# EXPLOITING STRUCTURE FOR SENTIMENT CLASSIFICATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ainur Uskemenovna Yessenalina

August 2012

EXPLOITING STRUCTURE FOR SENTIMENT CLASSIFICATION

Ainur Uskemenovna Yessenalina, Ph.D.

Cornell University 2012

This thesis studies the problem of sentiment classification at both the document and sentence level using statistical learning methods. In particular, we develop computational models that capture useful structure-based intuitions for solving each task, treating the intuitions as latent representations to be discovered and exploited during learning.

For document-level sentiment classification, we exploit structure in the form of informative sentences — those sentences that exhibit the same sentiment as the document, thus explain or support the document's sentiment label. We first show that incorporating automatically discovered informative sentences in the form of additional constraints for the learner improves performance on the document-level sentiment classification task. Next, we explore joint structured models for this task: our final proposed model does not need sentence-level sentiment labels, and directly optimizes document classification accuracy using inferred sentence-level information. Our empirical evaluation on two publicly available datasets shows improved performance over strong baselines.

For phrase-level sentiment classification, we investigate the compositional linguistic structure of phrases. We investigate compositional matrix-space models, learning matrix-space word representations and modeling composition as matrix multiplication. Using a publicly available dataset, we show that the matrix-space model outperforms the standard bag-of-words model for the phrase-level sentiment classification task.

## BIOGRAPHICAL SKETCH

Ainur Yessenalina is originally from Almaty, Kazakhstan. She received a Specialist degree (B.S) with Honours in Applied Mathematics and Informatics from Lomonosov Moscow State University, and Ph.D. degree in Computer Science from Cornell University.

To my family.

## ACKNOWLEDGMENTS

I am grateful to my advisor Claire Cardie for the great advice and support she provided. I gratefully acknowledge the advice from my committee members John Hale and John Hopcroft. Interactions with Cornell professors were beneficial for my professional development, special thanks to professors David Bindel, Rich Caruana, Thorsten Joachims, Lillian Lee, and Emin Gün Sirer. Insightful discussions with other students at Cornell were valuable for me, special thanks to Yejin Choi, Nikos Karampatziakis and Yisong Yue.

I am very grateful to my family and my dear friends, who supported me in this journey. I thank Nikos for inspiration and support through these years.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

CHAPTER 1

# INTRODUCTION

Understanding opinions, attitudes, emotions of people towards objects and towards each other is important for making decisions (e.g., which movie to watch, which camera to buy) and building a mental picture of the world and interpersonal relationships (e.g., who likes whom, who likes what). *Sentiment analysis* is a research area of natural language processing that aims to understand these types of private mental and emotional states in text.

There has been a great burst of interest in sentiment analysis research in recent years (Pang and Lee (2008), Liu (2012)). This interest is due to a number of factors. First, and most importantly, there are numerous practical applications of opinion analysis for corporate business intelligence, political analysis and personal decision-making. Second, there have been developments in machine learning and statistical natural language processing that allow for proper computational treatment of these real-world application scenarios. Third, since people often post their opinions on forums and various websites and access to user-generated content has become easier, it is now possible to create datasets to support the study of sentiment analysis using statistical methods.

Research in sentiment analysis can be roughly split into two main threads: coarse-grained sentiment analysis (e.g., Pang et al. (2002), Turney (2002)) and fine-grained sentiment analysis (e.g., Wiebe et al. (2005)). Coarse-grained sentiment analysis is concerned with analysing the overall sentiment of a document or larger snippet of text, while fine-grained sentiment analysis studies opinions at the sentence or phrase level, identifying the polarity, topic, and sometimes even the source of the opinion.

