considered as relevant to the main topic of the document. Unfortunately, the former solution did not work out as well as expected, yielding 69% accuracy on the corpus of Epinion reviews (Taboada et al., 2006), but the latter system could perform much better, achieving 73% on this two-class prediction task.

Other ways of adding discourse information to a sentiment system were explored by Heerschop et al. (2011), who experimented with three different approaches: (i) increasing the polarity scores of words that appeared near the end of the document, (ii) assigning higher weights to nucleus tokens, and finally (iii) learning separate scores for nuclei and satellites using a genetic algorithm. An evaluation of these methods on the movie review corpus of Pang and Lee (2004) showed better performance of the first option (60.8% accuracy and 0.597 macro-F_1), but the authors could significantly improve the results of the last classifier at the end by adding an offset to the decision boundary of this method, which increased both its accuracy and macro-averaged F_1 to 0.72.

Further notable contributions to RST-based sentiment analysis were made by Zhou et al. (2011), who used a set of heuristic rules to infer polarity shifts of discourse units based on their nuclearity status and outgoing relation links; Zirn et al. (2011), who used a lexicon-based sentiment system to predict the polarity scores of elementary discourse units and then enforced consistency of these assignments over the RST tree with the help of Markov logic constraints; and, finally, Wang and Wu (2013), who determined the semantic orientation of a document by taking a linear combination of the polarity scores of its EDUs and multiplying these scores with automatically learned coefficients.

Among the most recent advances in RST-aware sentiment research, we should especially

emphasize the work of Bhatia et al. (2015), who proposed two different DASA systems:

- discourse-depth reweighting (DDR)
- and rhetorical recursive neural network (R2N2).

In the former approach, the authors estimated the relevance λ_i of each elementary discourse unit i as:

$$\lambda_i = \max\left(0.5, 1 - d_i/6\right),$$

where d_i stands for the depth of the i-th EDU in the document's discourse tree. Afterwards, they computed the sentiment score σ_i of that unit by taking the dot product of its binary feature vector $\mathbf{w}_i$ (token unigrams) with polarity scores $\boldsymbol{\theta}$ of these unigrams:

$$\sigma_i = \boldsymbol{\theta}^\top \mathbf{w}_i;$$

and then calculated the overall semantic orientation of the document Ψ as the sum of sentiment scores for all units, multiplying these scores by their respective discourse-depth factors:

$$\Psi = \sum_i \lambda_i \boldsymbol{\theta}^T \mathbf{w}_i = \boldsymbol{\theta}^T \sum_i \lambda_i \mathbf{w}_i,$$

In the R2N2 system, the authors largely adopted the RNN method of Socher et al. (2013) by recursively computing the polarity scores of discourse units as:

$$\psi_i = \tanh\left(K_n^{(r_i)} \psi_{n(i)} + K_s^{(r_i)} \psi_{s(i)}\right),$$

where $K_n^{(r_i)}$ and $K_s^{(r_i)}$ stand for the nucleus and satellite coefficients associated with the rhetorical relation r_i, and $\psi_{n(i)}$ and $\psi_{s(i)}$ represent sentiment scores of the nucleus and satellite of the i-th vertex. This approach achieved 84.1% two-class accuracy on the movie review corpus of Pang and Lee (2004) and reached 85.6% on the dataset of Socher et al. (2013).

For the sake of completeness, we should also note that there exist discourse-aware sentiment approaches that build upon PDTB and SDRT. For example, Trivedi and Eisenstein (2013) proposed a method based on latent structural SVM (Yu and Joachims, 2009), where they represented each sentence as a vector of features produced by a feature function $\mathbf{f}(y, \mathbf{x}_i, h_i)$, in which $y \in \{-1, +1\}$ denotes the potential polarity of the whole document, $h_i \in \{0, 1\}$ stands for the assumed subjectivity class of sentence i, and $\mathbf{x}_i$ represents the surface form of that sentence; and then tried to infer the most likely semantic orientation of the document $\hat{y}$ over all possible assignments $\mathbf{h}$, *i.e.*,:

$$\hat{y} = \operatorname*{argmax}_y \left(\max_{\mathbf{h}} \mathbf{w}^\top \mathbf{f}(y, \mathbf{x}, \mathbf{h})\right).$$

To ensure that these assignments were still coherent, the authors additionally extended their feature space with special *transitional* attributes, which indicated whether two adjacent

sentences were likely to share the same subjectivity given the discourse connective between them. With the help of these features, Trivedi and Eisenstein could improve the accuracy of the connector-unaware model on the movie review corpus of Maas et al. (2011) from 88.21 to 91.36%.

The first step towards an SDRT-based sentiment approach was made by Asher et al. (2008), who presented an annotation scheme and a pilot corpus of English and French texts that were analyzed according to the SDRT theory and enriched with additional sentiment information. Specifically, the authors asked the annotators to ascribe one of four opinion categories (reporting, judgment, advice, or sentiment) along with their subclasses (*e.g.*, inform, assert, blame, recommend) to each discourse unit that had at least one opinionated word from a sentiment lexicon. Afterwards, they showed that with a simple set of rules, one could easily propagate opinions through SDRT graphs, increasing the strengths or reversing the polarity of the sentiments, depending on the type of the discourse relation that was linking two segments.

In general, however, PDTB- and SDRT-based sentiment systems are much less common than RST-inspired solutions. Because of this fact and due to the reasons described in Section 6.1, we will primarily concentrate on the RST-based of methods. In particular, for the sake of comparison, we replicated the linear combination approach of Wang and Wu (2013) and also reimplemented the DDR and R2N2 systems of Bhatia et al. (2015). Furthermore, to see how these techniques would perform in comparison with much simpler baselines, we additionally created two methods that predicted the polarity of a message by only considering its last or topmost nucleus EDU (henceforth LAST and ROOT), and also estimated the results of our original LBA classifier without any discourse-related modifications (henceforth NO-DISCOURSE).

Apart from the above baselines and existing methods, we propose several novel DASA solutions, which will be briefly described below.

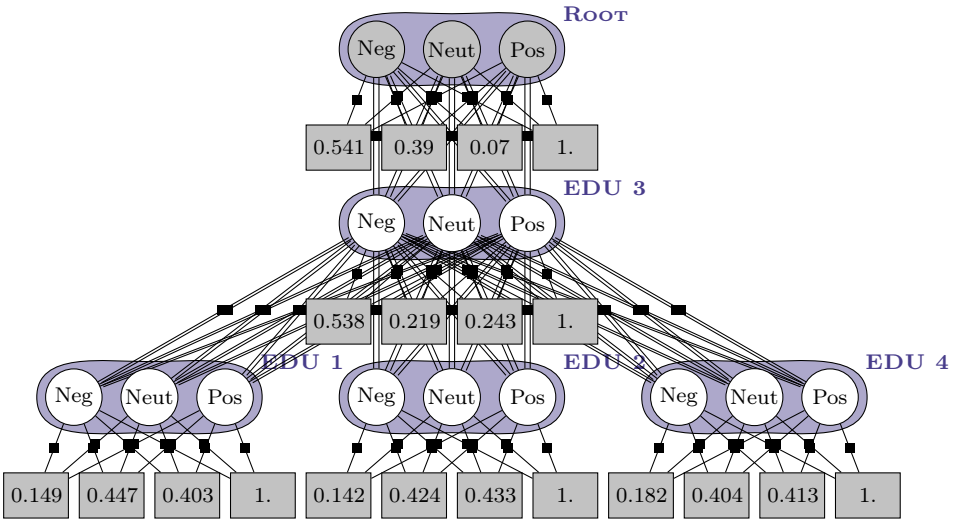
6.3.1 Latent CRF

In the first of these solutions, called *Latent Conditional Random Fields* or *LCRFs*, we consider the problem of message-level sentiment analysis as an inference task over an undirected graphical model, where the nodes of the model represent polarity probabilities of elementary discourse units and the structure of the graph reflects the RST dependency tree of the message.[5] In particular, we define CRF graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a set of vertices $\mathcal{V} = \mathcal{Y} \cup \mathcal{X}$,

[5]Drawing on the work of Bhatia et al. (2015), we obtain this representation using the DEP-DT algorithm of Hirao et al. (2013) with a minor modification that we do not follow any satellite branches while computing the heads of abstract RST nodes in Step 1 of this procedure (see Hirao et al., 2013, pp. 1516–1517).

in which $\mathcal{Y} = \{y_{(i,j)} \mid i \in \{\text{ROOT}, 1, 2, \ldots, T\}, j \in \{\text{NEGATIVE}, \text{NEUTRAL}, \text{POSITIVE}\}\}$ represents (partially observed) random variables (with T standing for the number of EDUs in the tweet), and $\mathcal{X} = \{x_{(i,j)} \mid i \in \{\text{ROOT}, 1, 2, \ldots, T\}, j \in [0, \ldots, 3]\}$ denotes the respective features of these nodes (three polarity scores returned by the LBA classifier plus an additional offset feature whose value is always `1` irrespectively of the input). Since the ROOT vertex, however, does not have a corresponding discourse segment in the RST tree, we use the polarity scores predicted by the LBA classifier for the whole message as features for this node.

Graph edges $\mathcal{E}$ connect random variables to their corresponding features and also link every pair of vertices $(v_{(k,\cdot)}, v_{(i,\cdot)})$ if node k appears as the parent of node i in the RST dependencies.[6] You can see an example of such automatically induced CRF tree in Figure 6.6.



[Gucke Lost]$_1$ [und esse Obst .]$_2$ [Fühlt sich fast an ,]$_3$ [als wäre das mein Leben .]$_4$

[*Watching Lost*]$_1$ [*and eating fruits* .]$_2$ [*Almost feels*]$_3$ [*as if it were my life* .]$_4$

Figure 6.6: Example of an automatically constructed RST-based latent-CRF tree (random variables are shown as circles, fixed input parameters appear as rectangles, and observed values are displayed in gray)

Now before we describe the training of our model, let us briefly recall that in the standard CRF optimization we typically try to find optimal parameters $\boldsymbol{\theta}^*$ that maximize the log-likelihood of all label sequences $\mathbf{y}^{(i)}$ on the training set $\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)\right\}_{i=1}^{N}$, *i.e.*:

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log\left(p\left(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta}\right)\right),$$

where the conditional likelihood is normally estimated as:

$$p\left(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta}\right) = \frac{\exp\left(\sum_{t=1}^{T_i} \sum_k \boldsymbol{\theta}_k \mathbf{f}_k\left(\mathbf{x}_t^{(i)}, \mathbf{y}_{t-t}^{(i)}, \mathbf{y}_t^{(i)}\right)\right)}{Z}.$$

[6] In fact, we use two edges to connect each child to its parent: one for the CONTRASTIVE relation and another one for the NON-CONTRASTIVE link.

Adapting this equation to our RST-based CRF structures, we obtain:

$$p\left(\mathbf{y}^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}\right) = \frac{\exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t^{(i)}, \mathbf{y}_c^{(i)}\right)\right]\right)}{Z}, \quad (6.1)$$

where $ch(t)$ denotes the children of node t, v stands for the indices of node features, and e represents the indices of edge attributes.

A crucial problem with this formulation though is that in our task, only a small subset of labels from $\mathbf{y}^{(i)}$ (namely those of the root node) are actually observed at the training time, whereas the rest of the tags (those which pertain to EDUs) are unknown. We will denote these observed and hidden subsets as $\mathbf{y}_o^{(i)}$ and $\mathbf{y}_h^{(i)}$ respectively. Using this notation, we can redefine the training objective of our model as finding such parameters $\boldsymbol{\theta}^*$ that maximize the log-likelihood of *observed* labels, *i.e.*:

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log\left(p\left(\mathbf{y}_o^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}\right)\right).$$

With this formulation, however, it is still unclear what we should do with hidden tags $\mathbf{y}_h^{(i)}$, because the values of their features remain undefined.

One possible way to approach the problem of unobserved states in the input is to assume that any label sequence $\mathbf{y}_h^{(i)}$ might be true, and then try to maximize the parameters along the path that leads to the maximum probability of the correct observed tag, *i.e.*:

$$\begin{aligned} \mathbf{y}^{(i)} &= [\mathbf{y}_o^{(i)}, \mathbf{y}_h^{*(i)}], \text{ where} \\ \mathbf{y}_h^{*(i)} &= \operatorname*{argmax}_{\mathbf{y}_h^{(i)}} p\left(\mathbf{y}_o^{(i)}|\mathbf{x}^{(i)}\right), \end{aligned} \quad (6.2)$$

and which we can easily find using standard Viterbi decoding.

Unfortunately, if we simply consider label sequence $\mathbf{y}^{(i)}$ from Equation 6.2 as the ground truth and penalize all labels that disagree with this sequence, we might overly commit ourselves to the model's guess of unknown tags and unduly discriminate against other possible hidden label assignments. To mitigate this effect, we can instead penalize only one other sequence, namely the one that maximizes the probability of an incorrect label at the observed state:

$$\begin{aligned} \mathbf{y}'^{(i)} &= \operatorname*{argmax}_{\mathbf{y}_o'^{(i)} \neq \mathbf{y}_o^{(i)}} p\left([\mathbf{y}_o'^{(i)}, \mathbf{y}_h^{*(i)}]|\mathbf{x}^{(i)}\right), \text{ where} \\ \mathbf{y}_h^{*(i)} &= \operatorname*{argmax}_{\mathbf{y}_h^{(i)}} p\left(\mathbf{y}_o'^{(i)}|\mathbf{x}^{(i)}\right). \end{aligned}$$

Correspondingly, we reformulate our objective and instead of maximizing the log-likelihood of the training set will now maximize the difference between the log-probabilities of the

correct and most likely wrong assignments:

$$\begin{aligned}\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \log\left(p\left(\mathbf{y}^{(i)}\right)\right) - \log\left(p\left(\mathbf{y}^{'(i)}\right)\right) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \log\left(\exp\left(\boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\right)\right) - \log\left(\exp\left(\boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{'(i)})\right)\right) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \boldsymbol{\theta}^\top \left(\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{'(i)})\right),\end{aligned} \tag{6.3}$$

where $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ and $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{'(i)})$ mean all features associated with label sequences $\mathbf{y}^{(i)}$ and $\mathbf{y}^{'(i)}$ respectively.

The only thing that we now need to do to the above objective is to introduce a regularization term $\frac{1}{2}\|\boldsymbol{\theta}\|^2$ in order to prevent its divergence to infinity in the cases when $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ and $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{'(i)})$ are perfectly separable. This brings us to the final formulation:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\theta}\|^2 - \sum_{i=1}^{N} \boldsymbol{\theta}^\top \left(\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{'(i)})\right) \tag{6.4}$$

At this point, we can notice that the resulting function is identical to the unconstrained minimization problem of structural SVM (Taskar et al., 2003), and we indeed can piggyback on one of the many efficient SVM-optimization techniques to learn the parameters of our model. In particular, we use the block-coordinate Frank-Wolfe algorithm (Lacoste-Julien et al., 2013), running it for 1,000 epochs or until convergence, whichever of these events occurs first.

6.3.2 Latent-Marginalized CRF

Another way to tackle unobserved labels is to estimate the probability of observed tags by marginalizing (summing) out hidden variables from the joint distribution, *i.e.*:

$$p\left(\mathbf{Y}_o{=}\mathbf{y}_o\right) = \sum_{\mathbf{y}_h} p\left(\mathbf{Y}_o{=}\mathbf{y}_o, \mathbf{Y}_h{=}\mathbf{y}_h\right).$$

Applying this formula to Equation 6.1, we get:

$$\begin{aligned}p\left(\mathbf{y}_o^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}\right) &= \sum_{\mathbf{y}_h^{(i)}} p\left([\mathbf{y}_o^{(i)}, \mathbf{y}_h^{(i)}]|\mathbf{x}^{(i)};\boldsymbol{\theta}\right) \\ &= \frac{\sum_{\mathbf{y}_h^{(i)}} \exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t^{(i)}, \mathbf{y}_c^{(i)}\right)\right]\right)}{Z},\end{aligned}$$

where $\mathbf{y}^{(i)}$ in the numerator is defined as before: $\mathbf{y}^{(i)} = [\mathbf{y}_o^{(i)}, \mathbf{y}_h^{(i)}]$.

This time again, we would like to maximize the probability of the correct assignment, setting it apart from its closest competitor by some margin. Unfortunately, due to the

summation over all $\mathbf{y}_h^{(i)}$, we cannot avail ourselves of the log-exp cancellation trick, which we used previously in Equation 6.3. Instead of this, we replace the difference of the log-likelihoods by the ratio of marginal probabilities:

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{N} \frac{p(\mathbf{y}^{(i)})}{p(\mathbf{y}'^{(i)})} \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{N} \frac{\sum_{\mathbf{y}_h^{(i)}} \exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t^{(i)}, \mathbf{y}_c^{(i)}\right)\right]\right)}{\sum_{\mathbf{y}_h^{(i)}} \exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t'^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t'^{(i)}, \mathbf{y}_c'^{(i)}\right)\right]\right)}\end{aligned} \tag{6.5}$$

To simplify this expression, we can introduce the following abbreviations:

$$\begin{aligned}a &:= \exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t^{(i)}, \mathbf{y}_c^{(i)}\right)\right]\right), \\ b &:= \exp\left(\sum_{t=0}^{T_i}\left[\sum_v \boldsymbol{\theta}_v \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t'^{(i)}\right) + \sum_{c\in ch(t)}\sum_e \boldsymbol{\theta}_e \mathbf{f}_e\left(\mathbf{y}_t'^{(i)}, \mathbf{y}_c'^{(i)}\right)\right]\right).\end{aligned}$$

Now we estimate the derivatives of functions a and b w.r.t. a single parameter $\boldsymbol{\theta}_v$ as:

$$\begin{aligned}\frac{\partial a}{\partial \boldsymbol{\theta}_v} &= a\sum_{t=0}^{T_i} \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) \propto \mathbb{E}_{\mathbf{y}^{(i)}}\left[\mathbf{f}_v\right], \\ \frac{\partial b}{\partial \boldsymbol{\theta}_v} &= b\sum_{t=0}^{T_i} \mathbf{f}_v\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)}\right) \propto \mathbb{E}_{\mathbf{y}'^{(i)}}\left[\mathbf{f}_v\right];\end{aligned}$$

and analogously obtain:

$$\begin{aligned}\frac{\partial a}{\partial \boldsymbol{\theta}_e} &= a\sum_{t=0}^{T_i}\sum_{c\in ch(t)} \mathbf{f}_e\left(\mathbf{y}_t^{(i)}, \mathbf{y}_c^{(i)}\right) \propto \mathbb{E}_{\mathbf{y}^{(i)}}\left[\mathbf{f}_e\right], \\ \frac{\partial b}{\partial \boldsymbol{\theta}_e} &= b\sum_{t=0}^{T_i}\sum_{c\in ch(t)} \mathbf{f}_e\left(\mathbf{y}_t'^{(i)}, \mathbf{y}_c'^{(i)}\right) \propto \mathbb{E}_{\mathbf{y}'^{(i)}}\left[\mathbf{f}_e\right].\end{aligned}$$

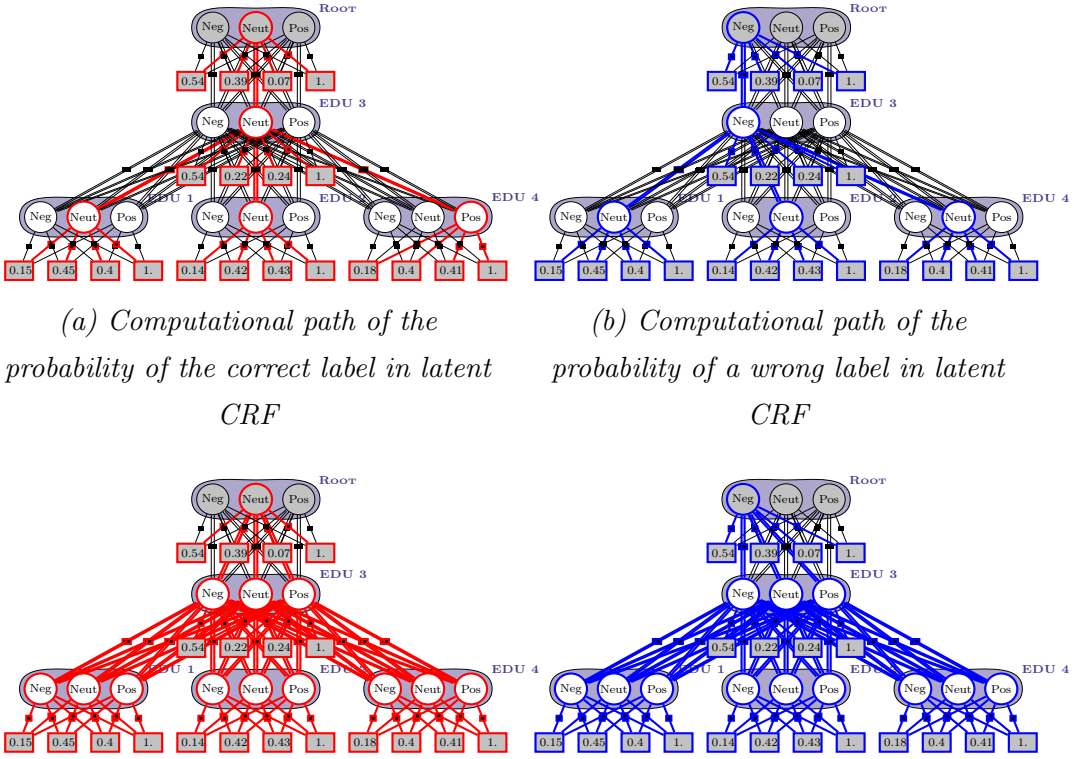
With the help of these expressions, we can easily compute the gradient of the objective function w.r.t. $\boldsymbol{\theta}$ by observing that:

$$\nabla_{\boldsymbol{\theta}} = \sum_{i=1}^{N} \frac{\sum_{\mathbf{y}_h^{(i)}} \nabla_{\boldsymbol{\theta}} a \sum_{\mathbf{y}_h^{(i)}} b - \sum_{\mathbf{y}_h^{(i)}} a \sum_{\mathbf{y}_h^{(i)}} \nabla_{\boldsymbol{\theta}} b}{\left(\sum_{\mathbf{y}_h^{(i)}} b\right)^2}. \tag{6.6}$$

We again use the block-coordinate Frank-Wolfe algorithm to optimize the parameters of our model, but instead of pushing these parameters in the direction $\boldsymbol{\psi} = \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}'^{(i)})$ (which is the derivative of latent CRFs, see Algorithm 2 in [Lacoste-Julien et al., 2013]), we now maximize them along the gradient from Equation 6.6.