*Sentiment classification*, one of the subtasks of sentiment analysis, identifies whether a given text carries positive or negative polarity. This thesis studies the task of supervised sentiment classification at both the document and sentence level. In document-level sentiment classification (or categorization), we are given a set of documents with document-level sentiment labels, the goal is to learn a statistical model, so that it can predict the sentiment labels for previously unseen documents. The same question of sentiment categorization could be posed at a much finer level as well — for example, at the sentence or phrase level. Though the question is the same, the intuitions for figuring out the answer in each of these settings could be different. For example, the sentiment of a document generally revolves around a few phrases and sentences that are subjective and that express the feelings of the author towards the topic under consideration, while the rest of the document might describe the topic itself. Thus, correctly identifying phrases and sentences that explain the document's sentiment label might improve document-level sentiment classification. In contrast, deciding the sentiment of a phrase requires accounting for more of the linguistic structure of the phrase: since phrases usually have many fewer words than documents, the exact way the words combined matters a lot.

As a result, *in this thesis we develop computational models that capture useful intuitions for solving document-level and phrase-level sentiment classification. We furthermore treat these intuitions as latent representations to be discovered and exploited during learning.*

For document-level sentiment classification, for example, our proposed methods rely on the intuition outlined above — that the overall sentiment is based on just a subset of the sentences or phrases in the document, i.e., the sentiment-bearing sentences or phrases. For phrase-level classification, our pro-

posed approach relies instead on the principle of semantic compositionality, which states that the meaning of a phrase is composed from the meaning of its words via a set of rules that combine them (Frege (1892), Montague (1974), Dowty et al. (1981)). We describe our approach to, and the intuitions behind, each in the sections below.

## 1.1 Document-Level Sentiment Classification: Beyond Conventional Models

A conventional approach to the document-level sentiment classification task is to treat it as a standard text categorization task: use a bag-of-words representation that treats all words in the document equally, and, thus, does not account for the structure of the document; and then apply off-the-shelf machine learning classifiers such as SVMs or Naive Bayes to create the sentiment classifier (Pang et al. (2002)). Now the natural question of interest is: *How can we do better than that?* To improve the performance of the document-level sentiment classifier, one option is to *exploit the structure of the document*. In particular, documents are comprised of sentences, and sentences, of phrases. And as discussed earlier, the sentiment of a document usually revolves around just a few sentences or even a few phrases that express the overall sentiment of the writer. We can see from the example in Table 1.1 that in a positive movie review not all sentences are sentiment-bearing (or subjective); instead, subjective sentences are interleaved with objective sentences that describe, for example, the plot. Moreover, negative sentences could be present as well.

The idea of using sentence-level information to aid document-level sentiment classification was first proposed by Pang and Lee (2004), who suggested

Table 1.1: Example of a positive movie review from the Movie Reviews dataset split by sentences. Positive sentences are denoted in bold, negative sentences are in italics.

| |
|---|
| **"Being John Malkovich" is the type of film we need to see more.** *Today's films are either blockbusters that entertain us with tiresome formula, or those that have similar themes.* **Malkovich falls under none of these categories , and it's quite refreshing to see that occur.** This strangely provoking story, is actually somewhat understandable. John Cusack plays a puppeteer trying to make it to the big time. His wife (Cameron Diaz) supports the both of them by working at a petstore, which explains the obscure pets they keep in their apartment.<br>...<br>I don't want to give away too much, but Cusack becomes too attached with his discovery. **In my opinion, this idea is absolutely brilliant.** *It's really quite scary to think that someone could become you, control you, be you.* It makes you wonder why we act like we do , and why sometimes we blurt out things or act out something out of the blue.<br>...<br>**"Being John Malkovich" isn't an excellent film, but it is definitly entertaining and will eaisly become a cult favorite. What's even better is the film's puzzling message...** am I Nick Lyons ? |

filtering out objective sentences and keeping only the *subjective* ones and then using the resulting subjective extracts as the training set for the classifier. However, the authors note that unless the filtering step is done carefully, the resulting system does not always lead to improved performance. Zaidan et al. (2007) instead proposed relying on *annotator rationales* — text segments identified by human annotators that support or explain the annotator's decision about the sentiment label of the document. The authors subsequently modify the machine learning algorithm to make use of annotator rationales. In this thesis we rely on *automatically* identified rationales, which we refer to as the **informative sentences** in the sense that they exhibit the same sentiment as the document. Thus, these sentences are not only subjective, but also support the *sentiment* la-

bel (positive or negative) of the document, i.e., for documents with overall positive sentiment, sentences that exhibit positive sentiment are considered informative; for documents with overall negative sentiment, sentences that exhibit negative sentiment are considered informative.

In this thesis we use *the set of informative sentences as latent structure for the document-level sentiment classification task* and investigate two ways to incorporate them into a sentiment classifier: first, as additional constraints for an SVM classifier, and, second, as latent variables in a two-level joint structured model. We provide a high-level description of each approach below.

**Incorporating informative sentences as additional constraints.** Zaidan et al. (2007) show that the use of manually annotated informative snippets of text can improve the performance of document-level sentiment classifiers. They incorporate the informative text segments as constraints for the SVM learner that aim to ensure that the resulting classifier is less confident in its classification of training documents that have the annotator rationales removed vs. the original document (that contained the rationales). The addition of these constraints leads to performance gains over an SVM learner without the constraints. Of course, obtaining the text snippets that support the sentiment label requires more time from human annotators than simply labeling the document as positive or negative. Ideally, we would like to have the best of both worlds: improved performance and a small annotation effort.

**Contribution 1.** In this thesis we investigate ways to use available sentiment analysis resources to automatically discover informative sentences for document-level sentiment classification. As in Zaidan et al. (2007) we incorporate them in the learning procedure in the form of additional constraints for an SVM classifier. Empirical results on a standard movie review corpus indicate that the automatically discovered informative sentences are just as helpful as human rationales. Furthermore, using both the human annotator rationales and automatically discovered informative sentences boosts performance even further for this domain.

**Using informative sentences in a joint structured model.** Though automatically discovered informative sentences are cheap to obtain, they are less than perfect: the automatic methods can miss some informative sentences or mistakenly include non-informative ones. Therefore, one option is to develop statistical models that jointly learn to predict both the sentiment of the document and the set of informative sentences, thus controlling the error propagation due to noisy sentence-level labels (Tsochantaridis et al. (2004), Yu and Joachims (2009)).

**Contribution 2.** In this thesis, we also investigate structured models for document-level sentiment classification. We introduce a two-level joint approach for document-level sentiment classification that simultaneously extracts informative sentences and predicts document-level sentiment based on the extracted sentences. The proposed approach (1) does not rely on gold standard sentence-level subjectivity annotations (which may be expensive to obtain), and (2) optimizes directly for document-level performance. Empirical evaluations on movie reviews and U.S. Congressional floor debates show improved performance over previous approaches.

## 1.2 Phrase-level Sentiment Classification

*Phrase-level sentiment classification* is the task of identifying the polarity of a phrase. It is an important step in systems that aim to summarize the opinions of people or entities toward each other or toward a particular topic as expressed throughout a document or a corpus (Stoyanov and Cardie (2011)). It is also important in the analysis of social media sources, where the sentiment expressed in short sentences or phrases (e.g., from Twitter) has been used to predict the results of political polls (O'Connor et al. (2010)) and find the sentiment w.r.t. various topics (Jiang et al. (2011)).

Though the task of phrase-level sentiment classification somewhat resembles document-level sentiment classification, there are certain differences between the two. The biggest difference is the length of the text: a phrase contains substantially fewer words than a document. As a result, we go about interpreting its sentiment differently: for example, instead of glancing through the document in a search of informative phrases or sentences that might explain the label of the document, one instead should try to understand the meaning of the phrase by looking at the how the words are combined. Thus, the computational treatment of this task will rely on intuitions from the linguistic study of compositional semantics.