It is probably easier to realize the difference between the two CRF methods (latent and latent-marginalized CRFs) more vividly by looking at Figure 6.7, in which we highlighted

the paths that are used to compute the probabilities of correct and wrong labels in both systems. As we can see from this picture, LCRF only considers one label sequence that leads to the maximum probability of the correct tag (NEUT) at the single observed ROOT node and then compares this sequence with the path that maximizes the probability of an incorrect tag (in this case NEG) at the same node. In contrast to this, LMCRF considers all possible label configurations of elementary discourse units and uses this total cumulative mass to estimate the probability of both (correct and wrong) observed tags.



(a) Computational path of the probability of the correct label in latent CRF

(b) Computational path of the probability of a wrong label in latent CRF

(c) Computational path of the probability of the correct label in latent-marginalized CRF

(d) Computational path of the probability of a wrong label in latent-marginalized CRF

Figure 6.7: Confronted computational paths in latent and latent-marginalized conditional random fields

6.3.3 Recursive Dirichlet Process

Finally, the last method that we present in this chapter, *Recursive Dirichlet Process* or *RDP*, goes a further step in the probabilistic direction by assuming that not only the probabilities of discourse units but also the parameters via which these probabilities are computed represent random variables.

In particular, we associate a variable $\mathbf{z}_j \in \mathbb{R}^3_+$, s.t. $\|\mathbf{z}\|_1 = 1$, with every RST node j (which in this case can be either an elementary discourse segment or an abstract span).[7] This

[7] In contrast to the previous CRF approaches, this time, we depart from the dependency tree represen-

variable specifies the multivariate probability of the three polarities (NEGATIVE, NEUTRAL, and POSITIVE) for the j-th node. Since every element of $\mathbf{z}_j$ has to be non-negative and their total sum must add up to one, it is natural to assume that the value of this variable is drawn from a Dirichlet distribution:

$$\mathbf{z}_j \sim Dir(\boldsymbol{\alpha}).$$

The only parameter accepted by this distribution, which simultaneously controls both the mean and the variance of its outcomes, is vector $\boldsymbol{\alpha}$. Consequently, our primary goal in this method is to find a way how to compute this parameter automatically for each node.

An obvious starting point for this computation is the polarity scores predicted by the base classifier for every elementary discourse unit, which we will henceforth denote as $\boldsymbol{z}_{j_0} \in \mathbb{R}^3_+$. Since these scores, however, are only available for elementary segments, we initialize the corresponding variables of the abstract spans to zeroes with the only exception being the root node, to which we again assign the scores returned by the LBA classifier for the whole message.

To compute the posterior distribution of the root ($\mathbf{z}_{\text{ROOT}}$), we sort all nodes of the RST tree in reverse topological order and estimate the polarities of the spans from the bottom up by joining the $\mathbf{z}$-scores of their children. But before we do this joining, we multiply the $\mathbf{z}$-vector of each child k with a special matrix M_r, where $r \in \{\{\text{NUCLEUS}, \text{SATELLITE}\} \times \{\text{CONTRASTIVE}, \text{NON-CONTRASTIVE}\}\}$ is the discourse relation holding between that child and its parent, and project the result of this multiplication back to the probability simplex using the sparsemax operation (Martins and Astudillo, 2016):

$$\mathbf{z}^*_k = sparsemax\left(M_r \mathbf{z}_k^\top\right). \tag{6.7}$$

The main goal of matrix M_r is to reflect contextual polarity changes that might be conveyed by discourse relations: for example, a contrastive link might stronger affect the polarity of the parent than a non-contrastive one (compare, for instance, the contrastive *Many people support Trump, but he behaves like an alpha male* with the non-contrastive *Many people support Trump, because he behaves like an alpha male*). Because this parameter also represents a random variable, we sample it from a multivariate normal distribution:

$$M_r \sim \mathcal{N}_{3\times 3}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r),$$

tation and adopt the discourse tree structure proposed by Bhatia et al. (2015) for their R2N2 method. In this structure, we keep all abstract nodes from the original RST tree, but relink all satellites to the abstract parents of their nuclei.

setting the mean of this distribution to:[8]

$$\boldsymbol{\mu}_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and initializing its covariance matrix to all ones:

$$\boldsymbol{\Sigma}_r = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

With this choice of parameters, we hope to dampen the effect of neutral EDUs[9] in order to prevent situations where multiple objective segments vanquish the meaning of a single polar discourse unit.

Afterwards, when seeing the k-th child of the j-th node in the RST tree, we compute the $\boldsymbol{\alpha}$ parameter of this node as follows:

$$\boldsymbol{\alpha}_{j_k} = \boldsymbol{\beta} \odot \mathbf{z}^*_k + (\mathbf{1} - \boldsymbol{\beta}) \odot \mathbf{z}_{j_{k-1}}, \tag{6.8}$$

where $\boldsymbol{\beta} \in \mathbb{R}^3$ is another multivariate random variable sampled from the Beta distribution $B(5., 5.)$, which controls the amount of information we want to pass from child to its parent; $\mathbf{z}_{j_{k-1}}$ is the value of the $\mathbf{z}$-vector for the j-th node after seeing its previous ($k-1$-th) child; and $\odot$ means elementwise multiplication.

The only thing that we now need to do to the above $\boldsymbol{\alpha}_{j_k}$ term before drawing the actual probability $\boldsymbol{z}_{j_k}$ is to scale this vector by a certain amount in order to reduce the variance of the resulting Dirichlet distribution.[10] In particular, we compute this scaling factor as follows:

$$scale = \frac{\xi \times \left(0.1 + \cos\left(\mathbf{z}^*_k, \mathbf{z}_{j_{k-1}}\right)\right)}{H\left(\boldsymbol{\alpha}_{j_k}\right)};$$

where ξ is a model parameter sampled from a χ^2-distribution: $\xi \sim \chi^2(34)$; 0.1 is a constant used to prevent zero scales in the cases when $\cos\left(\mathbf{z}^*_k, \mathbf{z}_{j_{k-1}}\right)$ is zero; and $H\left(\boldsymbol{\alpha}_{j_k}\right)$ stands for the entropy of the $\boldsymbol{\alpha}_{j_k}$ vector. Although this expression looks somewhat complicated, the intuition behind it is very simple: The ξ term encodes our prior belief in the correctness of model's prediction (the higher its value, the more we trust the model); the cosine measures

[8] Before we do the actual sampling, we unroll this parameter to a vector and then reshape the sampled value back to a 3×3 matrix.

[9] As you can see from Equation 6.7, the middle row of the M_r matrix is responsible for propagating the neutral score of the j-the node, and by setting this row to $[0, 0.3, 0]$ we effectively reduce the neutral polarity by two thirds.

[10] Because if we keep the $\boldsymbol{\alpha}_{j_k}$ vector from Equation 6.8 unchanged, most of its values will be in the range $[0, \ldots, 1]$ which will lead to an extremely high variance of the Dirichlet distribution.

the similarity between the probabilities of parent and child (the more similar these probabilities, the greater will be the scale); and, finally, the entropy in the denominator tells us how uniform the vector $\boldsymbol{\alpha}_{j_k}$ is (the more equal its scores, the less confident we will be in the final outcome).

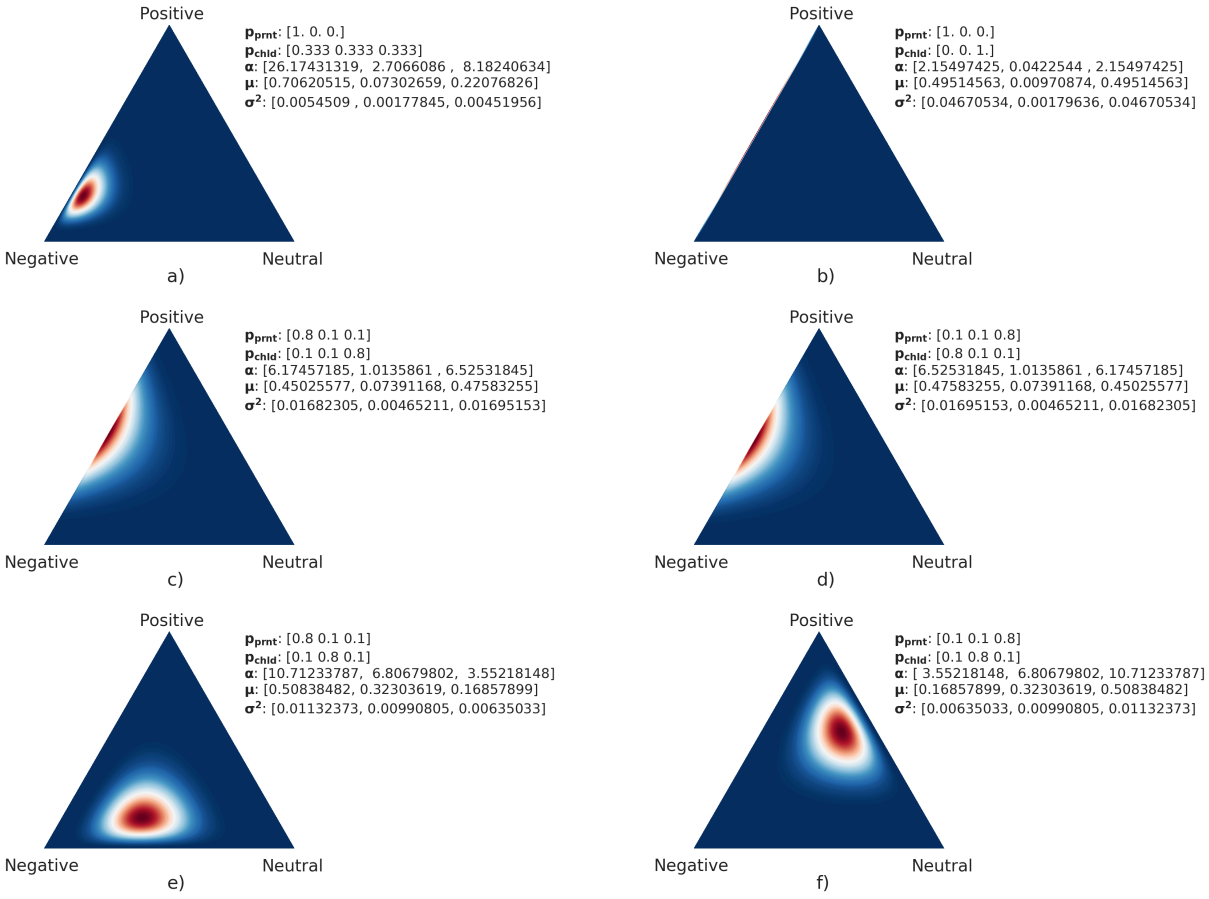


Figure 6.8: Probability distributions of polar classes computed by the Recursive Dirichlet Process (higher probability regions are highlighted in red; $\mathbf{p}_{prnt}$ means the probability of the parent node [the values in the vector represent the scores for the negative, neutral, and positive polarities respectively]; $\mathbf{p}_{chld}$ denotes the probability of the child; and $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}^2$ represent the parameters of the resulting joint distribution shown in the simplices)

With the *scale* and $\boldsymbol{\alpha}_{j_k}$ terms at hand, we are all set to compute the updated probability of polar classes for the j-th node after considering its k-th child:

$$\mathbf{z}_{j_k} \sim Dir(scale \times \boldsymbol{\alpha}_{j_k}).$$

You can see some examples of this computation in Figure 6.8, where we plotted different

configurations of parent and child probabilities ($\mathbf{z}_{j_{k-1}}$ and $\mathbf{z}_k$, shown to the right of each picture) and the resulting Dirichlet distributions (represented as simplices). For instance, in the top-left figure, we show a situation where the parent has a very strong probability of the negative class ($[1, 0, 0]$), but the probability of the child is absolutely uniform ($[0.33, 0.33, 0.33]$); in this case, the model keeps to the negative polarity, heaping almost all probability mass in this corner. At the same, to account for the uncertainty about the child, RDP slightly moves the crest of the probability hill (*i.e.*, its mean) towards the positive class and makes the slopes of this hill lower along all three axes (*i.e.*, increases its variance). On the other hand, if parent and child have completely opposite semantic orientations (say POSITIVE and NEGATIVE), which the base classifier is perfectly sure about, as shown in Subfigure b, RDP uniformly distributes the whole probability just along the POSITIVE–NEGATIVE edge. Another situation is depicted in the middle row, where parent and child again have opposite polarities, but the base predictor is less sure about its decisions and also admits a small chance that either of these nodes is neutral. In this case, RDP still spreads most probability along the main polar edge, but places the mean of this distribution right in-between the two polar corners and also screeds some part of that mass towards the center of the simplex. Finally, in the last row, we can see our intended discrimination of the neutral orientation: This time, the parent node is strictly polar (negative on the left and positive on the right), whereas its child is neutral. In contrast to the previous two examples, the mean of the resulting distribution is located closer to the polar corner and not in-between the two juxtaposed classes as before.

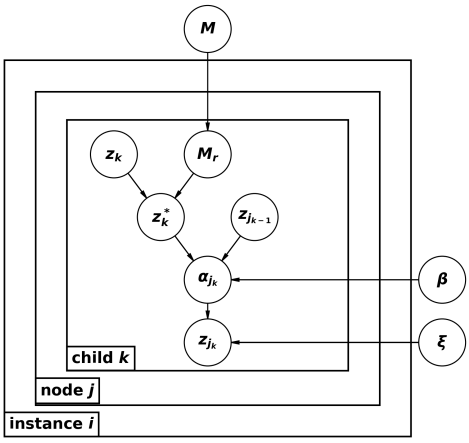


Figure 6.9: A plate diagram of the Recursive Dirichlet Process (without the final categorical draw)

Returning back to our model, after processing all K children of the j-th node, we regard the last outcome $\mathbf{z}_{j_K}$ as the final polarity distribution of that node and use this value to estimate the probabilities of the remaining ancestors in the RST tree. Finally, after finishing processing all descendants of the root, we use the resulting $\mathbf{z}_{\text{ROOT}_K}$ vector as a parameter of

a categorical distribution from which we draw the final prediction label y:

$$y \sim Cat(\mathbf{z}_{\text{ROOT}}).$$

Using this manually defined model as a starting point, we can estimate our prior belief in the joint probability of hidden and observed variables $p(y, \mathbf{z})$. As it turns out, knowing this belief is enough to derive another probability $q(\mathbf{z})$, which best approximates the distribution of only the latent nodes. In particular, we define the structure of $q(\mathbf{z})$ to be the same as in $p(y, \mathbf{z})$, but deprive it of the last step (drawing of the observed label) and optimize the parameters $\boldsymbol{\theta}$ of this model ($\boldsymbol{\mu}_r$, $\boldsymbol{\Sigma}_r$, and the parameters of the Beta and χ^2 distributions) by maximizing the evidence lower bound between p and q, using stochastic gradient descent(see Ranganath et al., 2014):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z})} \left[\log \left(p(y, \mathbf{z}) \right) - \log \left(q(\mathbf{z}) \right) \right].$$

We perform this optimization for 100 epochs, picking the parameters that yield the best macro-averaged F_1-score on the set-aside development data.

The results of our proposed and baseline methods are shown in Table 6.1.

As we can see from the table, our approaches perform fairly well in comparison with other systems, outperforming them in terms of macro- and macro-averaged F_1 on both datasets. Especially the latent-marginalized CRF shows fairly strong scores, yielding the best F_1-results for the positive and neutral classes on the PotTS and SB10k data, which in turn leads to the highest overall micro-averaged F_1-measure on these corpora. This solution is closely followed by the Recursive Dirichlet Process, whose F_1 for the positive class on the PotTS test set is identical to that attained by LMCRF and the F-score for the negative class is even one percent higher, which allows it to reach the best macro-average on this test set.

As it turns out, the strongest competitors to our systems are the NO-DISCOURSE approach and the R2N2 method by Bhatia et al. (2015). The former solution outperforms the latter on the PotTS corpus on both metrics (macro- and micro-F_1), but falls against it with respect to the macro-F_1 on the SB10k set. The DDR and WNG methods get sixth and seventh places respectively, followed by the simplest solutions, LAST and ROOT. Interestingly enough, the LAST approach beats the ROOT method on the SB10k data, but shows worse scores on the PotTS corpus, which is mostly due to the lower recall of the negative class.

6.3.4 Error Analysis

Although our methods performed quite competitive, we decided to still look at their remaining errors in order to understand the reasons for their potential weaknesses.

Method	Positive			Negative			Neutral			Macro F_1	Micro F_1
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1		
PotTS											
LCRF	0.76	0.79	**0.77**	**0.61**	0.53	0.56	0.7	0.71	0.71	0.67	0.709
LMCRF	**0.77**	0.77	**0.77**	**0.61**	0.54	0.57	0.69	**0.74**	**0.72**	0.671	**0.712**
RDP	0.73	0.82	**0.77**	**0.61**	0.56	**0.58**	**0.73**	0.65	0.69	**0.678**	0.706
DDR	0.73	0.77	0.75	0.54	**0.59**	0.56	0.69	0.61	0.65	0.655	0.674
R2N2	0.74	0.78	0.76	0.59	0.53	0.56	0.68	0.68	0.68	0.657	0.692
WNG	0.58	0.79	0.67	**0.61**	0.21	0.31	0.61	0.57	0.59	0.487	0.59
Last	0.52	**0.83**	0.64	0.57	0.17	0.26	0.61	0.43	0.5	0.453	0.549
Root	0.56	0.73	0.64	0.58	0.22	0.32	0.55	0.54	0.54	0.481	0.56
No-Discourse	0.73	0.82	**0.77**	**0.61**	0.56	**0.58**	0.72	0.66	0.69	0.677	0.706
SB10k											
LCRF	**0.64**	**0.69**	0.66	0.45	**0.45**	0.45	**0.82**	0.79	**0.8**	0.557	0.713
LMCRF	**0.64**	**0.69**	**0.67**	0.45	**0.45**	0.45	**0.82**	0.79	**0.8**	**0.56**	**0.715**
RDP	**0.64**	**0.69**	0.66	0.45	**0.45**	0.45	0.82	0.79	**0.8**	0.557	0.713
DDR	0.59	0.63	0.61	**0.48**	0.44	**0.46**	0.77	0.76	0.77	0.534	0.681
R2N2	**0.64**	**0.69**	0.66	0.46	**0.45**	0.45	0.81	0.79	**0.8**	0.559	0.713
WNG	0.61	0.63	0.62	0.46	0.29	0.36	0.76	**0.82**	0.79	0.488	0.693
Last	0.56	0.55	0.56	0.46	0.29	0.36	0.73	0.8	0.76	0.459	0.661
Root	0.51	0.55	0.53	0.4	0.3	0.35	0.74	0.76	0.75	0.438	0.64
No-Discourse	**0.64**	**0.69**	0.66	0.45	**0.45**	0.45	**0.82**	0.79	**0.8**	0.557	0.713

Table 6.1: Results of discourse-aware sentiment analysis methods
LCRF – latent conditional random fields, LMCRF – latent-marginalized conditional random fields, RDP – recursive Dirichlet process, DDR – discourse-depth reweighting (Bhatia et al., 2015), R2N2 – rhetorical recursive neural network (Bhatia et al., 2015), WNG – Wang and Wu (2013), Last *– polarity determined by the last EDU,* Root *– polarity determined by the root EDU(s),* No-Discourse *– discourse-unaware classifier*

The first such error shown in Example 6.3.2 was made by the latent CRF system, which erroneously considered a negative tweet as neutral. But as we can see from the picture of the automatic RST tree in this example, we can hardly expect the right decision in this case anyway, because neither EDUs nor the root node of this message were correctly classified as negative by the LBA classifier. Nevertheless, even in this apparently hopeless situation, messages propagated from leaves to the root during the max-product inference still tell the latter node that the predicted class better be negative. (We inspected the belief propagation messages passed in the forward direction and found that the total score for the negative class amounts to 0.597, whereas the belief in the positive class [its closest rival] only runs up to 0.462.) Unfortunately, these messages cannot outweigh the high score of the neutral class that results from the node features (the state score for this polarity is equal to 0.524, whereas the negative class only obtains a score of -0.118).[11]

[11] All scores for this example are given in the logarithm domain.

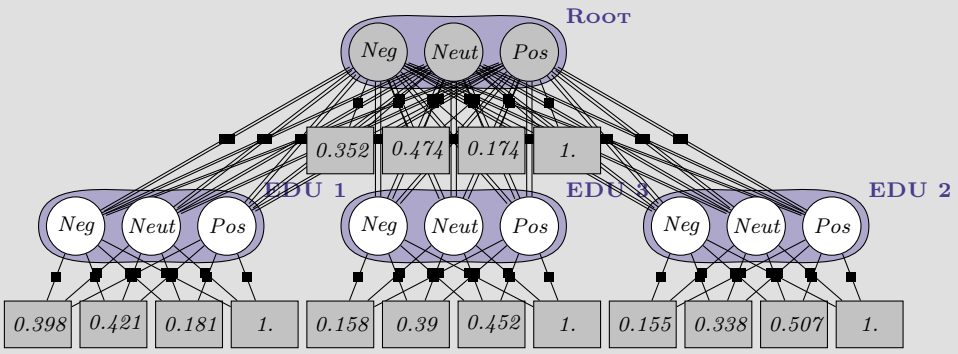
Example 6.3.2 (An Error Made by the LCRF System)

Tweet: [Boah , also doch wieder ein Mann , oder ?]$_1$ [ODER ?]$_2$ [papst]$_3$

[*Boah, a man again, isn't it ?*]$_1$ [*ISN'T ?*]$_2$ [*pope*]$_3$

Gold Label: **negative**

Predicted Label: **neutral***



As it turns out, high neutral node scores of the root are also the main reason for the misclassification in Example 6.3.3, where the LMCRF system also confuses the negative polarity with the neutral class. This time, however, messages coming from the leaves suggest almost equal probabilities for both semantic orientations, so that feature scores of the root completely call the shots in the final decision.