### 1.2.1 Semantic Compositionality for Sentiment Analysis

The *semantic compositionality principle* states that the meaning of a phrase is composed from the meaning of its words and the rules that combine them. A key effect of semantic compositionality in the context of sentiment analysis is a polarity change (e.g., flip, increase, decrease) when combining one word with other

words. Consider the following examples:

- prevent war
- limiting freedom
- absolutely delicious

In all of these phrases we observe changes in sentiment w.r.t. underlined word when the preceding word is considered. In the first example, "war" has a negative sentiment; however, the word "prevent" essentially flips the polarity of the phrase to positive (i.e., preventing war is good). In the second, "freedom" has positive sentiment; however, "limiting freedom" makes the resulting sentiment of the phrase negative. And in the final third example, the presence of the adverb "absolutely" strengthens the already positive sentiment of "delicious". The bottom line is that the computation of phrase-level sentiment follows compositional rules.

### 1.2.2 Modeling Semantic Compositionality

According to the semantic compositionality principle in the context of sentiment analysis, the sentiment of a phrase depends on the *sentiment of the words used in the phrase* and the *rules to combine them*. The sentiment of individual words could be determined by using a *sentiment lexicon* (Wilson et al. (2005b)) — a list of words with their corresponding sentiment. The next question is: *What are these compositional rules?* One might look at a number of sentiment-bearing phrases and provide a set of hand-written compositional rules for a sentiment analysis system, similar to Choi and Cardie (2008). However, writing the rules by hand could be a tedious process. For example, to obtain a set of rules such as "IF the

syntactic pattern is 'VB NP' and the verb is 'prevent' and noun phrase has a negative sentiment, THEN the resulting sentiment of a phrase is positive", one has to consider various syntactic patterns and observe how the resulting sentiment changes when composing with certain lexical items.

**Contribution 3.** In this thesis we develop *a model that learns latent semantic representations for words and is compositional: each word is represented by a matrix and the composition of words is modeled as matrix multiplication.* Thus, there is no need to hand-write the compositional rules: combinations of words are represented as the successive multiplication of the matrix corresponding to each word with that of its successor. Each word itself acts as a linear operator. We present an algorithm for learning matrix-space word representations for semantic composition from sentiment-labeled phrases. The empirical results indicate statistically significant improvements in performance over a bag-of-words model for the phrase-level classification task.

## 1.3 Structure of the Thesis

The rest of the thesis is organized as follows. In Chapter 2 we discuss related work in the sentiment analysis area, focusing on document-level and phrase-level classification tasks. Chapter 3 describes our work on automatically discovering and employing informative sentences for document-level sentiment classification in the form of additional constraints to the learner. Chapter 4 then introduces a joint structured model for document-level sentiment classification. In Chapter 5 we propose the compositional matrix-space model for phrase-level classification. Finally, we conclude and summarize our work in Chapter 6.

CHAPTER 2

# RELATED WORK

In this chapter we give an overview of research done in sentiment analysis and opinion mining. In particular, we begin the chapter with an overview of the sentiment analysis area and then describe prior research in sentiment classification both at the document level and at the sentence level. We specifically focus on research that is directly related to the contributions of this thesis. We further discuss (compare and contrast) the related work in the context of thesis contributions in the appropriate chapters.

## 2.1 Overview of Sentiment Analysis

As we mentioned in Chapter 1, there has been a great interest in recent years in the sentiment analysis area. Pang and Lee (2008) provide a very insightful and comprehensive overview of the area. Liu (2012) provides a more recent survey as well as introductory text about the area. Both surveys trace the very first work on detecting opinions and sentiment using statistical methods to be around 2001: Wiebe (2000), Das and Chen (2001), Tateishi et al. (2001), Tong (2001), Morinaga et al. (2002), Pang et al. (2002), Turney (2002). We briefly describe the early work in the area according to the domains and/or genres of the texts under study.

**Reviews.** A lot of research in sentiment analysis and opinion mining area has been done on movie and product reviews (e.g., Pang et al. (2002), Dave et al. (2003), Hu and Liu (2004b), Blitzer et al. (2007)). As discussed in Chapter 1, the burst of interest in analysing product reviews is in part due to the creation of *publicly available sentiment-labeled review datasets*. These datasets were created

using websites such as Amazon.com, Imdb.com, Epinions.com, etc., whose interface for writing a review typically requires a user to provide a star-rating as well as the textual description of the user's opinion about a certain product or movie. The star ratings are normally used during system development to infer whether the given review is positive or negative (Pang et al. (2002)). In this thesis we use publicly available sentiment-labeled corpora to evaluate the statistical methods that we develop.

Some of the work on product reviews falls into the category of coarse-grained sentiment analysis (e.g., Pang et al. (2002), Blitzer et al. (2007)) and tries to answer the question "is the review for the product positive or negative?". But there is a great body of research on product reviews that is trying to answer more fine-grained questions such as "what feature(s) of the product do customers like?", "what feature(s) of the product do customers not like?" (e.g., Hu and Liu (2004b), Mei et al. (2007), Snyder and Barzilay (2007), Titov and McDonald (2008)). This interest in fine-grained sentiment analysis of product reviews lead to the rapid formation of a sub-area of sentiment analysis called *aspect-based (or feature-based) sentiment analysis* (Liu (2012)). One of the crucial assumptions in this line of work is that the product is known in advance and that there are a few important aspects, or facets, of the product, that are also sometimes known in advance (e.g., Hu and Liu (2004b), Snyder and Barzilay (2007), Titov and McDonald (2008)).

Changing the domain of a product review might cause a performance drop: a classifier trained on movie reviews might not do well on reviews of kitchen appliances. Due to influential work by Blitzer et al. (2007), sentiment classification became one of the attractive tasks for developing *domain adaptation* algorithms (e.g., Glorot et al. (2011)). In this thesis we will not focus on the domain adap-

tation problem; however, we will use the product review datasets provided by Blitzer et al. (2007) for some of our experiments.

**Newswire.** In around 2003 a different group of researchers started thinking about other real-world needs and scenarios that involve fine-grained sentiment analysis (e.g., Cardie et al. (2003), Wiebe et al. (2003), Bethard et al. (2004), Stoyanov et al. (2005)). Researchers started looking at the newswire domain, where one wants to extract the opinions in each story, including the identification of the opinion holder, the opinion expression itself, its polarity, and the topic/target of the opinion. The early efforts in facilitating research in fine-grained sentiment analysis resulted in the creation of an annotation scheme together with the sentiment-annotated MPQA (Multi-Perspective Question Answering) dataset (Wiebe et al. (2005)). This became a test-bed for fine-grained sentiment analysis tasks and provided a framework for developing statistical methods for those tasks (e.g., Choi et al. (2005), Breck et al. (2007), Nakagawa et al. (2010), Stoyanov and Cardie (2011), Johansson and Moschitti (2011)). Furthermore, the output of fine-grained sentiment analysis system could be used to construct aggregate opinion summaries (Stoyanov (2009)). This type of summary is crucial for applications such as opinion-oriented question answering, for example finding the answers for the queries of the form: *"What is X's opinion toward Y?"* or *"What do people think about Z?"* (Stoyanov et al. (2005), Somasundaran et al. (2007)).

Fine-grained sentiment analysis for the newswire articles domain poses different challenges compared to product reviews domain (e.g., Bethard et al. (2004), Kim and Hovy (2005), Wiebe et al. (2005), Stoyanov et al. (2005), Choi et al. (2006), Somasundaran et al. (2007)). First, the opinion topic in case of product reviews is a pre-specified product, for example a movie or a camera. In contrast, a news article could be covering *any* event. Therefore, the lexical items

and syntactic structures and patterns involved in identifying and characterizing opinions in newswire domain vary substantially. Also the opinion topic could be changing throughout the news article and multiple topics could be discussed in the same article (Stoyanov and Cardie (2008)). Second, in contrast with the product reviews where the user usually expresses his/her opinion about a product, a news article could be covering a political event, where the opinions are expressed by multiple opinion sources (opinion holders) (Choi et al. (2005), Kim and Hovy (2005)).