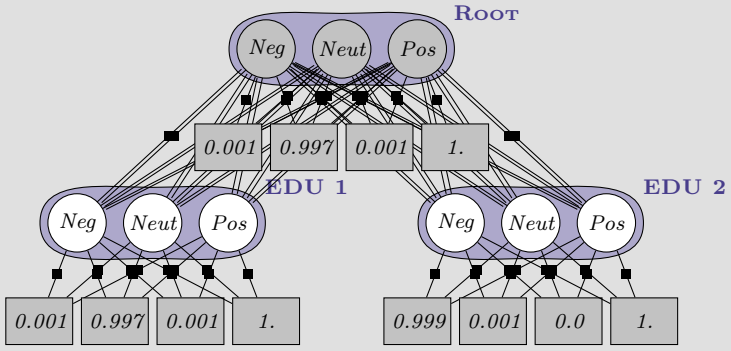
Example 6.3.3 (An Error Made by the LMCRF System)

Tweet: ' [Wissen ?]$_1$ [Igitt geh weg damit !]$_2$

[*Knowledge ?*]$_1$ [*Yuck , go away with it*]$_2$

Gold Label: **negative**

Predicted Label: **neutral***



Unfortunately, the recursive Dirichlet process cannot withstand the erroneous predictions of the base classifier either. For instance, in Example 6.3.4, LBA assigns the highest scores to the positive class in three out of four EDUs, even though each of these units by itself expresses a negative attitude of the author. Alas, the only case where the base classifier correctly predicts the negative label ("Das is noch lange nicht ausdiskutiert !" [*It's no way*

been talked out !]) drowns at the very beginning of the score propagation. (As it turned out, the learned $\boldsymbol{\beta}$ parameter, which controls the amount of information passed from child to its parent in Equation 6.8, is extremely low for the negative class, amounting to only 0.097, whereas for the positive and negative polarities it runs up to 0.212 and 0.279. Due to this low value, only one tenth of the negative score from the third EDU arrives at the parent when the model computes the polarity scores of the abstract span 2.)

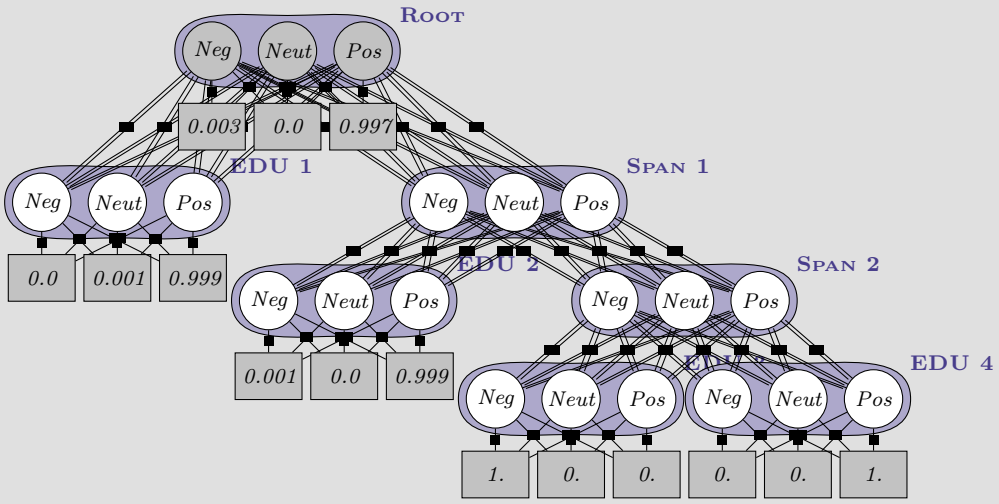
Example 6.3.4 (An Error Made by the RDP System)

Tweet: [Prima , was sind das für Idioten im DFB ?]$_1$ [Das is eine Muppetsshow auf LSD !]$_2$ [Das is noch lange nicht ausdiskutiert !]$_3$ [Kiessling ist ein Depp !]$_4$

[*Great, who are these idiots in the DFB ?*]$_1$ [*It is a muppet show on LSD*]$_2$ [*It's no way been talked out !*]$_3$ [*Kiessling is a goof !*]$_4$

Gold Label: **negative**

Predicted Label: **positive***



Another interesting error shown in Example 6.3.5 was made by the baseline ROOT system, which similarly to CRF-based approaches confused the negative class with the neutral polarity. This time, however, the misclassification is due to the discourse structure itself rather than wrong predictions of the underlying sentiment method. Because LBA correctly recognizes that the negative smiley at the end of tweet has a strictly negative semantic orientation, but the discourse-aware baseline does not see this EDU at all, as it only considers the segment at the top of the tree, which merely expresses a factual hypothesis, free of any polar connotation.

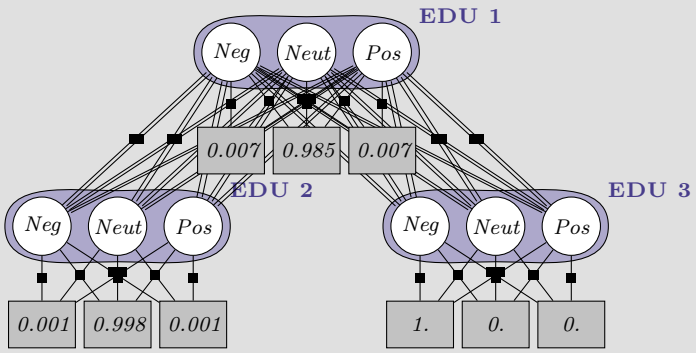
Example 6.3.5 (An Error Made by the ROOT System)

Tweet: [Die NSA weiss auch von dir ...]$_1$ [Nützt uns auch nichts .]$_2$ [%NegSmiley]$_3$

[*The NSA also knows about you . . .*]$_1$ [*It doesn't help us either*]$_2$ [*%NegSmiley*]$_3$

Gold Label: **negative**

Predicted Label: **neutral***



Finally, the last example (6.3.6) shows an error made by the LAST baseline system, which predicts the neutral label for a negative tweet based on the polarity of its right-most EDU. This unit indeed admits some positive moments with regard to the sad news expressed in the first segment, but in contrast to the movie description from Example 6.3.1, where the last sentence completely overturned the polarity of the whole text, this time, the final opinion does not alter the general negative mood of the message, but only dampens its effect.

Example 6.3.6 (An Error Made by the LAST System)

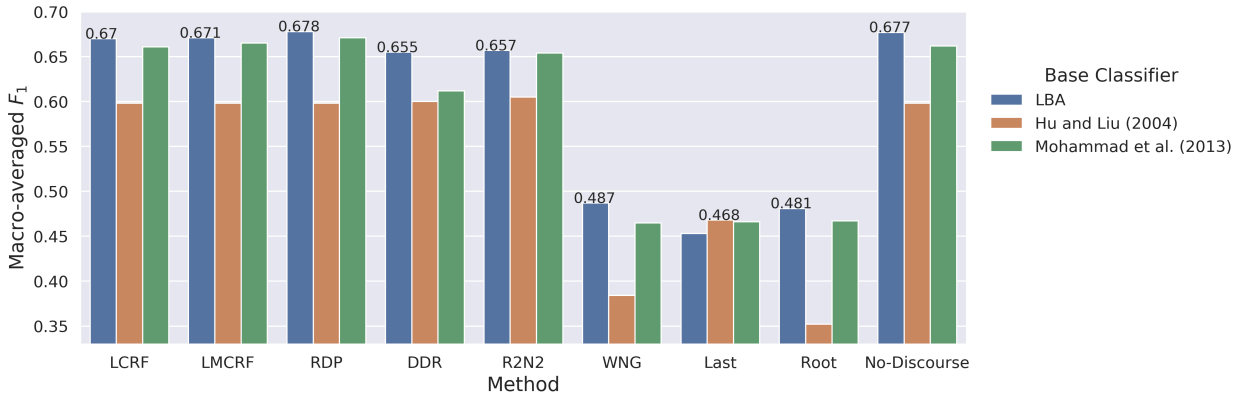
Tweet: [' (:'(:'(Die letzte Aussprache war wohl das schwerste Telefonat meines gesamten Lebens :'(:'(:'(]$_1$ [Aber wir gehen friedlich und als F . . .]$_2$ [' *(:'(:'(The last talk was probably the most difficult call in my entire life :'(:'(:'(*]$_1$ [*But we go apart peacefully and as f . . .*]$_1$

Gold Label: **negative**

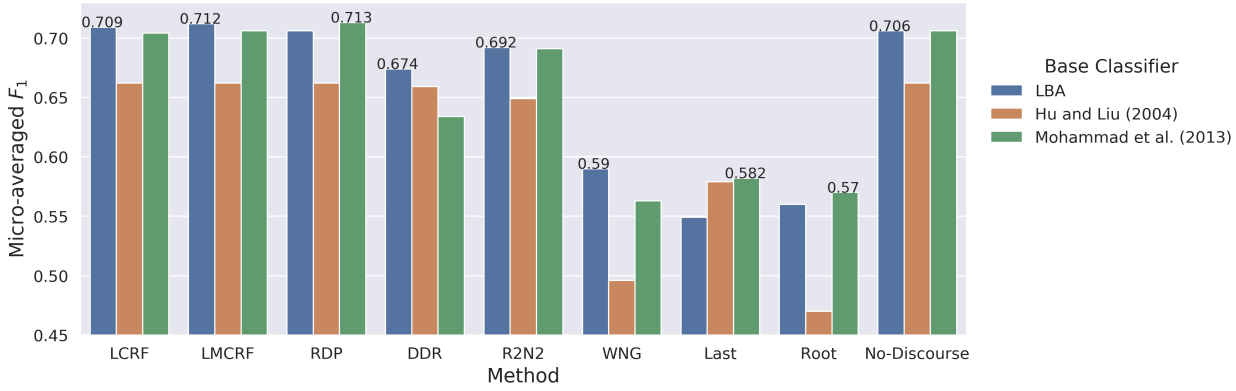
Predicted Label: **neutral***

6.4 Evaluation

As we could see from the examples in Section 6.3, the results of our proposed methods were significantly limited by two key factors: (i) scores predicted by the base sentiment system for tweets and EDUs and (ii) the structure of RST trees constructed for these messages. In order to estimate the effect of these factors more precisely, we decided to rerun our experiments, trying alternative solutions for each of these aspects.



(a) `Macro-`F_1



(b) `Micro-`F_1

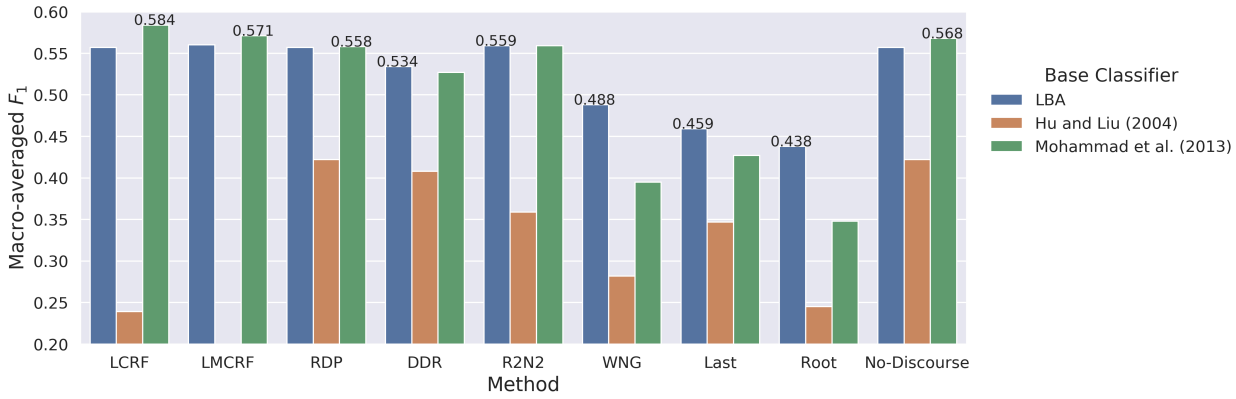
Figure 6.10: Results of discourse-aware sentiment analysis methods with different base classifiers on the PotTS corpus

6.4.1 Base Classifier

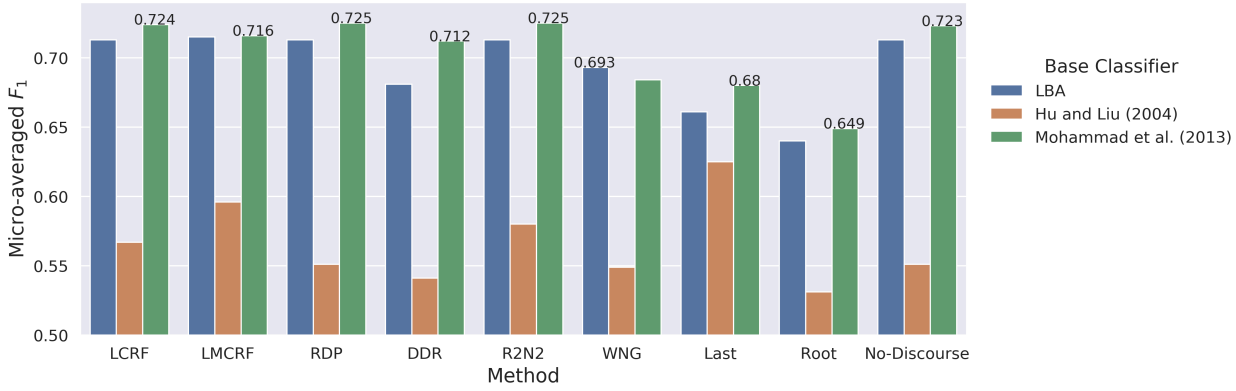
To assess the impact of the former factor (the quality of the base sentiment classifier), we replaced all polarity scores produced by the LBA system with the respective values predicted by the best lexicon- and machine-learning–based MLSA methods (the systems of Hu and Liu [2004] and Mohammad et al. [2013] respectively) and retrained all DASA approaches on the updated data, subsequently evaluating them on the PotTS and SB10k test sets. The results of this evaluation are shown in Figures 6.10 and 6.11.

As we can see from the first figure, our initially chosen LBA approach is indeed a more amenable basis to almost all discourse-aware sentiment methods on the PotTS corpus. A few exceptions to this general rule are the macro-averaged F_1-score of the Last baseline, which surprisingly improves in combination with the lexicon-based system, and the micro-average of the RDP and Last methods, which attain their best results (0.713 and 0.582) in conjunction with the SVM classifier of Mohammad et al. (2013).

A slightly different situation is observed on the SB10k corpus though. On this dataset, LBA still leads to higher macro-F_1–scores for DDR, R2N2, WNG, Last, and Root; but



(a) `Macro-`F_1



(b) `Micro-`F_1

Figure 6.11: Results of discourse-aware sentiment analysis methods with different base classifiers on the SB10k corpus

the approach of Mohammad et al. (2013) improves the results of LCRF, LMCRF, RDP, and No-Discourse. The SVM classifier is also the unequivocal leader in terms of the micro-averaged F_1, yielding the highest scores for all systems except WNG. Unfortunately, the lexicon-based predictor of Hu and Liu (2004) performs much weaker than SVM and LBA: the highest macro- and micro-averaged F_1-scores achieved with this approach run up to 0.422 (RDP) and 0.625 (Last) respectively. The most disappointing result for us, however, is that the LMCRF system completely fails to predict any polar class except Neutral on the SB10k test set when trained with the scores of this method (see Figure 6.11a). Similarly, LCRF yields considerably lower scores in combination with this solution, reaching only 0.239 macro-F_1.

6.4.2 Parsing Quality and Relation Scheme

Another factor that could significantly influence the results of discourse-aware methods was the quality of automatic RST parsing and the set of discourse relations distinguished by the parser system. Although improving the results of DPLP let alone manually annotating the

complete PotTS and SB10k datasets was beyond the scope of our dissertation (even though we have made such attempt, see [Sidarenka et al., 2015a]), we decided to check whether at least evaluating the DASA methods on manually annotated data would improve their results. For this purpose, we asked a student assistant to segment and parse 88% of the tweets from the PotTS test set[12] and tested all DASA approaches on these hand-crafted RST data.

Method	**Positive**			**Negative**			**Neutral**			**Macro**	**Micro**
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1	F_1
PotTS											
LCRF	0.82	0.82	**0.82**	**0.66**	0.55	0.6	0.69	0.75	0.72	0.71	0.747
LMCRF	**0.83**	0.81	**0.82**	0.65	0.55	0.6	0.69	**0.78**	**0.73**	0.709	0.749
RDP	0.8	0.84	**0.82**	0.64	0.58	0.61	**0.72**	0.71	0.72	**0.718**	0.751
DDR	0.78	0.75	0.77	0.58	**0.66**	**0.62**	0.66	0.63	0.64	0.693	0.698
R2N2	0.81	0.82	0.81	0.64	0.53	0.58	0.68	0.74	0.71	0.697	0.737
WNG	0.58	0.74	0.65	0.63	0.19	0.29	0.51	0.51	0.51	0.47	0.558
Last	0.55	**0.86**	0.67	0.51	0.11	0.18	0.56	0.35	0.43	0.426	0.55
Root	0.58	0.56	0.57	0.58	0.25	0.35	0.43	0.6	0.5	0.46	0.513
No-Discourse	0.81	0.84	**0.82**	0.65	0.57	0.61	**0.72**	0.73	**0.73**	0.716	**0.753**

Table 6.2: Results of discourse-aware sentiment analysis methods on the PotTS corpus with manually annotated RST trees

As we can see from the results in Table 6.2, the scores of all systems except WNG, Last, and Root increase by three to four percent. Even the macro-averaged F_1-measure of the discourse-unaware classifier improves from 0.677 to 0.716, as does its micro-F_1–score, which rises from 0.706 to 0.753 F_1. These last changes, however, are exclusively due to the reduced size of the test data (on which the base classifier performs better than on the full test set), since the discourse-unaware method does not take RST trees into account. Unfortunately, this time, No-Discourse also outperforms all discourse-aware approaches in terms of the micro-averaged F_1, achieving an accuracy of 75,3%, although it still loses to the Recursive Dirichlet Process on the macro-averaged metric, yielding a 0.2% worse result than RDP (0.716 versus 0.718 macro-F_1). Another surprising finding for us is that in the gold discourse annotation, EDUs that determine the actual polarity of the tweet are unlikely to appear either at the end of a message or at the top of its RST tree, which leads to the degradation of the scores for the Last and Root baselines.

Although manually annotated RST trees do improve the results of most discourse-aware sentiment methods, this fact is of little help to us if we are bound to the output of an automatic parser. A common way to improve the quality of automatic RST analysis and ease the task of DASA methods is to reduce the number of discourse relations distinguished by the parsing system. Drawing on the work of Bhatia et al. (2015), we also used this

[12]Unfortunately, due to the limited availability of the student, we could not annotate the whole test set.

approach, projecting all discourse relations from the Potsdam Commentary Corpus (Stede and Neumann, 2014) to the binary set of CONTRASTIVE and NON-CONTRASTIVE ones. Although similar approximations were made in almost all other discourse-aware solutions (cf. Chenlo et al., 2013; Heerschop et al., 2011; Zhou et al., 2011), we were not sure whether the subset that we used was indeed optimal and sufficient to reflect all possible discourse interactions that could play an important role in sentiment composition.

To answer this question, we retrained the DPLP parser on the PCC, using the subsets of relations proposed by Chenlo et al. (2013), Heerschop et al. (2011), and Zhou et al. (2011), and also tried the original set of all RST links from the Potsdam Commentary Corpus. A detailed overview of these sets is given in Table 6.3.

Scheme	**Relation Set**	**Equivalence Classes**
Bhatia et al.	{CONTRASTIVE, **NON-CONTRASTIVE**}	CONTRASTIVE := {ANTITHESIS, ANTITHESIS-E, COMPARISON, CONCESSION, CONSEQUENCE-S, CONTRAST, PROBLEM-SOLUTION}.
Chenlo et al.	{ATTRIBUTION, BACKGROUND, CAUSE, COMPARISON, CONDITION, CONSEQUENCE, CONTRAST, ELABORATION, ENABLEMENT, EVALUATION, EXPLANATION, JOINT, OTHERWISE, TEMPORAL, **OTHER**}	
Heerschop et al.	{ATTRIBUTION, BACKGROUND, CAUSE, CONDITION, CONTRAST, ELABORATION, ENABLEMENT, EXPLANATION, **OTHER**}	
PCC	{ANTITHESIS, BACKGROUND, CAUSE, CIRCUMSTANCE, CONCESSION, CONDITION, CONJUNCTION, CONTRAST, DISJUNCTION, E-ELABORATION, ELABORATION, ENABLEMENT, EVALUATION-N, EVALUATION-S, EVIDENCE, INTERPRETATION, JOINT, JUSTIFY, LIST, MEANS, MOTIVATION, OTHERWISE, PREPARATION, PURPOSE, REASON, RESTATEMENT, RESTATEMENT-MN, RESULT, SEQUENCE, SOLUTIONHOOD, SUMMARY, UNCONDITIONAL, UNLESS, UNSTATED-RELATION}	
Zhou et al.	{CONTRAST, CONDITION, CONTINUATION, CAUSE, PURPOSE, **OTHER**}	CONTRAST := {ANTITHESIS, CONCESSION, CONTRAST, OTHERWISE}; CONTINUATION := {CONTINUATION, PARALLEL}; CAUSE := {EVIDENCE, NONVOLITIONAL-CAUSE, NONVOLITIONAL-RESULT, VOLITIONAL CAUSE, VOLITIONAL-RESULT};

Table 6.3: RST relations used in the original Potsdam Commentary Corpus and different discourse-aware sentiment methods
*(default relation, which subsumes the rest of the links, is shown in **boldface**)*

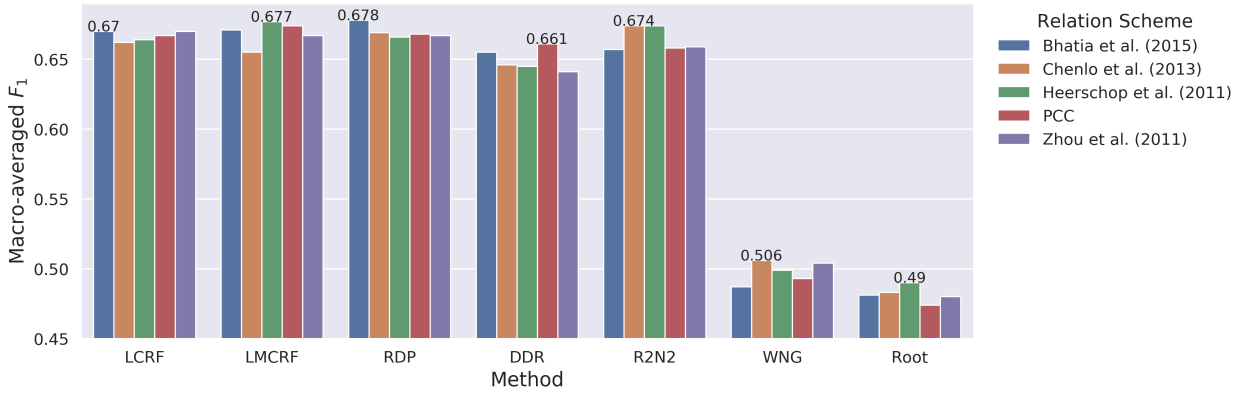
To check whether cardinalities of these sets indeed correlated with the quality of automatic RST parsing, we evaluated each retrained system on the held-out PCC test data and present the results of this evaluation in Table 6.4. As is evident from the scores, coarser relation

schemes in fact improve parsing quality, especially in terms of relation F_1. In the most extreme case (*e.g.*, Bhatia et al., which has only two links, versus PCC, which comprises 34 relations), these gains can reach up to seven percent. However, with respect to other metrics (span and nuclearity F_1), the gaps are notably smaller and might even be in favor of the richer relation set (cf. nuclearity F_1 for PCC).