**Political debates.** Another interesting domain for sentiment analysis proposed by work of Thomas et al. (2006) is Congressional floor debates. The speaker for each Congressional floor-debate speech is known, the votes of all speakers for the bills under discussion are known too; therefore the sentiment of the speaker towards the bill under discussion could be inferred assuming that the speaker's speech is motivated by his/her vote. This provides a simple means for obtaining speech-level gold standard sentiment labels. The NLP task, then, is to predict that sentiment of each speech towards the bill under discussion given its transcript. To do this, Thomas et al. (2006) exploit an agreement structure between speeches using minimum cuts. The graph is constructed as follows: the speeches are the nodes, and the edges model the same-speaker constraints and the agreements between different speakers. The authors train two classifiers: one is to predict the speech labels in isolation, and the other — to predict the agreement weights. A high agreement weight between different speeches encourages the assignment of the same label to those speeches. Inference is performed by finding the minimum cut which partitions the speeches in two groups: support or oppose. The experimental results show improved performance over a model that does not take into account agreement information.

In follow-up work, Bansal et al. (2008) take into account information about both *agreement and disagreement between speakers*. In our work, we use the Congressional floor-debates dataset for evaluation; however, we exploit a different and complementary structure in the form of informative sentences.

**Other domains.** There are a few other domains in which sentiment analysis has been applied. Niu et al. (2005) study the sentiment analysis problem for the medical domain, predicting outcomes for patients. Sentiment analysis has also been applied to conversations (e.g., Murray and Carenini (2009), Wang and Liu (2011), Murray and Carenini (2011)), blogs (e.g., Chesley et al. (2006), Kale et al. (2007), Godbole et al. (2007), Bautin et al. (2008)), financial news and reports (e.g., Das et al. (2005), Devitt and Ahmad (2007)).

## 2.2 Document-Level Sentiment Classification

Some of the pioneering work on sentiment classification started by tackling the document-level sentiment classification task (Turney (2002), Pang et al. (2002)). The two main approaches to this task are: (1) lexicon-based (e.g., Turney (2002), Hu and Liu (2004a)); and (2) machine learning based (e.g., Pang et al. (2002), Mao and Lebanon (2006), McDonald et al. (2007)). Real-world commercial systems use a hybrid approach that combines (1) and (2) (e.g., Blair-Goldensohn et al. (2008)).

Seminal work by Turney (2002) develops a lexicon-based method for *unsupervised document-level sentiment classification*. The method developed in that paper first identifies phrases with adjectives or adverbs in a review and then assigns a sentiment label to a review based on the average *semantic orientation* of those phrases. Semantic orientation of a phrase represents whether the phrase

is semantically associated with positive or negative words. Turney proposes to calculate it as a difference between the PMI-IR (Pointwise Mutual Information defined by search engine) w.r.t a known positive word ("excellent") and the PMI-IR w.r.t a known negative word ("poor"). The evaluation is performed on 410 reviews from Epinions.com from four different domains: automobiles, banks, movies and travel destinations. The average accuracy that is achieved by the proposed method is 74%, varying from 66% to 84%, depending on the domain. Other work that computes the semantic orientation of words includes Turney and Littman (2003), Takamura et al. (2005), etc.

Work by Pang et al. (2002) was the first to consider the task of supervised document-level sentiment classification. They start by creating a sentiment-labeled dataset of movie reviews, which facilitated research in the sentiment analysis area. The movie reviews were obtained from the Imdb website and automatically labeled as "thumbs up" (positive) or "thumbs down" (negative), by utilizing heuristic rules based on the user assigned star rating; the reviews with mixed sentiment, i.e., with the rating at the middle of the star rating scale were skipped. Then the authors employ various supervised machine learning methods such as Naive Bayes, maximum entropy and support-vector machines and use bag-of-features representation of the documents, with features such as unigrams, bigrams, trigrams, unigrams with part-of-speech tags, etc. Pang et al. (2002) conclude that the support-vector machine classifier that uses unigram features works the best, achieving performance of around 83% accuracy and outperforming the human-produced baselines.