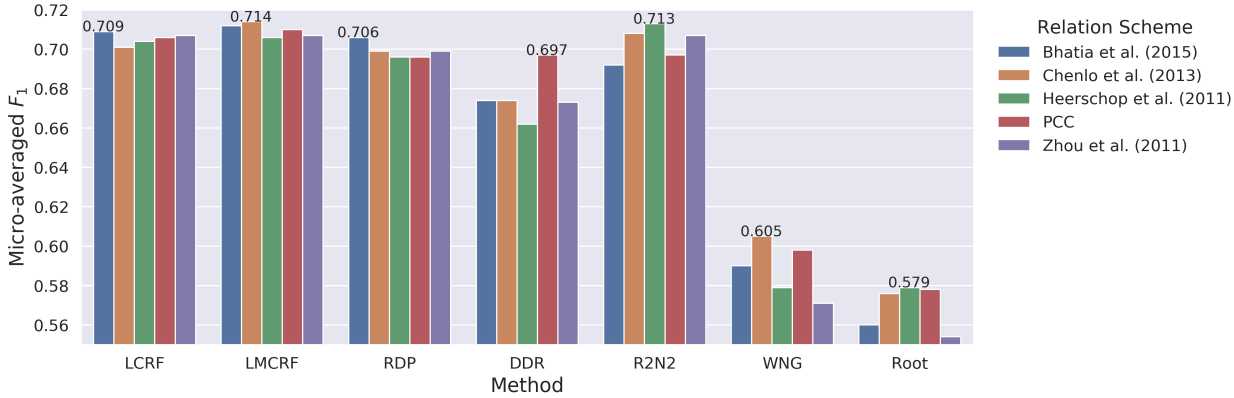
Relation Scheme	**Span** F_1	**Nuclearity** F_1	**Relation** F_1
Bhatia et al.	**0.777**	0.512	**0.396**
Chenlo et al.	0.769	0.505	0.362
Heerschop et al.	0.774	0.51	0.361
PCC	0.776	**0.534**	0.326
Zhou et al.	0.776	0.501	0.388

Table 6.4: Results of the DPLP parser on PCC 2.0 with different relation schemes

To see how this varying quality affected the net results of discourse-aware sentiment methods, we re-evaluated all DASA approaches on the updated automatic RST trees and show the results of this evaluation in Figures 6.12 and 6.13.



(a) `Macro`-F_1

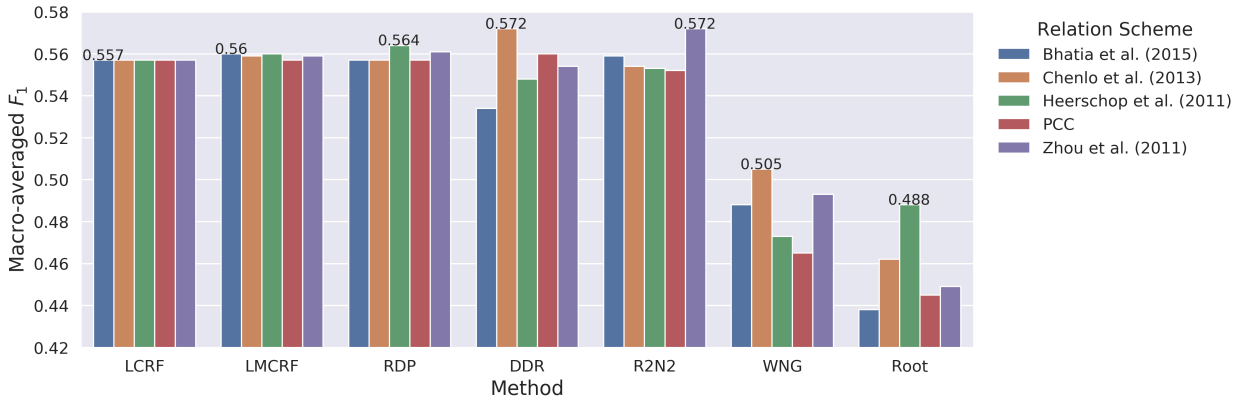


(b) `Micro`-F_1

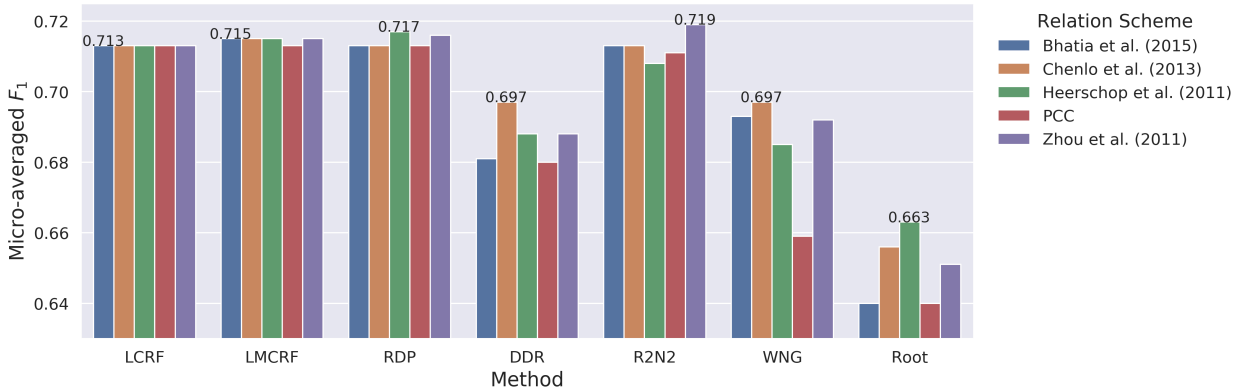
Figure 6.12: Results of discourse-aware sentiment classifiers for different relation schemes on the PotTS corpus

As it turns out, latent-marginalized CRF can still hold the overall record in both macro- and micro-averaged F_1 on the PotTS corpus, although its margin to the closest competitor (R2N2) is relatively small, amounting to only 0.1 percent. Interestingly enough, both top-performing methods (LMCRF and R2N2) achieve their best results with richer relation sets than the one we used in our initial experiment: For example, LMCRF attains its highest macro-score in combination with the relation scheme of Heerschop et al. (2011) and yields the best micro-F_1 when used with the scheme of Chenlo et al. (2014). The rhetorical recursive neural network, vice versa, attains its highest macro-average with the latter relation set and reaches its best micro-F_1 in conjunction with the former subset.

A different situation is observed with other DASA approaches though. For example, LCRF and RDP perform best when used with the initially chosen set of Bhatia et al. (2015). On the other hand, discourse-depth reweighting strongly benefits from the full unconstrained set of PCC relations, which is probably due to the better nuclearity classification achieved with this scheme. Finally, WNG and ROOT reach their best results with the relation subsets proposed by Chenlo et al. and Heerschop et al., respectively.



(a) Macro-F_1



(b) Micro-F_1

Figure 6.13: Results of discourse-aware sentiment classifiers for different relation schemes on the SB10k corpus

A much more uniform situation is observed on the SB10k corpus (see Figure 6.13), where

the F_1-scores of our methods vary only slightly across different relation schemes. The only significant improvements that we can notice this time are higher macro- and micro-averaged F_1s achieved by the RDP approach in combination with the Heerschop et al.'s subset. This subset is also most amenable to the ROOT baseline, which reaches 0.488 macro-F_1 and 0.663 micro-F_1, significantly improving on its initial results. At the same time, discourse-depth reweighting and the approach of Wang and Wu capitalize on the relations defined by Chenlo et al. so much that the former system even achieves the highest overall macro-F_1–score (0.572), being on a par with the R2N2 system.

6.5 Summary and Conclusions

At this point, our chapter has come to an end and, concluding it, we would like to recap that in this part of the thesis:

- we have presented an overview of the most popular approaches to automatic discourse analysis (RST, PDTB, and SDRT) and explained why we think that one of these frameworks (Rhetorical Structure Theory) would be more amenable to the purposes of discourse-aware sentiment analysis than the others;
- to substantiate our claims and to see whether the lexicon-based attention system introduced in the previous chapter would indeed benefit from information on discourse structure, we segmented all microblogs from the PotTS and SB10k corpora into elementary discourse units using the SVM-based segmenter of Sidarenka et al. (2015b) and parsed these messages with the RST parser of Ji and Eisenstein (2014), which had been previously retrained on the Potsdam Commentary Corpus (Stede and Neumann, 2014);
- afterwards, we estimated the results of existing discourse-aware sentiment methods (the systems of Wang et al. [2015b] and Bhatia et al. [2015]) and also evaluated two simpler baselines (in which we the predicted semantic orientation of a tweet by taking the polarity of its last and root EDUs), getting the best results with the R2N2 solution of Bhatia et al. (2015) (0.657 and 0.559 macro-F_1 on PotTS and SB10k respectively);
- we could, however, improve on these scores and also outperform the plain LBA system (although by a not very large margin) with our three proposed discourse-aware sentiment solutions: latent and latent-marginalized conditional random fields and Recursive Dirichlet Process; pushing the macro-averaged F_1-score on PotTS up to 0.678 and increasing the result on SB10k to 0.56 macro-F_1;

- a subsequent evaluation of these approaches with different settings showed that the results of all discourse-aware methods largely correlated with the scores of the base sentiment classifier and also revealed an important drawback of the latent-marginalized CRFs, which failed to predict any positive or negative instance on the test set of the SB10k corpus when trained in combination with the lexicon-based approach of Hu and Liu (2004);
- nevertheless, almost all DASA solutions could improve their scores when tested on manually annotated RST trees or used with a richer set of discourse relations.

Afterword

It is hard to believe, but at this point we have finally reached the home stretch of our 161-page long marathon and, preparing the final spurt, we should first recall the main milestones that we have seen along this way:

- As you might remember, we started off by summarizing the history of sentiment analysis, going back to its very origins in the ancient Greek philosophy and tracing its development to the present day;
- Afterwards, to see what the current state of the art in opinion mining would yield on German Twitter, we created a corpus of $\approx 8,000$ German tweets, collecting these messages for four different topics (federal elections, papal conclave, general political discussions, and casual everyday conversations). To ensure a good recall of opinionated statements in the resulting dataset, we grouped all microblogs into three formal categories (tweets with a polar term from the SentiWS lexicon, messages containing a smiley, and all remaining microblogs) and sampled an equal number of tweets (666) for each of the four topics from each of these three categories. After annotating the corpus in three steps (initial, adjudication, and final), we attained a reliable level of inter-annotator agreement for all elements (sentiments, sources, targets, polar terms, downtoners, negations, and intensifiers), finding that both selection criteria (topics and formal traits) significantly affected the distribution of sentiments and polar terms and the reliability of their annotation;
- Then, at the first checkpoint, we compared existing German sentiment lexicons, which were translated from English resources and revised by human experts, with lexicons that were generated automatically from scratch with the help of state-of-the-art dictionary-, corpus-, and word-embedding–based methods. An evaluation of these approaches on our corpus showed that semi-automatically translated polarity lists were generally better than the automatically induced ones, reaching 0.587 macro-F_1 and attaining 0.955 micro-F_1–score on the prediction of polar terms. Furthermore, among fully automatic methods, dictionary-based systems showed stronger results than their corpus- and word-embedding–based competitors, yielding 0.479 macro-F_1 and 0.962

micro-F_1. We could, however, improve on the latter metric (pushing it to 0.963) with our proposed linear projection solution, in which we first found a line that maximized the mutual distance between the projections of seed vectors with opposite semantic orientations and then projected the embeddings of all remaining words on that line, considering the distance of these projections to the median as polarity scores of respective terms;

- In Chapter 4, we turned our attention to the fine-grained sentiment analysis, in which we tried to predict the spans of sentiments, targets, and holders of opinions using two most popular approaches to this task: conditional random fields and recurrent neural networks. We obtained our best results (0.287 macro-F_1) with the first-order linear-chain CRFs. We could, however, increase these scores by using alternative topologies of CRFs (second-order linear-chain and semi-Markov CRFs) and also boost the macro-averaged F_1 to 0.38 by taking a narrower interpretation of sentiment spans (in which we only assigned the SENTIMENT tag to polar terms). Further evaluation of these methods proved the utility of the text normalization step (which raised the macro-F_1 of the CRF-method by almost 3%) and task-specific word embeddings with the least-squares fallback (which improved the macro-F_1–score of the GRU system by 1.4%);

- Afterwards, in Chapter 5, we addressed one of the most popular objective in contemporary sentiment analysis—message-level sentiment analysis (MLSA). To get a better overview of the numerous existing systems, we compared three larger families of MLSA methods: dictionary-, machine-learning–, and deep-learning–based ones; finding that the last two groups performed significantly better than the lexicon-based approaches (the best macro-F_1–scores of machine- and deep-learning methods run up to 0.677 and 0.69 respectively, whereas the best lexicon-based solution [Hu and Liu, 2004] only reached 0.641 macro-F_1). Apart from this, we improved the results of many reimplemented approaches by changing their default configuration (*e.g.*, abandoning polarity changing rules of lexicon-based systems, using alternative classifiers in ML-based systems, or taking the least-squares embeddings for DL-based methods). In addition to the numerous reimplementations of popular existing algorithms, we also proposed our own solution—lexicon-based attention (LBA), in which we tried to unite the lexicon and deep-learning paradigms by taking a bidirectional LSTM network and explicitly pointing its attention to polar terms that appeared in the analyzed messages. With this solution, we not only outperformed all alternative DL systems but also improved on the scores of ML-based classifiers, attaining 0.69 macro-F_1 and 0.73 micro-F_1 on the PotTS corpus. Similarly to our findings of the previous chapter, we observed a strong positive effect of text normalization and task-specific embeddings with the least-squares approximation;

- Finally, in the last part, we tried to improve the results of the proposed LBA method by making it aware of the discourse structure. For this purpose, we segmented all microblogs from the PotTS and SB10k corpora into elementary discourse units, individually analyzing each of these segments with our MLSA classifier, and then estimated the overall polarity of a tweet by joining the polarity scores of its EDUs over the RST tree. We proposed three different ways of doing this joining: latent CRFs, latent-marginalized CRFs, and Recursive Dirichlet Process; obtaining better results than existing discourse-aware sentiment methods and also outperforming the original discourse-unaware baseline. In the concluding experiments, we further improved these scores by using manually annotated RST trees and richer subsets of discourse relations.

Conclusions

Now that we have gone past all these landmarks, it is time to unbag the questions which we had asked ourselves at the beginning of this endeavor, and try to answer them again, equipped with all knowledge that we have acquired during our run. Here we go:

- **Can we apply opinion mining methods devised for standard English to German Twitter?**

  Yes, we can, but the success of these approaches might significantly vary depending on the task, the size, and the reliability of the training data, as well as the evaluation metric that we use. For example, dictionary-based lexicon methods achieved fairly good results on their objective, but this success was mostly due to the high quality of the GermaNet annotation. On the other hand, our manually labeled PotTS corpus was evidently too small for fine-grained sentiment systems, which failed to generalize to unseen tweets despite their very high scores on the training set. Message-level sentiment approaches, vice versa, seemed to be quite happy with the size of the training dataset, attaining good results on both corpora (PotTS and SB10k). Nevertheless, we again experienced a lack of data while working on discourse-aware enhancements, many of which hit the same ceiling of the macro-averaged F_1-scores.

  Apart from these difficulties arising from insufficient data, we also noticed a significant degradation of the scores for systems whose original tasks and evaluation metrics were different from ours. For example, the lexicon generation method of Esuli and Sebastiani (2005) was originally designed to assign polarity scores to all *synsets* found in the WordNet and not to produce a list of polar *words*. Similarly, the RNTN approach of Socher et al. (2013) was trained and evaluated on all syntactic subtrees of a document and not only at the top text level. Likewise, the system of Yessenalina and Cardie

(2011) was devised for doing ordinal logistic regression and not polarity classification, as in our case. As a result, all these approaches showed lower scores than their competitors in our evaluation, even though they are undoubtedly well suited for their original data and tasks.

Due to the high diversity of methods, metrics, and tasks, it is difficult to provide a general recipe for transferring existing English sentiment systems to German Twitter, but we still would like to formulate at least a few rules of thumb, which came up during our experiments:

- **Prefer methods that are closest to your training objective** and that were trained under similar conditions w.r.t. the amount of data, their class distribution and domain;
- **Put every single setting of these methods into question**—bear in mind that things that work well in the original cases are not guaranteed to work in your situation.[13] The more options you try, the better will be your results;
- **Try using manually labeled resources for your target domain**, if they are available, but pay attention to the quality of their annotation—it often matters more than the corpus size;
- If there are manually annotated data, **prefer machine-learning methods to hard-coded rules**—they will penalize their bad components automatically by themselves;
- **Do not use randomly initialized word embeddings for deep-learning systems**—initialize them with language-model vectors (which are cheap to obtain). Otherwise, your model might get stuck in a very bad local optimum.

- **Which groups of approaches are best suited for which sentiment tasks?**

  Based on our evaluation, we answer this question as follows:

  - *Sentiment lexicon generation* is more amenable to dictionary-based solutions, provided that there exists a sufficiently big, reliably annotated lexical taxonomy for these systems. If there is no such resource, one should better resort to word-embedding–based algorithms;
  - With a limited amount of training data, *fine-grained sentiment analysis* can be better addressed with probabilistic graphical models, such as conditional random fields with hand-crafted features;

[13] In this respect, it is important to realize that every classification task is merely an attempt to solve a system of equations, so that methods that are good at solving one system might completely fail to solve another set of equations.

  - On the other hand, plain *message-level sentiment analysis* can be efficiently tackled with both machine- and deep-learning algorithms, such as SVM, logistic regression, or RNN;
  - But probabilistic graphical models strike back at *discourse-aware sentiment methods*, where they might even outperform pure neural-network solutions, although the margin of these improvements is not that large.

  Thus, probabilistic model can still hold their ground when it comes to structured prediction, but the difference of these algorithms from and their improvements upon neural networks are gradually vanishing.

- **How much do word- and discourse-level analyses affect message-level sentiment classification?**

  Our evaluation in Section 5.6.2 showed that the macro-averaged F_1-scores of our proposed lexicon-based attention system varied by up to 14% (from 0.64 to 0.69 macro-F_1 on the PotTS corpus, and from 0.44 to 0.58 on SB10k) depending on the lexicon used by this approach. At the same, discourse enhancements could only improve the results of LBA by at most 1.5% percent (from 0.677 to 0.678 on PotTS, and from 0.557 to 0.572 on SB10k). Although it appears as if the lexicon component were more important to a sentiment system, we would like to preclude such incorrect conclusion, because (a) a full-fledged sentiment solution should take into account both linguistic levels (words and discourse) and (b) these relative results might look different if we expand the analyzed domain to longer documents or apply discourse-aware methods to complete discussion threads.

- **Does text normalization help analyze sentiments?**

  Yes, it definitely does. As we could see in Chapters 4 and 5, normalization significantly improves the quality of fine-grained and message-level sentiment analyses, boosting the results on the former task by up to 4% (see Table 4.9) and improving the macro-averaged F_1-measure of message-level sentiment methods by up to 25% (see Table 5.12).

  The only question that remained unanswered in this context is which normalization steps exactly improve the scores of sentiment systems. To make up for this omission, we separately deactivated each individual step of our text normalization pipeline (unification of Twitter phenomena, spelling correction, and normalization of slang terms) and rerun our message-level classification experiments using the lexicon-based attention system. As we can see from the results in Table 6.5, the micro-averaged F_1-scores on both datasets benefit most from the unification of Twitter-specific phenomena, sinking by almost 19% when this component is deactivated. This step is also most useful for the macro-F_1 on the SB10k corpus, whereas the macro-average on PotTS mostly capitalizes on the normalization of slang terms.

Method	Positive			Negative			Neutral			Macro	Micro
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	$F_1^{+/-}$	F_1
PotTS											
with normalization	0.76	0.84	0.79	0.6	0.56	0.58	0.75	0.68	0.72	0.69	0.73
w/o unification of Twitter phenomena	$0.51^{-0.25}$	$0.87^{+0.03}$	$0.64^{-0.05}$	$0.57^{-0.03}$	$0.4^{-0.16}$	$0.47^{-0.11}$	$0.68^{-0.07}$	$0.22^{-0.46}$	$0.34^{-0.38}$	$0.56^{-0.13}$	$0.54^{-0.19}$
w/o spelling correction	$0.67^{-0.09}$	0.84	$0.74^{-0.05}$	$0.61^{+0.01}$	$0.34^{-0.22}$	$0.44^{-0.14}$	$0.74^{-0.01}$	0.68	$0.71^{-0.01}$	$0.59^{-0.1}$	$0.69^{-0.04}$
w/o slang normalization	$0.59^{-0.17}$	$0.87^{+0.03}$	$0.7^{-0.09}$	0.6	$0.17^{-0.39}$	$0.26^{-0.32}$	$0.72^{-0.03}$	$0.6^{-0.08}$	$0.65^{-0.07}$	$0.48^{-0.21}$	$0.64^{-0.09}$
SB10k											
with normalization	0.6	0.72	0.66	0.47	0.42	0.44	0.84	0.8	0.82	0.55	0.73
w/o unification of Twitter phenomena	$0.36^{-0.24}$	$0.85^{+0.13}$	$0.5^{-0.16}$	$0.6^{+0.13}$	$0.25^{-0.17}$	$0.35^{-0.09}$	0.84	$0.51^{-0.29}$	$0.63^{-0.19}$	$0.43^{-0.12}$	$0.55^{-0.18}$
w/o spelling correction	$0.54^{-0.06}$	$0.71^{-0.01}$	$0.61^{-0.05}$	$0.54^{+0.07}$	$0.26^{-0.16}$	$0.35^{-0.09}$	$0.79^{-0.05}$	$0.79^{-0.01}$	$0.79^{-0.03}$	$0.48^{-0.07}$	$0.7^{-0.03}$
w/o slang normalization	$0.55^{-0.05}$	$0.71^{-0.01}$	$0.62^{-0.04}$	$0.64^{+0.17}$	$0.2^{-0.22}$	$0.3^{-0.14}$	$0.78^{-0.06}$	$0.82^{+0.02}$	$0.8^{-0.02}$	$0.46^{-0.09}$	$0.7^{-0.03}$

Table 6.5: $LBA^{(1)}$ results without single text normalization steps

- **Can we do better than existing approaches?**

  Yes, we can:

  - we improved the macro-averaged results of existing lexicon-generation methods with our proposed linear-projection algorithm;
  - we increased the scores of fine-grained analysis by redefining the topologies of CRFs;
  - our lexicon-based attention network outperformed many of its competitors on message-level classification;
  - and, finally, we surpassed the discourse-unware baseline and other existing discourse-aware sentiment solutions with the proposed latent-marginalized CRFs and Recursive Dirichlet Process.

Contributions

Apart from answering the above questions and pushing the state of the art for several major sentiment tasks on the PotTS and SB10k corpora, we have also paved the way for other researchers who want to work on the same topics by releasing the data and the code that we used in our experiments:

- the Potsdam Twitter Sentiment (PotTS) corpus is available at: `https://github.com/WladimirSidorenko/PotTS`;

- scripts and executables used in our lexicon generation chapter can be downloaded from: `https://github.com/WladimirSidorenko/SentiLex`;
- for our text normalization pipeline and fine-grained sentiment methods, please refer to: `https://github.com/WladimirSidorenko/TextNormalization`;
- furthermore, you can find our MLSA approaches at: `https://github.com/WladimirSidorenko/CGSA`;
- and get all discourse-aware solutions from: `https://github.com/WladimirSidorenko/DASA`;
- last but not least, we have released the discourse segmenter, the adapted DPLP parser, and our modified version of RST-Tool, which was adjusted to the annotation of multilogues, at:
  - `https://github.com/WladimirSidorenko/DiscourseSegmenter`,
  - `https://github.com/WladimirSidorenko/RSTParser`, and
  - `https://github.com/WladimirSidorenko/RSTTool`, respectively.

In addition to open-sourcing all projects, we have also made a few attempts to increase the visibility of our research with the following publications:

- the rule-based text normalization was described in (Sidarenka et al., 2013);
- the PotTS corpus was presented in (Sidarenka, 2016b);
- in (Sidarenka and Stede, 2016), we summarized the evaluation of existing sentiment lexicons (a separate paper on the linear projection algorithm was withdrawn due to a mistake in the initial implementation);
- in (Sidarenka, 2016a) and (Sidarenka, 2017), we described our initial experiments on message-level classification;
- furthermore, we introduced the SVM-based discourse segmenter in (Sidarenka et al., 2015b);
- and sketched our pilot study of discourse annotation in Twitter in (Sidarenka et al., 2015a).

Unfortunately, due to the lack of experience at the initial stage of working on this dissertation and limited time at the concluding stage, I[14] was not able to publish more or at higher-level venues. I apologize for that.

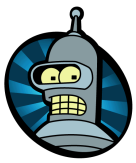
[14]Throughout this work we have been using the scientific "we," considering the reader as a companion in

Limitations

Much to my regret, the initial lack of academic expertise has also prevented me from running this scientific marathon faster, better, and, most sadly, along more exciting places. Alas, in the "Sentiment Analysis of German Twitter", I have concentrated more on the "Sentiment Analysis" part, much at the cost of "German Twitter". I wish I had tried out more sophisticated cross-lingual methods for adapting English methods to German, elaborated more on linguistic traits of microblogs, and addressed the social aspect of the Twitter network. The main reason why I have not done all these things is that, six years ago, when I started working on this dissertation completely from scratch, with neither code, nor data, nor any proper plan in my head, I was so overwhelmed by the abundance of works on opinion mining and text normalization that I decided to answer the questions whether existing sentiment methods work, whether text normalization helps, and whether I could improve on these methods, first. Regrettably, these questions have pretty much preoccupied me since then. Another reason why I refrained from addressing certain topics (why, for example, I did not extend the Rhetorical Structure Theory to multilogues) is because properly handling these problems would require another dissertation and would certainly first need to be done for English in order to get any international attention.

Final Remarks

Nevertheless, I believe that with this thesis I have basically built a theme park on the moon. Although this park is still missing a slot machine and a few carousels, it does have a nice card table and many other kinds of funny amusements. Another good thing about it is that new carousels are now easier to build; existing amusement rides work for free and better than in many other places; and, finally, the park itself might still entertain its occasional guests. You might have enjoyed this theme park as well, or you might have not—I appreciate either of your decisions and would like to thank you in any case for your visit.



our marathon. But because at this point I start describing the limitations of this work, which I am the only person responsible for, I would like to exclude the reader from this criticism by switching to the solitary "I".

Appendix A

Annotation Guidelines of the Sentiment Corpus

A.1 Introduction

In this assignment, your task is to annotate sentiments in a corpus of Twitter messages. We define *sentiments* as polar (either positive or negative) evaluative opinions about some persons, entities, or events. Your goal is to annotate both: text spans denoting the opinions (*sentiments*) and text spans signifying the evaluated entities and events (*sentiment targets*). In addition to that, you also have to label opinions' holders (*sentiment sources*) and lexical elements that might significantly affect the polarity or the intensity of a sentiment. These elements are:

- *polar terms*, which are words or phrases that unequivocally possess an evaluative lexical meaning in and of themselves (these are typically words like *hassen* [*hate*], *bewundern* [*admire*], *schön* [*nice*] etc.);
- *intensifiers* and *diminishers* (or *downtoners*), which are words and expressions that increase or decrease the evaluative sense of a polar term. Examples of intensifiers are words like *sehr* (*very*), *besonders* (*especially*), or *insbesondere* (*particularly*). Typical examples of diminishers are *ein wenig* (*a little*), *ein bisschen* (*a bit*), *gewissermaßen* (*to a certain degree*), etc.;
- and, finally, *negations*, which are words or expressions that completely flip the polarity of a polar term or sentiment to the opposite (*e.g.*, *nicht* gut [*not* good] or *kein* Talent [*not* a talent]).

A.2 Annotation Tool

For annotating this corpus, you need to install `MMAX2`, a freely available annotation tool, which you can download at:

`http://sourceforge.net/projects/mmax2/files/mmax2/mmax2_1.13.003/MMAX2_1.13.003b.zip/download`

After you have downloaded this file, unzip the received archive, change to the newly created directory `1.13.003/MMAX2` in your shell and execute the following commands:

```
chmod u+x ./mmax2.sh
nohup ./mmax2.sh &
```

An `MMAX2` window will then appear on your screen. If you have never used `MMAX2` before, please read its user manual `mmax2quickstart.pdf`, which you can find in the subdirectory `MMAX2/Docs` of the downloaded archive.

A.3 Corpus Files

You should also have received a copy of corpus files either as a tar-gzipped archive or via a version control system. In the former case, you need to unpack the downloaded `.tgz` file using the following command:

```
tar -xzf archive-name.tgz
```

After that, a directory called `PotTS` will appear in your current folder.

You can find your annotation files in the subdirectory `PotTS/corpus/annotator-ANNOTATOR_ID`, where ANNOTATOR_ID is the ID number that has been previously assigned to you by the supervisor. In order to load an annotation file into your `MMAX2` program, click on the menu `File -> Load`. In the displayed pop-up window, select the path to the `PotTS/annotator-ANNOTATOR_ID` directory, and click on one of the `*.mmax` files in this folder.

A.4 Tags and Attributes

Below, you can find a short list of all labels and their possible attributes that will be used in this assignment:

1. `sentiment`s with the attributes:
   (a) `polarity`,
   (b) `intensity`,
   (c) `sarcasm`;
2. `target`s with the attributes:
   (a) `preferred`,

   (b) `anaph-ref`,

   (c) `sentiment-ref`;

3. `source`s with the attributes:

   (a) `anaph-ref`,

   (b) `sentiment-ref`;

4. `polar-term`s with the attributes:

   (a) `polarity`,

   (b) `intensity`,

   (c) `sarcasm`,

   (d) `sentiment-ref`;

5. `intensifier`s with the attributes:

   (a) `degree`,

   (b) `polar-term-ref`;

6. `diminisher`s with the attributes:

   (a) `degree`,

   (b) `polar-term-ref`;

7. and, finally, `negation`s with the single attribute:

   (a) `polar-term-ref`.

A more detailed description of these attributes is given in the following sections.

A.4.1 sentiment

Definition. *Sentiments* are polar subjective evaluative opinions about people, entities, or events.

According to this definition, a sentiment must always fulfill the following three criteria:

- it has to be **polar**, *i.e.*, it must always reflect either positive or negative attitude to its respective target. Neutral, non-evaluative statements such as *Ich glaube, er wird heute früher kommen* (*I think he will be earlier today*) must not be marked as `sentiment`s;
- it has to be **subjective**, *i.e.*, you must not assign this tag to statements of objective facts, such as *Beim Angriff wurden 14 Glasscheiben beschädigt* (*14 glass plates were broken during the attack*), even if you have a personal polar attitude to such events. Sentiments should always reflect *the personal opinion of their holder, not yours*;
- a sentiment has to be **evaluative**, which means that it must always refer to an explicit target and judge about its properties. You should not regard cases like *Ich bin heute so glücklich* (*I am so happy today*) as `sentiment`s, because such statements do not evaluate anything in particular, but only express general mood of the author.

Example. Typical examples of sentiments are evaluative sentences similar to the one shown below.

Example A.4.1
[*Ich mag den neuen James Bond Film nicht.*]$_{sentiment}$

([I don't like the new James Bond movie.]$_{sentiment}$)

This example expresses a personal subjective evaluation; the opinion is strictly negative; and it also has an explicit evaluation target—the *movie*. Therefore, we enclose this sentence in the `sentiment` tags.

We also consider contrastive comparisons as a special type of evaluations. But unlike other sentiments, comparisons typically express a relative subjective judgment, *i.e.*, an object is regarded as better or worse than another, but we usually do not know whether the author actually likes or dislikes any of them. To distinguish such cases, we have introduced a special `comparison` value for the `polarity` attribute of this element, which you should use to distinguish such cases.

You should not label as `sentiments` polar opinions whose truth status is unknown. These are sentences like *Ich weiß nicht, ob ich meinen Bruder mag* (*I don't know whether I like my brother*), where neither we nor the author actually know whether the author likes or dislikes her brother. Exceptions from this rule are cases like *Ich zweifle, dass er ein guter Mensch ist* (*I doubt that he is a good man*) or *Ich glaube nicht, dass er diesen Preis verdient hat* (*I don't think that he has deserved this award*), which express author's disagreement with positive evaluations and, consequently, acts as a negative judgment. Special care should be taken when dealing with questions and subjunctive sentences though (see FAQ Section in the extended version[1] of these guidelines).

Boundaries. `sentiment` tags should enclose both the evaluated object (target) and the evaluative expression (typically a polar-term), *i.e.*, you should put these tags around the *minimal complete syntactic or discourse-level unit in which both (target and evaluation expression) appear together.*

In Example A.4.2, for instance, the evaluated object is *Buch* (*book*), the evaluative expression is *langweiliges* (*boring*), and the minimal syntactic unit that simultaneously comprises both of these elements is the noun phrase *ein langweiliges Buch* (*a boring book*). Therefore, we annotate the noun phrase with the `sentiment` tags, but do not enclose anything else inside these labels.

Example A.4.2

Auf dem Tisch lag [ein langweiliges Buch]$_{sentiment}$.

(There was [a boring book]$_{sentiment}$ on the table.)

Sentiments are not restricted to just noun phrases, they can also be expressed by complete clauses or even multiple sentences (discourse units). In these cases, a `sentiment` span still has to be *complete*, *i.e.*, it should capture the common syntactic or discourse-level ancestor of the evaluative expression and its target, as well as all other descendents of that common ancestor element; and it has to be *minimal*, *i.e.*, it should only enclose the closest possible ancestor, without including its parent or sibling elements.

Example A.4.3 demonstrates a sentiment expressed by a clause:

[1] `https://github.com/WladimirSidorenko/PotTS/blob/master/docs/annotation_guidelines.pdf`

Example A.4.3
Wir akzeptieren das, weil [*wir alle ein bisschen in Petterson verliebt sind*]$_{sentiment}$.

(We accept this because [*we all are a little bit in love with Petterson*]$_{sentiment}$.*)*

In this sentence, the evaluative statement is made about *Petterson*, who acts as sentiment's target; the author says that they all *in ihn verliebt sind* (*are in love with him*), which is a subjective evaluation. Both (target and evaluative expression) appear in one verb phrase, whose head is the link verb *sein* (*to be*). Consequently, we enclose the complete verb phrase including its grammatical subject *wir* (*we*) in the `sentiment` tags.

Attributes. After you have annotated the `sentiment` span, you should next set the values of its attributes, which are summarized in Table A.1.

A.4.2 target

Definition. *Targets* are objects or events that are evaluated by a sentiment.

Because sentiments are required to be evaluative, there always must be at least one target for each `sentiment` element.

Example. An example of a sentiment target is given in sentence A.4.4:

Example A.4.4
Mein Bruder ist nicht begeistert von [*dem neuen Call of Duty*]$_{target}$.

(My brother is not impressed by [*the new Call of Duty*]$_{target}$.*)*

In this message, the author tells us about the opinion of her brother regarding the new version of a computer game. The computer game is the object of this evaluation, so you shall label it as a `target`.

Boundaries. As for `sentiments`, you have to put the `target` tags around the minimal complete syntactic or discourse-level unit that denotes the evaluated entity or event. These are usually noun phrases (*e.g.*, *Mir wird's schlecht, wenn ich* [*diese Werbung*]$_{target}$ *im Fernsehen sehe* [*I feel sick when I see this* [*ad*]$_{target}$ *on TV*]) or clauses (*e.g.*, *Ich hasse wenn* [*Voldemort mein Shampoo benutzt*]$_{target}$. [*I hate when* [*Voldemort is using my shampoo*]$_{target}$]).

If a sentiment has multiple targets, you shall label each one of them separately (see Example A.4.5).

Attribute	Value	Meaning
polarity	*positive*	sentiment expresses a positive attitude to its respective target, *e.g.*, *Es war ein fantastischer Abend (It was a fantastic evening)*;
	negative (default)	sentiment expresses a negative attitude to its respective target, *e.g.*, *Seine Schwester ist einfach unausstehlich (His sister is simply obnoxious)*
	comparison	sentiment expresses a comparison of two objects with preference given to one of them, *e.g.*, *Mir gefällt das rote Kleid mehr als das blaue (I like the red dress more than the blue one)*
intensity	*weak*	sentiment expresses a weak evaluative opinion, *e.g.*, *Der Auftritt war mehr oder weniger gut (The appearance was more or less good)*
	medium (default)	sentiment has a middle emotional expressivity, *e.g.*, *Mir hat das neue Album gut gefallen (I enjoyed the new album)*
	strong	sentiment expresses a very emotional polar statement, *e.g.*, *Dieses Festival war einfach umwerfend!!! (This festival was simply terrific!!!)*
sarcasm	*true*	the opinion is derisive, *i.e.*, its actual polarity is the opposite of its literal meaning, although there are no immediate modifiers in the nearby context. An example of a sarcastic sentiment is the following passage: *Mein Jüngerer ist in der Prüfung durchgefallen. Klasse! (My youngest has failed his exam. Well done!)* In this case, you should set the polarity attribute of the sentiment to `negative` and the value of the `sarcasm` attribute to `true`.
	false (default)	no sarcasm is present—polar attitude has its literal meaning.

Table A.1: Attributes of `sentiments`

Example A.4.5

Meiner Mutter haben [*Nelken*]$_{target}$ *und* [*Dahlien*]$_{target}$ *immer gefallen.*

(My mother has always liked [*carnations*]$_{target}$ *and* [*dahlias*]$_{target}$*.)*

Similarly, in comparisons, you have to annotate each compared object with a separate tag. In addition to that, you should also set the value of the `preferred` attribute to `false` for the object that is dispreferred in the comparison (see Example A.4.6).

Example A.4.6

Ich mag [*Domino-Eis*]$_{target:preferred=true}$ *mehr als* [*Magnum*]$_{target:preferred=false}$*.*

(I like [*Domino ice cream*]$_{target:preferred=true}$ *more than* [*Magnum*]$_{target:preferred=false}$*.)*

Attributes. Further possible attributes of `targets` are given in Table A.2.

A.4.3 source

Definition. Sentiment *sources* are immediate author(s) or holder(s) of evaluative opinions. These are typically the author of a message, or officials whose opinion is cited.

If sentiment's holder is not explicitly mentioned in the tweet, it is implicitly assumed that it is the user who wrote that microblog, and you need not annotate anything as a source in this case.

Example. An example of an explicitly mentioned source is the pronoun *Sie* (*she*) in the following sentence.

Example A.4.7

[*Sie*]$_{source}$ *mag die neue Farbe nicht*

([*She*]$_{source}$ *doesn't like the new color)*

Note that in citations you should only label the immediate person or the institution whose original opinion is cited, but should not annotate the citing person as a `source` (see Example A.4.8).

Example A.4.8

Laut Staatsanwalt soll die [*Angeklagte*]$_{source}$ *sich missbilligend über ihren Vorgesetzten geäußert haben.*

(According to the attorney, the [*defendant*]$_{source}$ *had made disapproving remarks about her boss.)*

Attribute	Value	Meaning
preferred	*true (default)*	in comparisons, this value means that the respective target is considered better than another compared object, *e.g.*, Die neue Frisur *passt ihr garantiert besser als die alte (*The new hairstyle *suits her definitely better than the old one)*;
	false	in comparisons, this value signifies the target element that is considered worse than its counterpart, *e.g.*, *Die zweite Saison von Breaking Bad war viel spannender als* die dritte *(The second season of Breaking Bad was much more exciting than* the third one*)*;
sentiment-ref	$\overrightarrow{}$ *(directed edge)*	a directed edge pointing from `target` to its respective `sentiment`. You need to draw this edge in two cases: • when the `target` is located at intersection of two different `sentiment`s (in this case, you should draw an edge from `target` to `sentiment`, which this `target` actually belongs to), • when the target of an opinion is expressed outside the `sentiment` span;
anaph-ref	$\overrightarrow{}$ *(directed edge)*	a directed edge pointing from `target` expressed by a pronoun or pronominal adverb to its respective non-pronominal antecedent (in order to draw this edge, you also need to annotate the antecedent as `target`)

Table A.2: Attributes of `targets`

Boundaries. For determining the boundaries of `source`s, you should proceed in a similar way as you did for `target`s and `sentiment`s, *i.e.*, only annotate complete minimal syntactic units. Sources are most commonly expressed by noun phrases. As with `target`s, if the source of a sentiment is expressed by multiple separate noun phrases, you should label each of them separately (see Example A.4.9).

Example A.4.9
[*Ihr*]$_{source}$ *und* [*ihrer Mutter*]$_{source}$ *gefällt die neue Farbe nicht.*
(Neither [*she*]$_{source}$ *and* [*her mother*]$_{source}$ *likes the new color)*

Attributes. The attributes of the `source` tag are fully identical to the attributes of the `target` elements and are recapped in Table A.3.

Attribute	Value	Meaning
sentiment-ref	$\overrightarrow{\textit{(directed edge)}}$	see Table A.2
anaph-ref	$\overrightarrow{\textit{(directed edge)}}$	see Table A.2

Table A.3: Attributes of `sources`

A.4.4 polar-term

Definition. *polar-terms* are words or phrases that have an inherent evaluative meaning.

Example. An example of a polar-term is the word *ekelhaft* (*disgusting*) in sentence A.4.10.

Example A.4.10
Beim Aufräumen des Zimmers haben wir einen [*ekelhaften*]$_{\textit{polar-term}}$ *Teller mit verschimmeltem Essen unter dem Bett gefunden.*

(When we cleaned the room, we found a [*disgusting*]$_{\textit{polar-term}}$ *plate with moldy food under the bed.)*

In contrast to `source`s and `target`s, which should only be annotated in the presence of a `sentiment`, you always have to label polar terms in the text irrespective of any other tags.

Note, however, that because many words and idioms are ambiguous and can have several different meanings, it can often be the case that only some of these meanings are evaluative and subjective. In such cases, you should only label such words if their actual sense in the given context is polar. If these words denote an objective entity or fact, you must not use this tag.

Example A.4.11
Dieser Wein ist ein echtes [*Juwel*]$_{\textit{polar-term}}$ *in meiner Kollektion.*

(This wine is a real [*jewel*]$_{\textit{polar-term}}$ *in my collection.)*

Koh-i-Noor ist das teuerste Juwel heutzutage.

(Koh-i-Noor is the most expensive jewel nowadays.)

In Example A.4.11, for instance, the meaning of the word *Juwel* (*jewel*) is metaphoric and subjective in the first sentence, but literal and objective in the second statement. So you should only annotate this word as `polar-term` in the former case, but disregard it in the latter.

Boundaries. `polar-terms` are typically expressed by:

- nouns, *e.g.*, *Held (hero)*, *Ideal (ideal)*, *Betrüger (fraudster)*;
- adjectives or adverbs, *e.g.*, *schön (nice)*, *zuverlässig (reliably)*, *hinterhältig (devious)*, *heimtückisch (insidiously)*;
- verbs, *e.g.*, *lieben (to love)*, *bewundern (to admire)*, *hassen (to hate)*;
- idioms, *e.g.*, *auf die Nerven gehen (to get on one's nerves)*;
- smileys, *e.g.*, :), :-(, ☺, ☹.

If a `polar-term` represents an idiomatic phrase, you shall always annotate the complete idiom. If a verb has an evaluative sense only in conjunction with certain prepositions (*e.g.*, *to go for sth.* in the sense of *to like*), you shall annotate both the verb and the preposition with a single pair of tags (check the `MMAX` manual to see how to annotate discontinuous spans).

Attributes. When determining the `polarity` of a `polar-term`, you should disregard any possible contextual modifiers such as intensifiers or negations and set the value of this attribute to the lexical (*i.e.*, *prior*) polarity of that term (see Example A.4.12).

Example A.4.12

Es war keine [*gute*]$_{\textit{polar-term:polarity=positive}}$ *Idee.*

(It was not a [*good*]$_{\textit{polar-term:polarity=positive}}$ *idea.)*

Apart from that, when determining the value of the `polarity` attribute of a `polar-term`, you should analyze its polarity from the perspective of the holder of the opinion towards the evaluated object. This means that in cases like *Ich vermisse meine Freundin* (*I miss my girlfriend*), the polarity of the polar-term *vermissen* (*to miss*) is still positive because the author has a positive attitude to his girlfriend, and consequently feels sad about of her absence.

Further attributes of `polar-terms` include `intensity`, `sarcasm`, and `sentiment-ref`; their possible values are summarized in Table A.4.

A.4.5 intensifier

Definition. *Intensifiers* are elements that increase the expressivity or the evaluative sense of a polar term.

Example. An example of intensifier is the word *sehr* (*very*) in sentence A.4.13.

Attribute	Value	Meaning
polarity	*positive*	polar term has a positive evaluative meaning, *e.g.*, *gut (good), verhimmeln (to ensky), Prachtkerl (corker)* etc.
	negative (default)	polar term expresses a negative evaluation of its target, *e.g.*, *versauen (to botch up), rotzig (snotty), Dreckskerl (scum)* etc.
intensity	*weak*	polar-term has a weak evaluative sense, *e.g.*, *so-lala (so-so), nullachtfünfzehn (vanilla), durchschnittlich (mediocre)* etc.
	medium (default)	polar-term has middle stylistic expressivity, *e.g.*, *gut (good), schlecht (bad), robust (tough)* etc.
	strong	polar-term expresses a very strong positive or negative evaluation, *e.g.*, *allerbeste (bettermost), zum Kotzen (to make one puke), Kacke (shit)* etc.
sarcasm	*true*	polar-term is derisive, *i.e.*, its actual polarity is the opposite of its primary lexical sense even though there are no negations in the surrounding context
	false (default)	no sarcasm is present—the term has its literal polar meaning; this is the default
sentiment-ref	$\overrightarrow{}$ *(directed edge)*	an arrow pointing to the `sentiment` that this `polar-term` belongs to. You should only draw this edge if a `polar-term` is located at an intersection of two `sentiments` or outside of the `sentiment` span that it belongs to

Table A.4: Attributes of `polar-terms`

Example A.4.13

Wir suchen eine [*sehr*]$_{\textit{intensifier}}$ *zuverlässige Polin als Haushaltshilfe.*

(We are looking for a [*very*]$_{\textit{intensifier}}$ *reliable Polish woman as domestic help.)*

Boundaries. Intensifiers are usually expressed by adverbs or adjectives such as *sehr* (*very*) or *sicherlich* (*certainly*), but other ways of intensification are still possible (see Example A.4.14).

Example A.4.14

Dieser Junge ist stark [*wie ein Pferd*]$_{\textit{intensifier}}$.

(This boy is strong [as a horse]$_{\textit{intensifier}}$.)

Attributes. An `intensifier` must always relate to some `polar-term`, and you always have to explicitly show this relation by drawing a `polar-term-ref` edge from the `intensifier` to its modified polar expression.

Further possible attributes of `intensifier`s are shown in Table A.5.

Attribute	Value	Meaning
degree	*medium (default)*	the intensifier moderately increases the polar sense of the polar term, *e.g.*, *ziemlich (quite), recht (fairly)* etc.
	strong	the intensifier strongly increases the polar sense and stylistic markedness of the polar term, *e.g.*, *sehr (very), super (super), stark (strongly)* etc.
polar-term-ref	$\longrightarrow$ *(directed edge)*	a directed edge pointing from the intensifier to the `polar-term` whose meaning is being intensified

Table A.5: Attributes of `intensifier`s

A.4.6 diminisher

Definition. *Diminisher*s or *downtoner*s are words or phrases that decrease the polar lexical sense of a `polar-term`.

Example. In Example A.4.15, the diminisher is expressed by the adverb *weniger* (*less*).

Example A.4.15
[*Weniger*]$_{\textit{diminisher}}$ *erfolgreiche Unternehmen verzichten auf externe Berater.*

The [*less*]$_{\textit{diminisher}}$ *successful companies do not use external consultants.*

Attributes. Like intensifiers, diminishers must always relate to a polar term, and you also have to explicitly show this relation by using the `polar-term-ref` attribute; other attributes of `diminisher`s mainly coincide with those of intensifiers and are summarized in Table A.6.

A.4.7 negation

Definition. *Negation*s are elements that turn the polarity of a `polar-term` to the opposite.

Example. In Example A.4.16, for instance, the negative article *kein* (*not*) makes the *contextual* polarity of the word *interessant* (*interesting*) negative, even though the prior semantic orientation of this term is positive.

> **Example A.4.16**
> *Diese Geschichte war überhaupt nicht* [*interessant*]$_{negation}$*!*
>
> *This story was* [*not*]$_{negation}$ *interesting at all!*

The role of negations is closely related to that of diminishers. In order to help you better distinguish between these entities, we have listed the most obvious differences between the two elements:

- *Semantic differences:* diminishers only decrease the lexical sense of an polar-term, a but part of its original sense still remains active (*i.e., a hardly understandable speech* is still understandable); negations, on the other hand, fully deny that meaning and turn it to the complete opposite (*a not understandable speech* is absolutely unintelligible);
- *Part-of-speech differences:* diminishers are usually expressed by adjectives or adverbs, whereas negations are typically represented by the negative article *kein (no)*, the negation particle *nicht (not)*, or verbs or adjectives, *e.g., Es ist sehr zweifelhaft, dass die neue Version von Windows besser wird* (*It is very doubtful that the new Windows version will be any better*)

Attributes. The only attribute of negations is the mandatory edge `polar-term-ref`. You have to draw this edge from the `negation` to that `polar-term` that is negated. Like intensifiers and diminishers, negations must always refer to at least one polar item.

Attribute	Value	Meaning
degree	*medium (default)*	diminisher moderately decreases the polar sense of its respective `polar-term`, *e.g., wenig (few), bisschen (little)* etc.
	strong	diminisher strongly decreases the polar sense of the `polar-term`, *e.g., kaum (hardly)* etc.
polar-term-ref	$\overrightarrow{\text{}}$ *(directed edge)*	see Table A.5

Table A.6: Attributes of `diminishers`

Attribute	Value	Meaning
polar-term-ref	$\overrightarrow{\quad}$ *(directed edge)*	an edge from `negation` to the `polar-term` being negated

Table A.7: Attributes of `negations`

A.5 Summary

Summarizing all of the above, your task in this assignment is to find subjective evaluative opinions about some entities or events. You need to annotate these opinions with the `sentiment` tags and also determine the polarity and the intensity of the expressed attitudes. After that, you should assign the `target` tags to objects or events that are evaluated, and label the holders of these attitudes as `sources`. Both, `sources` and `targets`, can only exist in the presence of a `sentiment`.

Another important task is to annotate words and phrases that have a polar evaluative meaning. We call these words `polar-terms`, and you need to annotate them always, regardless of whether there is a targeted sentiment or not. If a `polar-term` is intensified, diminished, or negated by another word or phrase, you should also annotate the modifying element as well.

A.6 Examples

We conclude these guidelines with a couple of real-world annotation examples from our corpus, explaining our decisions for these annotations.

Example A.6.1

WAS HABEN ALLE MIT [IHREN
*[VERF*CKTEN]*$_{polar\text{-}term:polarity=negative,intensity=strong,sarcasm=false}$
[GRÜNEN AUGEN]$_{target}$*]*$_{sentiment:polarity=negative,intensity=strong,sarcasm=false}$

(WHAT DO THEY ALL HAVE WITH [THEIR
*[F*CKED]*$_{polar\text{-}term:polarity=negative,intensity=strong,sarcasm=false}$
[GREEN EYES]$_{target}$*]*$_{sentiment:polarity=negative,intensity=strong,sarcasm=false}$*)*

Explanation: In this case, there is an evaluative opinion about the green eyes of some persons. It is, however, unclear what is the author's attitude to the people themselves, we only can see that she thinks that the eyes of these people are *verf*ckt* (*f*cked*). Therefore, the `target` of this sentiment is the word *Augen* (*eyes*), and the `sentiment` span should enclose the noun phrase comprising that target and its evaluative term *f*cked*. Since the polar term is an intense abusive word, we set the polarity of this word and its enclosing `sentiment` to `negative` and the intensity of both tags to `strong`.[2]

[2]In the cases where we do not specify an attribute in the example, this attribute is assumed to have the default value.

Example A.6.2

[*Wo ist der* [*#Jubel*]$_{\textit{polar-term:polarity=positive,intensity=strong,sarcasm=true}}$ *von* [*#CDU*]$_{\textit{target}}$ [*#CSU*]$_{\textit{target}}$ *&* [*#FDP*]$_{\textit{target}}$ *über den Tod der Mieterin nach #Zwangsräumung?*]$_{\textit{sentiment:polarity=negative,intensity=medium,sarcasm=true}}$

[*Where is the* [*#exultation*]$_{\textit{polar-term:polarity=positive,intensity=strong,sarcasm=true}}$ *of* [*#CDU*]$_{\textit{target}}$ [*#CSU*]$_{\textit{target}}$ *&* [*#FDP*]$_{\textit{target}}$ *about the death of the renter after forced #eviction?*]$_{\textit{sentiment:polarity=negative,intensity=medium,sarcasm=true}}$

Explanation: In Example A.6.2, we have not labeled *Jubel von #CDU ... über den Tod von ...* (*the exultation of the #CDU ... about the death of ...*) as `sentiment`, because the truth status of this statement is unknown. But, on the other hand, the mere hypothesis that a political party could experience a glee feeling because of a renter's death is sarcastic. We can recognize it from the `polar-term` *#Jubel* (*#exultation*), whose prior semantic orientation is positive, but which suggests a negative attitude to the CSU party in this context, without any explicit contextual modifiers. Accordingly, we set the (prior) `polarity` of the term to `positive`, the polarity of its `sentiment` to `negative`, and the `sarcasm` attribute of both labels to `true`. Apart from having different polarities, `sentiment` and `polar-term` also have different intensities: since *#Jubel* (*#exultation*) expresses a higher degree of excitement than the word *Freude* (*joy*), we set its `intensity` to `high`. On the other hand, the overall sentiment expression is rather subtle and does not show high exaggeration of the author. So, we set the `intensity` of the `sentiment` to `medium` rather than `high`.

Another non-trivial case is shown in Example A.6.3, which we will analyze step by step:

Example A.6.3

RT @JochenFlasbarth : Guter #Spiegel-Titel , wie Welzer , Sloterdijk und andere Promi #Nichtwähler die Demokratie verspielen : Träge , frustriert

(RT @JochenFlasbarth : A good #Spiegel title , how Welzer , Sloterdijk, and other celebrity non-voters squander the democracy : Sluggish , frustrated)

Explanation: First of all, we have to look at words with unambiguous lexical polarity (polar-terms), as they are our primary cues for detecting sentiments. This tweet features one positive terms, *guter* (*good*), and there negative polar items, *verspielen* (*to squander*), *träge* (*sluggish*), and *frustriert* (*frustrated*). Since we have two sets of polar-terms with contradicting polarities, it is most likely that there also are two sentiments—one positive and one negative. The positive evaluation obviously pertains to the suggested #Spiegel title "wie Welzer , Sloterdijk und andere Promi #Nichtwähler die Demokratie verspielen: Träge , frustriert" ("*how Welzer , Sloterdijk, and other celebrity non-voters squander the democracy: Sluggish , frustrated*"). The author finds this title good, and the annotation of this sentiment then looks as follows:

Example A.6.4

[*RT* [*@JochenFlasbarth*]$_{\textit{source:sentiment\_ref=1}}$ *:*
[*Guter*]$_{\textit{polar-term:polarity=positive,intensity=medium,sarcasm=false, sentiment\_ref=1}}$ *#Spiegel-Titel ,*
[*wie Welzer , Sloterdijk und andere Promi #Nichtwähler die Demokratie verspie-*

len : Träge ,
frustriert$]_{target:sentiment\_ref=1}]_{sentiment:polarity=positive,intensity=medium,sarcasm=false,id=1}$

([*RT* [*@JochenFlasbarth*]$_{source:sentiment\_ref=1}$ *: A*
[*good*]$_{polar\text{-}term:polarity=positive,intensity=medium,sarcasm=false,\ sentiment\_ref=1}$ *#Spiegel title ,*
[*how Welzer , Sloterdijk, and other celebrity non-voters squander the democracy : Sluggish ,*
frustrated$]_{target:sentiment\_ref=1}]_{sentiment:polarity=positive,intensity=medium,sarcasm=false,id=1}$ *)*

The negative opinion, which is expressed by the terms *verspielen* (*to squander*), *träge* (*sluggish*), and *frustriert* (*frustrated*), obviously relates to the celebrity non-voters, *Welzer* and *Sloterdijk*; so, we annotate this evaluation as:

Example A.6.5

[*RT* [*@JochenFlasbarth*]$_{source:sentiment\_ref=2}$ *: Guter #Spiegel-Titel , wie*
[*Welzer*]$_{target:sentiment\_ref=2}$ *,* [*Sloterdijk*]$_{target:sentiment\_ref=2}$
und [*andere Promi #Nichtwähler*]$_{target:sentiment\_ref=2}$ *die Demokratie*
[*verspielen*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,sentiment\_ref=2}$ *:*
[*Träge*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,sentiment\_ref=2}$ *,*
[*frustriert*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,sentiment\_ref=2}$
$]_{sentiment:polarity=negative,intensity=medium,sarcasm=false,id=2}$

([*RT* [*@JochenFlasbarth*]$_{source:sentiment\_ref=2}$ *: A good #Spiegel title , how*
[*Welzer*]$_{target:sentiment\_ref=2}$ *,* [*Sloterdijk*]$_{target:sentiment\_ref=2}$*,*
and [*other celebrity non-voters*]$_{target:sentiment\_ref=2}$
[*squander*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,sentiment\_ref=2}$ *the democracy :*
[*Sluggishly*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,\ sentiment\_ref=2}$ *,*
[*frustrated*]$_{polar\text{-}term:polarity=negative,intensity=medium,sarcasm=false,\ sentiment\_ref=2}$
$]_{sentiment:polarity=negative,intensity=medium,sarcasm=false,id=2}$ *)*

In both cases, *@JochenFlasbarth* is the original author of the cited opinion, so we should label it as a `source`. But since there are two sentiment relations, we assign this tag twice, drawing an edge (in our example denoted by attribute `sentiment_ref`) to the respective `sentiment` element in each case.

Appendix B

Gradient Computation of the Optimized Projection Line

In order to prove the correctness of the gradient shown in Equation 3.2, let us first compute the partial derivative of the optimized distance function $f = \sum_{\vec{p}_+} \sum_{\vec{p}_-} \frac{1}{2}\left(\frac{\vec{b}\cdot(\vec{p}_+ - \vec{p}_-)}{\vec{b}^2}\vec{b}\right)^2$ w.r.t. to a single element $\vec{b}_j$ of the projection vector $\vec{b}$. Assuming that the length of this vector is normalized at each iteration step prior to calculating the derivative, we obtain:

$$\begin{aligned}
\frac{\partial}{\partial \vec{b}_j} f &= \frac{\partial}{\partial \vec{b}_j} \sum_{\vec{p}_+} \sum_{\vec{p}_-} \frac{1}{2}\left(\frac{\vec{b}\cdot(\vec{p}_+ - \vec{p}_-)}{\vec{b}^2}\vec{b}_j\right)^2 \\
&= \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \vec{b}_j \frac{\partial}{\partial \vec{b}_j} \frac{\vec{b}\cdot(\vec{p}_+ - \vec{p}_-)}{\vec{b}^2}\vec{b}_j \\
&= \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \vec{b}_j \left(\frac{(\vec{p}_+ - \vec{p}_-)_j \vec{b}^2 - 2\gamma \vec{b}_j}{\vec{b}^4}\vec{b}_j + \frac{\gamma}{\vec{b}^2}\right) \\
&= \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \vec{b}_j \left((\vec{p}_+ - \vec{p}_-)_j \vec{b}_j - 2\gamma \vec{b}_j^2 + \gamma\right) \\
&= \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \left((\vec{p}_+ - \vec{p}_-)_j \vec{b}_j^2 - 2\gamma \vec{b}_j \vec{b}_j^2 + \gamma \vec{b}_j\right),
\end{aligned} \tag{B.1}$$

where γ is defined as previously:

$$\gamma = \vec{b}\cdot(\vec{p}_+ - \vec{p}_-)\,.$$

Since Expression B.1 is identical for all j, we can estimate the final form of the gradient as:

$$\nabla f = \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \left((\vec{p}_+ - \vec{p}_-)\,\vec{b}^2 - 2\gamma \vec{b}\vec{b}^2 + \gamma \vec{b}\right) \tag{B.2}$$

$$= \sum_{\vec{p}_+} \sum_{\vec{p}_-} \gamma \left(\Delta - \gamma \vec{b}\right), \tag{B.3}$$

which is exactly the solution we provide in Equation 3.2.

Appendix C

CRF Training and Inference

C.1 Training

Traditionally, the main objective of CRF's training consists in finding such feature parameters $\boldsymbol{\Theta}$ that maximize the log-likelihood of a training set $\mathcal{D} = \{(\boldsymbol{x}[m], \boldsymbol{y}[m])\}_{m=1}^{M}$ (where M represents the total number of training examples, $\boldsymbol{x}[m]$ denotes the feature vector of the m-th example, and $\boldsymbol{y}[m]$ stands for the vector of its gold labels):

$$\boldsymbol{\Theta} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{m=1}^{M} \ell_{\boldsymbol{Y}|\boldsymbol{X}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{m=1}^{M} \ln p_{\boldsymbol{\Theta}}(\boldsymbol{y}[m]|\boldsymbol{x}[m]). \tag{C.1}$$

The conditional probability of gold labels for the m-th training instance ($p_{\boldsymbol{\Theta}}(\boldsymbol{y}[m]|\boldsymbol{x}[m])$) is typically computed as:

$$p_{\boldsymbol{\Theta}}(\boldsymbol{y}[m]|\boldsymbol{x}[m]) = \frac{\exp\Big(\sum_{i=1}^{n}\sum_{k}\theta_k f_k(\boldsymbol{y}[m], \boldsymbol{x}, i)\Big)}{Z_{\boldsymbol{x}}},$$

where n represents the total length of that instance (in the case of sentences, n is usually the number of tokens); θ_k and f_k are the weight and the value of the k-th feature; and $Z_{\boldsymbol{x}}$ is a normalization factor, which is estimated over all possible features f and label assignments $\mathcal{Y}^n$:

$$Z_{\boldsymbol{x}} := \sum_{\boldsymbol{y}' \in \mathcal{Y}^n} \exp\Big(\sum_{i=1}^{n}\sum_{k}\theta_k f_k(\boldsymbol{y}', \boldsymbol{x}[m], i)\Big).$$

The partial derivatives of feature weights, which are needed for optimizing the log-likelihood in Equation C.1, are known to be equal to the difference between the empirical and model's expectation of features over the whole training corpus:

$$\frac{\partial}{\partial\theta_k}\ell_{\boldsymbol{Y}|\boldsymbol{X}} = \sum_{m=1}^{M}\sum_{i=1}^{n}\big(f_k(\boldsymbol{y}[m], \boldsymbol{x}[m], i) - \boldsymbol{E}_{\boldsymbol{\Theta}}[f_k(\mathcal{Y}^n, \boldsymbol{x}, i)]\big) \tag{C.2}$$

C.1.1 Linear-chain CRFs

In the case of first-order linear-chain CRFs, these derivatives are usually estimated with the help of the forward-backward algorithm (a specific case of the belief-propagation method [Pearl, 1982]), which can be briefly described as follows.

For each position i of training instance $\boldsymbol{x}[m]$ and for each label y of tagset $\mathcal{Y}$, one first computes the forward score $\alpha[y][i]$ as:

$$\alpha[y][i] = \sum_{y' \in \mathcal{Y}} \alpha[y'][i-1] t(y', y, i-1, i) s(y, i) \tag{C.3}$$

where $t(y', y, i-1, i)$ is the exponentiated sum of transition features $f_t(y', y, \boldsymbol{x}, i-1, i)$ (which denote the transition from label y' at position $i-1$ to label y at position i) multiplied with their respective weights θ_t:

$$t(y', y, i-1, i) := \exp\Big(\sum_t \theta_t f_t(y', y, \boldsymbol{x}, i-1, i)\Big).$$

Similarly, $s(y, i)$ denotes the exponent of the sum of state features f_s times their weights θ_s:

$$s(y, i) := \exp\Big(\sum_s \theta_s f_s(y, \boldsymbol{x}, i)\Big).$$

The normalizing factor $Z_{\boldsymbol{x}}$ is then easily estimated as the sum of all values from the last column of matrix α:

$$Z_{\boldsymbol{x}} = \sum_{y \in \mathcal{Y}} \alpha[y][n].$$

After estimating the forward scores, one compute the backward scores β by applying the same procedure in reverse—from right to left:

$$\beta[y][i] = \sum_{y' \in \mathcal{Y}} \beta[y'][i+1] t(y, y', i, i+1) s(y', i+1). \tag{C.4}$$

The marginal probabilities p_m of state and transition features are then estimated as:

$$p_m(f_s(y, \boldsymbol{x}, i)) = \frac{1}{Z_{\boldsymbol{x}}} \alpha[y][i] \beta[y][i];$$
$$p_m(f_t(y', y, \boldsymbol{x}, i-1, i)) = \frac{1}{Z_{\boldsymbol{x}}} \alpha[y'][i-1] \beta[y][i] s(y, i).$$

Knowing these probabilities, one can easily obtain the gradient of feature weights using Equation C.2.

C.1.2 Semi-Markov CRFs

In contrast to linear-chain CRFs, semi-Markov conditional random fields do not model transitions between identical labels (*e.g.*, SOURCE → SOURCE), but instead try to partition the input into contiguous spans of identical tags and infer the most likely label assignment for these spans.

In order to do so, the model first determines the maximum possible length (L) of a segment with identical labels that exists in the training set. The forward and backward scores are then calculated as:

$$\begin{aligned}\alpha[y][i] = &\sum_{d=0}^{L-1} \sum_{\{y' \in \mathcal{Y} | y' \neq y\}} \alpha[y'][i-d-1] \\ &\times t(y', y, i-d-1, i-d) s(y, [i-d, i]);\end{aligned} \tag{C.5}$$

$$\begin{aligned}\beta[y][i] = &\sum_{d=0}^{L-1} \sum_{\{y' \in \mathcal{Y} | y' \neq y\}} \beta[y'][i+d+1] \\ &\times t(y, y', \boldsymbol{x}, i+d, i+d+1) s(y, [i, i+d]);\end{aligned} \tag{C.6}$$

where $s(y, [i, i+d])$ is the exponentiated sum of all state features $s(y, j)$ that are activated on the interval $[i, \ldots, i+d]$.

The marginal probabilities of state and transition features are then computed as:

$$\begin{aligned}p_m(f_s(y, \boldsymbol{x}, [i-d, i])) = &\frac{1}{Z_{\boldsymbol{x}}} s(y, [i-d, i]) \\ &\times \sum_{\{y' \in \mathcal{Y} | y' \neq y\}} \alpha[i-d-1][y'] t(y', y, i-d-1, i-d) \\ &\times \sum_{\{y'' \in \mathcal{Y} | y'' \neq y\}} \beta[i+1][y''] t(y, y'', i, i+1);\end{aligned}$$

and

$$\begin{aligned}p_m(f_t(y', y, \boldsymbol{x}, [i-d, i])) = &\frac{1}{Z_{\boldsymbol{x}}} \alpha[y'][i-1] \beta[y][i] \\ &\times t(y', y, i-1, i).\end{aligned}$$

C.1.3 Higher-order CRFs

In contrast to first-order models, which only consider the scores of one immediate label to the left when computing the α values or one immediate label to the right when estimating the β scores, higher-order CRFs keep separate track of each *sequence of labels* that might precede or follow the currently analyzed token.

In particular, instead of simply computing the scores for each tagset label $y \in \mathcal{Y}$ at each sentence position i, higher-order conditional random fields estimate these values for complete sequence of tags $y_1, \ldots, y_d$, where d is the order of the model.

This extension is possible for both linear-chain- and semi-Markov CRFs, and the way of estimating forward and backward scores as well as computing marginal probabilities p_m is almost identical to the respective original implementations. The only differences in the higher-order case are that

- it now becomes possible to use state and transition features that are associated with label chains up to length d and not only single tags (*e.g.*, instead of using a state feature which tells that verbs starting with "mis" are likely to be `sentiments`, one can refine the feature function and say that such words very probably represent `sentiments` preceded by `sources`, *i.e.*, are associated with the label sequence <`source`, `sentiment`>);

- secondly, when estimating the scores $\alpha[y_1, \ldots, y_d][i]$ and $\beta[y_1, \ldots, y_d][i]$, one does not simply iterate over all cells of the previous or next column of the corresponding matrix, but only considers those preceding or following states that allow the label sequence $y_1, \ldots, y_d$ at the i-th position. That is, Equations C.3 and C.4 become:

$$\alpha[y_1, \ldots, y_d][i] = \sum_{\{y'_1, \ldots, y'_d \in \mathcal{Y}^d | y'_2, \ldots, y'_d = y_1, \ldots, y_{d-1}\}} \alpha[y'_1, \ldots, y'_d][i-1] \times t\left((y'_1, \ldots, y'_d), (y_1, \ldots, y_d), i-1, i\right) s\left((y_1, \ldots, y_d), i\right)$$

  and

$$\beta[y_1, \ldots, y_d][i] = \sum_{\{y'_1, \ldots, y'_d \in \mathcal{Y}^d | y_2, \ldots, y_d = y'_1, \ldots, y'_{d-1}\}} \beta[y'_1, \ldots, y'_d][i+1] \times t\left((y_1, \ldots, y_d), (y'_1, \ldots, y'_d), i, i+1\right) s\left((y'_1, \ldots, y'_d), i+1\right),$$

  respectively. The same change also applies to Equations C.5 and C.6 in the case of semi-Markov models.

C.1.4 Tree-structured CRFs

The main difference between applying the belief-propagation algorithm to trees instead of linear chains is that the inference flow happens in a "vertical" way—from tree's leaves to its root and vice versa—whereas in the standard forward-backward setting, we typically compute the scores "horizontally"—from the left-most word of a sequence to the right-most one and then in the opposite direction.

More precisely, the α and β scores for trees are estimated as:

$$\alpha[y][p] = \prod_{c \in \text{children}(p)} \left(\sum_{y' \in \mathcal{Y}} \alpha[y'][c] t(y', y, \boldsymbol{x}, c, p) \right) s(y, p);$$
$$\beta[y][c] = \sum_{y' \in \mathcal{Y}} \frac{\alpha[y'][p]\beta[y'][p]}{\alpha_{c \to p}} t(y, y', \boldsymbol{x}, c, p);$$

where p is the index of the parent node of token c, and $\alpha_{c \to p}$ is the part of the α score of token p that has been previously propagated to it from its child c:

$$\alpha_{c \to p} := \sum_{y'' \in \mathcal{Y}} \alpha[y''][c] t(y'', y, \boldsymbol{x}, c, p)$$

The normalizing factor $Z_{\boldsymbol{x}}$ and marginal probabilities of state features are calculated in the same way as for the linear-chain models with the only difference that the partition factor Z is computed as the sum of the α-scores of the root word r and not of the last word n of the instance.

The marginal probabilities of transition features are computed using the following equation:

$$p_m(f_t(y', y, \boldsymbol{x}, c, p)) = \frac{\alpha[y'][c] t(y', y, c, p) \alpha[y][p] \beta[y][p]}{\alpha_{c \to p} Z_{\boldsymbol{x}}}.$$

C.2 Inference

Once model parameters have been learned, one applies the optimized model to new, unseen instances in order to predict their most probable labels—a task which is commonly refered to as *inference*.

For CRFs, inference actually boils down to computing the matrix α with the following minor modifications:

- First of all, instead of taking the sum over all previous labels $y' \in \mathcal{Y}$ when computing $\alpha[y][i]$, one only estimates the maximum possible score for that cell that is possible w.r.t. the probabilities of its preceding labels;
- Second, apart from storing the maximum (unnormalized) probability of label y at the i-th position, one also stores the label at the previous position $(i-1)$ that has lead to the maximum value of $\alpha[y][i]$.

In other words, in the case of linear-chain CRFs, we transform Equation C.3 into:

$$\alpha[y][i] = < \max_{y' \in \mathcal{Y}} \left(a(y', y, i-1, i) \right), \operatorname*{argmax}_{y' \in \mathcal{Y}} \left(a(y', y, i-1, i) \right) > \tag{C.7}$$

where

$$a(y', y, i-1, i) := \alpha[y'][i-1][0]t(y', y, i-1, i)s(y, i).$$

We similarly modify the α-computation in the semi-Markov case, but this time, apart from remembering the highest possible probability of the y-th label at the i-th position and its most likely predecessor, we also need to store the most probable length of the tag span y, *i.e.*,:

$$\alpha[y][i] = < \max_{y' \in \mathcal{Y}, d \in [1,\ldots,L]} \left(a(y', y, d, i)\right), \operatorname*{argmax}_{y' \in \mathcal{Y}, d \in [1,\ldots,L]} \left(a(y', y, d, i)\right) >$$

with a now defined as:

$$a(y', y, d, i) := \alpha[y'][i-d-1][0]t(y', y, i-d-1, i-d)s(y, [i-d, i]).$$

Finally, in the case of tree-structured CRFs, we could have basically completely re-used the formula from Equation C.7 if each tree node only had one child. But since, most of the time, this is rarely the case, we need to circumvent the need for storing multiple child labels in a single α cell because this significantly slows down the inference due to additional memory allocation on the fly. The way we do that is by applying the following trick: instead of storing in each cell $\alpha[y][i]$ the maximum possible score for the y-th tag at the i-th position and the labels of its children that have lead to this score, we store the score and the most likely tag of the i-th node that yielded the maximum possible value for the y-th tag at the parent position, *i.e.*,:

$$\alpha[y][c] = < \max_{y' \in \mathcal{Y}} \left(a(y', y, c, p)\right), \operatorname*{argmax}_{y' \in \mathcal{Y}} \left(a(y', y, c, p)\right) > \tag{C.8}$$

where

$$a(y', y, i-1, i) := \alpha[y'][c][0]t(y', y, c, p)s(y, p).$$

with c denoting the index of the child, and p standing for the index of the parent.

After computing the scores for the final node, we scan the last (root) column of the α matrix for the maximum value and trace back the complete sequence of labels that has yielded this score—a procedure which is commonly known as the Viterbi algorithm.

Bibliography

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment Analysis of Twitter Data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Antweiler, W. and Frank, M. Z. (2004). Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. *Journal of Finance*, 59(3):1259–1294.

Araque, O., Corcuera, I., Román, C., Iglesias, C. A., and Sánchez-Rada, J. F. (2015). Aspect Based Sentiment Analysis of Spanish Tweets. In Villena-Román, J., García-Morera, J., Cumbreras, M. Á. G., Martínez-Cámara, E., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015.*, volume 1397 of *CEUR Workshop Proceedings*, pages 29–34. CEUR-WS.org.

Aristotle (1954). *Rhetoric*. Modern Library, New York.

Aristotle (2010). *Rhetoric*. Cosimo Classics Philosophy. Cosimo, Incorporated.

Arnold, M. B. (1960). *Emotion and Personality: Vol. I Psychological Aspects*. Columbia University Press.

Asher, N., Benamara, F., and Mathieu, Y. Y. (2008). Distilling Opinion in Discourse: A Preliminary Study. In Scott, D. and Uszkoreit, H., editors, *COLING 2008, 22nd International Conference on Computational Linguistics, Posters Proceedings, 18-22 August 2008, Manchester, UK*, pages 7–10.

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

Awadallah, A. H. and Radev, D. R. (2010). Identifying Text Polarity Using Random Walks. In Hajic, J., Carberry, S., and Clark, S., editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 395–403. The Association for Computer Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Banfield, A. (1982). *Unspeakable Sentences*. Routledge and Kegan Paul.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA.

Barbosa, L. and Feng, J. (2010). Robust Sentiment Detection on Twitter from Biased and Noisy Data. In Huang, C. and Jurafsky, D., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 36–44. Chinese Information Processing Society of China.

Basile, V. and Nissim, M. (2013). Sentiment analysis on Italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107, Atlanta.

Bautin, M., Vijayarenu, L., and Skiena, S. (2008). International Sentiment Analysis for News and Blogs. In Adar, E., Hurst, M., Finin, T., Glance, N. S., Nicolov, N., and Tseng, B. L., editors, *Proceedings of the Second International Conference on Weblogs and Social Media, ICWSM 2008, Seattle, Washington, USA, March 30 - April 2, 2008*. The AAAI Press.

Baziotis, C., Pelekis, N., and Doulkeridis, C. (2017). DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *SemEval@ACL*, pages 747–754. The Association for Computer Linguistics.

Becker, L., Erhart, G., Skiba, D., and Matula, V. (2013). AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 333–340, Atlanta, Georgia, USA. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Bernard, J. R. L. (1986). *The Macquarie Thesaurus*. Macquarie Library, Sydney, Australia.

Bhatia, P., Ji, Y., and Eisenstein, J. (2015). Better Document-level Sentiment Analysis from RST Discourse Parsing. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2212–2218. The Association for Computational Linguistics.

Blair-Goldensohn, S., Neylon, T., Hannan, K., Reis, G. A., Mcdonald, R., and Reynar, J. (2008). Building a sentiment summarizer for local service reviews. In *In NLP in the Information Explosion Era*.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Blum, A. and Chawla, S. (2001). Learning from Labeled and Unlabeled Data using Graph Mincuts. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 19–26. Morgan Kaufmann.

Bohnet, B. (2009). Efficient Parsing of Syntactic and Semantic Dependency Structures. In Hajic, J., editor, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2009, Boulder, Colorado, USA, June 4, 2009*, pages 67–72. ACL.

Bohnet, B., Nivre, J., Boguslavsky, I., Farkas, R., Ginter, F., and Hajic, J. (2013). Joint Morphological and Syntactic Analysis for Richly Inflected Languages. *TACL*, 1:415–428.

Bosco, C., Patti, V., and Bolioli, A. (2013). Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT. *IEEE Intelligent Systems*, 28(2):55–63.

Breck, E., Choi, Y., and Cardie, C. (2007). Identifying Expressions of Opinion in Context. In Veloso, M. M., editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2683–2688.

Brin, S. and Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30(1-7):107–117.

Brody, S. and Diakopoulos, N. (2011). Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! Using Word Lengthening to Detect Sentiment in Microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 562–570. ACL.

Brown, P. F., Pietra, V. J. D., de Souza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-Based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2005). The AMI Meeting Corpus: A Pre-announcement. In Renals, S. and Bengio, S., editors, *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39. Springer.

Carlson, L. and Marcu, D. (2001). Discourse tagging reference manual. Technical report, University of Southern California / Information Sciences Institute.

Carlson, L., Marcu, D., and Okurovsky, M. E. (2001). Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the SIGDIAL 2001 Workshop, The 2nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Saturday, September 1, 2001 to Sunday, September 2, 2001, Aalborg, Denmark*. The Association for Computer Linguistics.

Cesteros, J. S., Almeida, A., and de Ipiña, D. L. (2015). DeustoTech Internet at TASS 2015: Sentiment Analysis and Polarity Classification in Spanish Tweets. In Villena-Román, J., García-Morera, J., Cumbreras, M. Á. G., Martínez-Cámara, E., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015.*, volume 1397 of *CEUR Workshop Proceedings*, pages 23–28. CEUR-WS.org.

Chenlo, J. M., Hogenboom, A., and Losada, D. E. (2013). Sentiment-Based Ranking of Blog Posts Using Rhetorical Structure Theory. In Métais, E., Meziane, F., Saraee, M., Sugumaran, V., and Vadera, S., editors, *Natural Language Processing and Information Systems - 18th International Conference on Applications of Natural Language to Information Systems, NLDB 2013, Salford, UK, June 19-21, 2013. Proceedings*, volume 7934 of *Lecture Notes in Computer Science*, pages 13–24. Springer.

Chenlo, J. M., Hogenboom, A., and Losada, D. E. (2014). Rhetorical Structure Theory for Polarity Estimation: An Experimental Study. *Data Knowl. Eng.*, 94:135–147.

Chesley, P., Vincent, B., Xu, L., and Srihari, R. K. (2006). Using Verbs and Adjectives to Automatically Classify Blog Sentiment. In *Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03, Stanford, California, USA, March 27-29, 2006*, pages 27–29. AAAI.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder

for Statistical Machine Translation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Choi, Y., Breck, E., and Cardie, C. (2006). Joint Extraction of Entities and Relations for Opinion Recognition. In Jurafsky, D. and Gaussier, É., editors, *EMNLP 2007, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 431–439. ACL.

Choi, Y. and Cardie, C. (2009). Adapting a Polarity Lexicon using Integer Linear Programming for Domain-Specific Sentiment Classification. In *EMNLP*, pages 590–598. ACL.

Choi, Y. and Cardie, C. (2010). Hierarchical Sequential Learning for Extracting Opinions and Their Attributes. In Hajic, J., Carberry, S., and Clark, S., editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, Short Papers*, pages 269–274. The Association for Computer Linguistics.

Cieliebak, M., Deriu, J., Egger, D., and Uzdilli, F. (2017). A Twitter Corpus and Benchmark Resources for German Sentiment Analysis. *SocialNLP 2017*, page 45.

Clematide, S. and Klenner, M. (2010). Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 7–13.

Cliche, M. (2017). BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. In *SemEval@ACL*, pages 573–580. Association for Computational Linguistics.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Crammer, K. and Singer, Y. (2001). Pranking with Ranking. In *NIPS*, pages 641–647.

Crammer, K. and Singer, Y. (2003). Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, 3:951–991.

Das, S. and Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *In Asia Pacific Finance Association Annual Conf. (APFA)*.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In Hencsey, G., White, B., Chen, Y. R., Kovács, L., and Lawrence, S., editors, *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 519–528. ACM.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. In Huang, C. and Jurafsky, D., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 241–249. Chinese Information Processing Society of China.

de Saussure, F. and Engler, R. (1990). *Cours de linguistique générale*. Number Bd. 2 in Cours de linguistique générale. Harrassowitz.

de Sousa, R. (2014). Emotion. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition.

Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., Luca, V. D., and Jaggi, M. (2016). SwissCheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision. In Bethard, S., Cer, D. M., Carpuat, M., Jurgens, D., Nakov, P., and Zesch, T., editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 1124–1128. The Association for Computer Linguistics.

Ding, X., Liu, B., and Zhang, L. (2009). Entity discovery and assignment for opinion mining applications. In IV, J. F. E., Fogelman-Soulié, F., Flach, P. A., and Zaki, M. J., editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 1125–1134. ACM.

Dragut, E. C., Yu, C. T., Sistla, A. P., and Meng, W. (2010). Construction of a sentimental word dictionary. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1761–1764. ACM.

Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Eisenstein, J. (2013). What to do about bad language on the internet. In Vanderwende, L., III, H. D., and Kirchhoff, K., editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 359–369. The Association for Computational Linguistics.

Eisenstein, J. (2017). Unsupervised Learning for Lexicon-Based Classification. In Singh, S. P. and Markovitch, S., editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3188–3194. AAAI Press.

Esuli, A. and Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In Herzog, O., Schek, H., Fuhr, N., Chowdhury, A., and Teiken, W., editors, *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 617–624. ACM.

Esuli, A. and Sebastiani, F. (2006a). SentiWordNet: a high-coverage lexical resource for opinion mining. Technical Report ISTI-PP-002/2007, Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR).

Esuli, A. and Sebastiani, F. (2006b). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pages 417–422.

Feng, S., Bose, R., and Choi, Y. (2011). Learning General Connotation of Words using Graph-based Algorithms. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1092–1103. ACL.

Feng, V. W. and Hirst, G. (2014). A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 511–521. The Association for Computer Linguistics.

Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L., and Vinyals, O. (2015). Sentence Compression by Deletion with LSTMs. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 360–368. The Association for Computational Linguistics.

Frege, G. (1892). Über Sinn und Bedeutung. In Textor, M., editor, *Funktion - Begriff - Bedeutung*, volume 4 of *Sammlung Philosophie*. Vandenhoeck & Ruprecht, G"ottingen.

Funk, A., Li, Y., Saggion, H., Bontcheva, K., and Leibold, C. (2008). Opinion analysis for business intelligence applications. In Duke, A., Hepp, M., Bontcheva, K., and Vilain, M. B., editors, *Proceedings of the First International Workshop on Ontology-supported Business Intelligence, OBI 2008, Karlsruhe, Germany, October 27, 2008*, volume 308 of *ACM International Conference Proceeding Series*, page 3. ACM.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*.

Ghose, A., Ipeirotis, P. G., and Sundararajan, A. (2007). Opinion Mining using Econometrics: A Case Study on Reputation Systems. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual LSTM (CLSTM) models for Large scale NLP tasks. *CoRR*, abs/1602.06291.

Gill, A. J., Gergle, D., French, R. M., and Oberlander, J. (2008). Emotion rating from short blog texts. In Czerwinski, M., Lund, A. M., and Tan, D. S., editors, *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*, pages 1121–1124. ACM.

Glance, N. S., Hurst, M., Nigam, K., Siegler, M., Stockton, R., and Tomokiyo, T. (2005). Deriving marketing intelligence from online discussion. In Grossman, R., Bayardo, R. J., and Bennett, K. P., editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 419–428. ACM.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Technical report*, pages 1–6.

Goyal, N. and Eisenstein, J. (2016). A Joint Model of Rhetorical Discourse Structure and Summarization. In *EMNLP Workshop on Structured Prediction for NLP*.

Günther, T. and Furrer, L. (2013). GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh*

International Workshop on Semantic Evaluation (SemEval 2013), pages 328–332, Atlanta, Georgia, USA. Association for Computational Linguistics.

Günther, T., Vancoppenolle, J., and Johansson, R. (2014). RTRGO: enhancing the GU-MLT-LT system for sentiment analysis of short messages. In Nakov, P. and Zesch, T., editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 497–502. The Association for Computer Linguistics.

Hamp, B. and Feldweg, H. (1997). GermaNet - a lexical-semantic net for German. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Han, B. and Baldwin, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 368–378. The Association for Computer Linguistics.

Hatzivassiloglou, V. and McKeown, K. (1997). Predicting the Semantic Orientation of Adjectives. In Cohen, P. R. and Wahlster, W., editors, *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 7-12 July 1997, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.*, pages 174–181. Morgan Kaufmann Publishers / ACL.

Hatzivassiloglou, V. and Wiebe, J. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31 - August 4, 2000, Universität des Saarlandes, Saarbrücken, Germany*, pages 299–305. Morgan Kaufmann.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society.

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., and de Jong, F. (2011). Polarity analysis of texts using discourse structure. In Macdonald, C., Ounis, I., and Ruthven, I., editors, *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1061–1070. ACM.

Hernault, H., Prendinger, H., duVerle, D. A., and Ishizuka, M. (2010). HILDA: A discourse parser using support vector machine classification. *D&D*, 1(3):1–33.

Hirao, T., Yoshida, Y., Nishino, M., Yasuda, N., and Nagata, M. (2013). Single-Document Summarization as a Tree Knapsack Problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1515–1520. ACL.

Hjelmslev, L. (1970). *Language: An Introduction*. University of Wisconsin Press.

Hobbs, J. R. (1979). Coherence and Coreference. *Cognitive Science*, 3(1):67–90.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoey, M. (1983). A tentative map of discourse studies and their place in linguistics. *Analysis, Quaderni di Anglistica*, 1.1:7–26.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In Kim, W., Kohavi, R., Gehrke, J., and DuMouchel, W., editors, *KDD*, pages 168–177. ACM.

James, W. (1884). What is an Emotion? *Mind Association*, 9(34):188–205.

Ji, Y. and Eisenstein, J. (2014). Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 13–24. The Association for Computer Linguistics.

Jindal, N. and Liu, B. (2006a). Identifying comparative sentences in text documents. In Efthimiadis, E. N., Dumais, S. T., Hawking, D., and Järvelin, K., editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 244–251. ACM.

Jindal, N. and Liu, B. (2006b). Mining Comparative Sentences and Relations. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1331–1336. AAAI Press.

Johansson, R. and Moschitti, A. (2010). Syntactic and Semantic Structure for Opinion Expression Detection. In Lapata, M. and Sarkar, A., editors, *Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL 2010, Uppsala, Sweden, July 15-16, 2010*, pages 67–76. ACL.

Joty, S. R., Carenini, G., and Ng, R. T. (2015). CODRA: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Jurek, A., Mulvenna, M. D., and Bi, Y. (2015). Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4(1):9.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1700–1709. ACL.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. In *ACL (1)*, pages 655–665. The Association for Computer Linguistics.

Kennedy, A. and Inkpen, D. (2006). Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125.

Kim, S. and Hovy, E. H. (2004). Determining the Sentiment of Opinions. In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland.*

Kim, S. and Hovy, E. H. (2006). Identifying and Analyzing Judgment Opinions. In Moore, R. C., Bilmes, J. A., Chu-Carroll, J., and Sanderson, M., editors, *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *TACL*, 4:313–327.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014). Sentiment Analysis of Short Informal Texts. *J. Artif. Intell. Res. (JAIR)*, 50:723–762.

Kleinberg, J. M. (1999). Authoritative Sources in a Hyperlinked Environment. *J. ACM*, 46(5):604–632.

Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Kökciyan, N., Çelebi, A., Özgür, A., and Üsküdarlı, S. (2013). BOUNCE: Sentiment Classification in Twitter using Rich Feature Sets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 554–561, Atlanta, Georgia, USA. Association for Computational Linguistics.

Kolchyna, O., Souza, T. T. P., Aste, T., and Treleaven, P. C. (2015). In Quest of Significance: Identifying Types of Twitter Sentiment Events that Predict Spikes in Sales. *CoRR*, abs/1508.03981.

Kouloumpis, E., Wilson, T., and Moore, J. D. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*. The AAAI Press.

Krippendorff, K. (2007). Computing Krippendorff's Alpha Reliability. Technical report, University of Pennsylvania, Annenberg School for Communication.

Lacoste-Julien, S., Jaggi, M., Schmidt, M. W., and Pletscher, P. (2013). Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 53–61. JMLR.org.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, (33):159–174.

Lange, C. G. (18852010). *Om Sindsbevaegelse (1885)*. Kessinger Publishing, LLC, Whitefish.

Lascarides, A. and Asher, N. (2001). Segmented Discourse Representation Theory: Dynamic Semantics with Discourse Structure. *COMPUTING MEANING*, 3.

Li, S. and Zong, C. (2008). Multi-domain Sentiment Classification. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers*, pages 257–260. The Association for Computer Linguistics.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Longacre, R. (1979). The Paragraph as a Grammatical Unit. *Syntax and Semantics*, 12:1–92.

Longacre, R. (1996). *The Grammar of Discourse*. Interdisciplinary Contributions to Archaeology. Springer.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Manning, C. D. (2015). Computational Linguistics and Deep Learning. *Computational Linguistics*, 41(4):701–707.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Marcu, D. (1998). Improving summarization through rhetorical parsing tuning. In Charniak, E., editor, *Sixth Workshop on Very Large Corpora, VLC@COLING/ACL 1998, Montreal, Quebec, Canada, August 15-16, 1998*.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

Martins, A. F. T. and Astudillo, R. F. (2016). From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1614–1623. JMLR.org.

McCallum, A., Freitag, D., and Pereira, F. C. N. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. In Langley, P., editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 591–598. Morgan Kaufmann.

McDonald, R. T., Hannan, K., Neylon, T., Wells, M., and Reynar, J. C. (2007). Structured Models for Fine-to-Coarse Sentiment Analysis. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of ACM*, 38(11):39–41.

Miller, G. A. and Fellbaum, C. (2007). WordNet then and now. *Language Resources and Evaluation*, 41(2):209–214.

Miltsakaki, E., Joshi, A. K., Prasad, R., and Webber, B. L. (2004a). Annotating Discourse Connectives and Their Arguments. In *Proceedings of the Workshop Frontiers in Corpus Annotation@HLT-NAACL 2004, Boston, MA, USA, May 6, 2004*.

Miltsakaki, E., Prasad, R., Joshi, A. K., and Webber, B. L. (2004b). The Penn Discourse Treebank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Mishne, G. (2005). Experiments with mood classification in blog posts. Association for Computational Linguistics.

Mishne, G., Balog, K., de Rijke, M., and Ernsting, B. (2007). MoodViews: Tracking and Searching Mood-Annotated Blog Posts. In Glance, N. S., Nicolov, N., Adar, E., Hurst, M., Liberman, M., and Salvetti, F., editors, *Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, March 26-28, 2007*.

Miura, Y., Sakaki, S., Hattori, K., and Ohkuma, T. (2014). TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Mohammad, S., Dunne, C., and Dorr, B. J. (2009). Generating High-Coverage Semantic Orientation Lexicons From Overtly Marked Words and a Thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 599–608. ACL.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. *CoRR*, abs/1308.6242.

Mohammad, S. M. and Turney, P. D. (2010). Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mukherjee, S. and Bhattacharyya, P. (2012). Feature Specific Sentiment Analysis for Product Reviews. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text*

Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part I, volume 7181 of *Lecture Notes in Computer Science*, pages 475–487. Springer.

Musto, C., Semeraro, G., and Polignano, M. (2014). A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *Information Filtering and Retrieval*, 59.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *SemEval@NAACL-HLT*, pages 1–18. The Association for Computer Linguistics.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Nasukawa, T. and Yi, J. (2003). Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In Gennari, J. H., Porter, B. W., and Gil, Y., editors, *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003), October 23-25, 2003, Sanibel Island, FL, USA*, pages 70–77. ACM.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Ng, V., Dasgupta, S., and Arifin, S. M. N. (2006). Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. In *ACL*. The Association for Computer Linguistics.

Nicholas, N. (1994). Problems in the Application of Rhetorical Structure Theory to Text Generation. Master's thesis, University of Melbourne, Unpublished.

Nielsen, F. Å. (2011). A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. In *#MSM*, volume 718 of *CEUR Workshop Proceedings*, pages 93–98. CEUR-WS.org.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *HLT-NAACL*, pages 380–390. The Association for Computational Linguistics.

Pak, A. and Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S.,

Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Pang, B. and Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Scott, D., Daelemans, W., and Walker, M. A., editors, *ACL*, pages 271–278. ACL.

Pang, B. and Lee, L. (2005). Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Knight, K., Ng, H. T., and Oflazer, K., editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. The Association for Computer Linguistics.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *CoRR*, cs.CL/0205070.

Pearl, J. (1982). Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. In Waltz, D. L., editor, *Proceedings of the National Conference on Artificial Intelligence. Pittsburgh, PA, August 18-20, 1982.*, pages 133–136. AAAI Press.

Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A., and Booth, R. J. (2007). *The development and psychometric properties of LIWC2007. Software manual.* The University of Texas at Austin and The University of Auckland, New Zealand.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). The Edinburgh Twitter Corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, pages 25–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Plato and Bloom, A. D. (1991). *The Republic of Plato*. Basic Books.

Polanyi, L. and Zaenen, A. (2006). Contextual Valence Shifters. In Shanahan, J. G., Qu, Y., and Wiebe, J., editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer.

Popescu, A. and Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics.

Potts, C. (2010). On the negativity of negation. In *Proceedings of Semantics and Linguistic Theory*, volume 20, pages 636–659. CLC Publications.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Prasad, R., Miltsakaki, E., Joshi, A., and Webber, B. (2004). Annotation and Data Mining of the Penn Discourse TreeBank. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, DiscAnnotation '04, pages 88–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Quasthoff, U. (2010). *Deutsches Kollokationswörterbuch*. deGruyter, Berlin, New York.

Rabiner, L. R. (1990). Readings in Speech Recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rabiner, L. R. and Juang, B. H. (1986). An introduction to hidden Markov models. *IEEE ASSp Magazine*.

Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black Box Variational Inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 814–822. JMLR.org.

Rao, D. and Ravichandran, D. (2009). Semi-Supervised Polarity Lexicon Induction. In Lascarides, A., Gardent, C., and Nivre, J., editors, *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 675–682. The Association for Computer Linguistics.

Rao, K., Senior, A. W., and Sak, H. (2016). Flat start training of CD-CTC-SMBR LSTM RNN acoustic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 5405–5409. IEEE.

Read, J. (2005). Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. In Knight, K., Ng, H. T., and Oflazer, K., editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. The Association for Computer Linguistics.

Rehbein, I. (2013). Fine-Grained POS Tagging of German Tweets. In Gurevych, I., Biemann, C., and Zesch, T., editors, *Language Processing and Knowledge in the Web - 25th*

International Conference, GSCL 2013, Darmstadt, Germany, September 25-27, 2013. Proceedings, volume 8105 of *Lecture Notes in Computer Science*, pages 162–175. Springer.

Remus, R., Quasthoff, U., and Heyer, G. (2010). SentiWS - A publicly available German-language resource for sentiment analysis. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the Demonstrations Session, NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 97–101. The Association for Computational Linguistics.

Riloff, E., Wiebe, J., and Wilson, T. (2003). Learning subjective nouns using extraction pattern bootstrapping. In *CoNLL*, pages 25–32. ACL.

Rorty, A. O., editor (1980). *Explaining Emotions*. University of California Press.

Rosenthal, S., Farra, N., and Nakov, P. (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter. In Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S. M., Cer, D. M., and Jurgens, D., editors, *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 502–518. Association for Computational Linguistics.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., and Stoyanov, V. (2015). SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado. Association for Computational Linguistics.

Rosenthal, S., Ritter, A., Nakov, P., and Stoyanov, V. (2014). SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Rouvier, M. (2017). LIA at semeval-2017 task 4: An ensemble of neural networks for sentiment classification. In *SemEval@ACL*, pages 760–765. Association for Computational Linguistics.

Rouvier, M. and Favre, B. (2016). SENSEI-LIF at semeval-2016 task 4: Polarity embedding fusion for robust sentiment analysis. In *SemEval@NAACL-HLT*, pages 202–208. The Association for Computer Linguistics.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of Research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120.

Schachter, S. and Singer, J. E. (1962). Cognitive, social, and physiological determinants of emotional state. *Psychological Review*, 69(5):379–399.

Scheffler, T. (2014). A German Twitter Snapshot. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 2284–2289. European Language Resources Association (ELRA).

Schmid, H. (1995). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the ACL SIGDAT-Workshop*.

Severyn, A. and Moschitti, A. (2015a). On the Automatic Learning of Sentiment Lexicons. In Mihalcea, R., Chai, J. Y., and Sarkar, A., editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1397–1402. The Association for Computational Linguistics.

Severyn, A. and Moschitti, A. (2015b). UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado. Association for Computational Linguistics.

Sidarenka, U. (2016a). PotTS at SemEval-2016 Task 4: Sentiment Analysis of Twitter Using Character-level Convolutional Neural Networks. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 235–242, San Diego, California. Association for Computational Linguistics.

Sidarenka, U. (2016b). PotTS: The Potsdam Twitter Sentiment Corpus. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* European Language Resources Association (ELRA).

Sidarenka, U. (2017). PotTS at GermEval-2017 Task B: Document-Level Polarity Detection Using Hand-Crafted SVM and Deep Bidirectional LSTM Network. In *Proceedings of*

the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback, Berlin, Germany.

Sidarenka, U., Bisping, M., and Stede, M. (2015a). Applying Rhetorical Structure Theory to Twitter Conversations. In *Proceedings of DiSpol 2015*, Saarbrücken, Germany.

Sidarenka, U., Peldszus, A., and Stede, M. (2015b). Discourse Segmentation of German Texts. *JLCL*, 30(1):71–98.

Sidarenka, U., Scheffler, T., and Stede, M. (2013). Rule-Based Normalization of German Twitter Messages. In *Language Processing and Knowledge in the Web - 25th International Conference, GSCL 2013: Proceedings of the workshop Verarbeitung und Annotation von Sprachdaten aus Genres internetbasierter Kommunikation*, Darmstadt, Germany.

Sidarenka, U. and Stede, M. (2016). Generating Sentiment Lexicons for German Twitter. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES 2016)*, Osaka, Japan.

Snyder, B. and Barzilay, R. (2007). Multiple Aspect Ranking Using the Good Grief Algorithm. In Sidner, C. L., Schultz, T., Stone, M., and Zhai, C., editors, *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 300–307. The Association for Computational Linguistics.

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic Compositionality through Recursive Matrix-Vector Spaces. In Tsujii, J., Henderson, J., and Pasca, M., editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1201–1211. ACL.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 151–161. ACL.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Somasundaran, S., Namata, G., Getoor, L., and Wiebe, J. (2009a). Opinion Graphs for Polarity and Discourse Classification. In *Proceedings of the 2009 Workshop on Graph-*

based Methods for Natural Language Processing, August 7, 2009, Singapore, pages 66–74. The Association for Computer Linguistics.

Somasundaran, S., Namata, G., Wiebe, J., and Getoor, L. (2009b). Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 170–179. ACL.

Somasundaran, S., Ruppenhofer, J., and Wiebe, J. (2008a). Discourse Level Opinion Relations: An Annotation Study. In Schlangen, D. and Hockey, B. A., editors, *Proceedings of the SIGDIAL 2008 Workshop, The 9th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 19-20 June 2008, Ohio State University, Columbus, Ohio, USA*, pages 129–137. The Association for Computer Linguistics.

Somasundaran, S., Wiebe, J., and Ruppenhofer, J. (2008b). Discourse Level Opinion Interpretation. In Scott, D. and Uszkoreit, H., editors, *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pages 801–808.

Stede, M. (2011). *Discourse Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Stede, M. and Neumann, A. (2014). Potsdam Commentary Corpus 2.0: Annotation for Discourse Research. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 925–929. European Language Resources Association (ELRA).

Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Taboada, M., Anthony, C., and Voll, K. (2006). Methods for creating semantic orientation dictionaries. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 427–432.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. D., and Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566. The Association for Computer Linguistics.

Takamura, H., Inui, T., and Okumura, M. (2005). Extracting Semantic Orientations of Words using Spin Model. In Knight, K., Ng, H. T., and Oflazer, K., editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. The Association for Computer Linguistics.

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014a). Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In Hajic, J. and Tsujii, J., editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 172–182. ACL.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014b). Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1555–1565. The Association for Computer Linguistics.

Taskar, B., Guestrin, C., and Koller, D. (2003). Max-Margin Markov Networks. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 25–32. MIT Press.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In Jurafsky, D. and Gaussier, É., editors, *EMNLP 2007, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 327–335. ACL.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Trivedi, R. S. and Eisenstein, J. (2013). Discourse Connectors for Latent Subjectivity in Sentiment Analysis. In Vanderwende, L., III, H. D., and Kirchhoff, K., editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 808–813. The Association for Computational Linguistics.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In Cohen, W. W. and Gosling, S., editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press.

Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *ACL*, pages 417–424. ACL.

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346.

van der Maaten, L. and Hinton, G. (2008). Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

van Dijk, T. (1972). *Some aspects of text grammars: A study in theoretical linguistics and poetics*. Janua linguarum: Series maior. Mouton.

van Dijk, T. and Kintsch, W. (1983). *Strategies of discourse comprehension*. Monograph Series. Academic Press.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., and McDonald, R. T. (2010). The viability of web-derived polarity lexicons. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 777–785. The Association for Computational Linguistics.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269.

Vo, D. and Zhang, Y. (2016). Don't Count, Predict! An Automatic Approach to Learning Sentiment Lexicons for Short Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Voll, K. D. and Taboada, M. (2007). Not All Words Are Created Equal: Extracting Semantic Orientation as a Function of Adjective Relevance. In Orgun, M. A. and Thornton, J., editors, *AI 2007: Advances in Artificial Intelligence, 20th Australian Joint Conference on*

Artificial Intelligence, Gold Coast, Australia, December 2-6, 2007, Proceedings, volume 4830 of *Lecture Notes in Computer Science*, pages 337–346. Springer.

Waltinger, U. (2010). GermanPolarityClues: A Lexical Resource for German Sentiment Analysis. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *LREC*. European Language Resources Association.

Wang, F. and Wu, Y. (2013). Exploiting Hierarchical Discourse Structure for Review Sentiment Analysis. In *2013 International Conference on Asian Language Processing, IALP 2013, Urumqi, China, August 17-19, 2013*, pages 121–124. IEEE.

Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015a). Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. *CoRR*, abs/1510.06168.

Wang, X., Liu, Y., Sun, C., Wang, B., and Wang, X. (2015b). Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1343–1353. The Association for Computer Linguistics.

Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to Use t-SNE Effectively. *Distill*.

Wei, W. and Gulla, J. A. (2010). Sentiment Learning on Product Reviews via Sentiment Ontology Tree. In Hajic, J., Carberry, S., and Clark, S., editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 404–413. The Association for Computer Linguistics.

Wiebe, J. (1990). Identifying Subjective Characters in Narrative. In *13th International Conference on Computational Linguistics, COLING 1990, University of Helsinki, Finland, August 20-25, 1990*, pages 401–406.

Wiebe, J. (1994). Tracking Point of View in Narrative. *Computational Linguistics*, 20(2):233–287.

Wiebe, J., Breck, E., Buckley, C., Cardie, C., Davis, P., Fraser, B., Litman, D. J., Pierce, D. R., Riloff, E., Wilson, T., Day, D. S., and Maybury, M. T. (2003). Recognizing and Organizing Opinions Expressed in the World Press. In Maybury, M. T., editor, *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 12–19. AAAI Press.

Wiebe, J., Bruce, R. F., and O'Hara, T. P. (1999). Development and Use of a Gold-Standard Data Set for Subjectivity Classifications. In Dale, R. and Church, K. W., editors, *27th*

Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999. ACL.

Wiebe, J. and Riloff, E. (2005). Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *CICLing*, volume 3406 of *Lecture Notes in Computer Science*, pages 486–497. Springer.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wilson, T. (2007). *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of private states.* PhD thesis, Intelligent Systems Program, University of Pittsburgh.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics.

Winter, E. (1977). A Clause-Relational Approach to English Texts. *Instructional Science*, 6(1):1–92.

Wojatzki, M., Ruppert, E., Holschneider, S., Zesch, T., and Biemann, C. (2017). GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, Berlin, Germany.

Wolf, F. and Gibson, E. (2005). Representing Discourse Coherence: A Corpus-Based Study. *Comput. Linguist.*, 31(2):249–288.

Xu, S., Liang, H., and Baldwin, T. (2016). UNIMELB at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In *SemEval@NAACL-HLT*, pages 183–189. The Association for Computer Linguistics.

Yang, Y. and Eisenstein, J. (2017). Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics (TACL)*, in press.

Yessenalina, A. and Cardie, C. (2011). Compositional Matrix-Space Models for Sentiment Analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 172–182. ACL.

Yessenalina, A., Yue, Y., and Cardie, C. (2010). Multi-Level Structured Models for Document-Level Sentiment Classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1046–1056. ACL.

Yoshida, Y., Suzuki, J., Hirao, T., and Nagata, M. (2014). Dependency-based Discourse Parser for Single-Document Summarization. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1834–1839. ACL.

Yu, C. J. and Joachims, T. (2009). Learning Structural SVMs with Latent Variables. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 1169–1176. ACM.

Yu, H. and Hatzivassiloglou, V. (2003). Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.

Zhou, L., Li, B., Gao, W., Wei, Z., and Wong, K. (2011). Unsupervised Discovery of Discourse Relations for Eliminating Intra-sentence Polarity Ambiguities. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 162–171. ACL.

Zhu, X. and Ghahramani, Z. (2002). Learning from Labeled and Unlabeled Data with Label Propagation, CMU-CALD-02-107. Technical report, Carnegie Mellon University.

Zirn, C., Niepert, M., Stuckenschmidt, H., and Strube, M. (2011). Fine-Grained Sentiment Analysis with Structural Features. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 336–344. The Association for Computer Linguistics.