Back in 2002, Pang et al. (2002) as well as Turney (2002) note that the task of classifying reviews is hard, compared to topic-based text categorization, since the reviews might contain a "thwarted expectations" narrative, when the author

uses a contrast to previous discussions (example from Pang et al. (2002)):

> "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."

The existence of words like "great", "brilliant", "good", in this review passage could be misleading for the machine learning classifier that uses bag-of-words representation of a document. As a result, Pang et al. (2002) hypothesize that a document-level sentiment classifier potentially could benefit from sentence-level analysis, which would identify whether the author is expressing his/her opinion about the topic of interest or not. Thus, the subsequent work on the document-level sentiment classification task tries various ways of incorporating the sentence-level information in the final model. *We will describe it in detail in Section 2.2.1 and 2.2.2, since it is directly related to the contributions described in Chapter 3 and Chapter 4.*

### 2.2.1 Document-Level Sentiment Classification: Using Sentence-Level Information

Pang and Lee (2004) noted that an opinionated text usually consists of evaluative sub-parts (phrases or sentences) that express the sentiment of the author towards a topic of interest, as well as non-evaluative sub-parts. Consider the following example sentence (Pang and Lee (2004)):

> The protagonist tries to protect her good name.

The presence of the word "good" does not tell us anything about the author's

sentiment towards the topic (movie); it simply states the fact and potentially could be a part of a negative movie review.

Pang and Lee (2004) address this problem by trying to operate only on the subjective sentences from a movie review and, thus, reduce the influence of potentially objective sentences for the document-level sentiment classification. They employ a two-step approach: (1) they label the sentences in the document as subjective vs. objective using minimum cuts; (2) apply machine-learning techniques to the subjective extracts. For step (1), the authors first classify the sentences in a document as subjective or objective using a classifier trained on an automatically labeled set of subjective-objective sentences (from plot summaries and reviews, respectively) and then construct a graph. The nodes of this graph are the sentences and the edges are determined by the proximity of sentences to each other. They further use a minimum cut algorithm which assumes that subjective sentences, as well as objective ones, are usually grouped together, transitions between subjective and objective and vice-versa are preferable when the classifier is confident on the sentence label. The experimental results show that when employing a Naive Bayes classifier, the use of subjectivity extracts leads to improved accuracy compared to the full reviews from 83% to 86%, but for the SVM classifier there is no significant difference. So, the resulting set of subjective sentences can be noisy, providing less than ideal support for the document-level sentiment categorization.

Another notable work by Mao and Lebanon (2006) incorporates the discourse structure of the document into the statistical model. Their approach operates in two stages. First, given the training data with ordinal sentence-level sentiment labels the isotonic conditional random fields model is learned, so that given an unseen document from the test set, it can predict the sequence of

sentence-level sentiment labels. Next, this sequence is converted to a *local sentiment flow*, which is in essence a smooth length-normalized representation of the whole document. Finally, the authors use the local sentiment flow representation of the document for k-nearest neighbour classification. The experimental results on small set of 249 movie reviews show that the sentiment flow representation outperforms the bag-of-words representation. The advantage of their proposed model is that it outperforms a conventional model for the document-level sentiment prediction. However, their model needs sentence-level sentiment annotations, obtaining which requires significant human annotator effort. The other drawback is that the model works in two stages and there is no feedback between those two stages to control the error propagation from the first stage to the second one.

The work by McDonald et al. (2007) proposed a joint structured fine-to-coarse model for document-level sentiment classification. The proposed model is a graphical model that has a linear chain of sentence-level sentiment variables each of which is connected to the respective sentence, as well as an additional variable that represents the document-level sentiment. The document-level variable is connected to all sentence-level sentiment variables. The authors used the MIRA learning algorithm (Crammer and Singer (2003)) for training their joint structured model and the predictions were made using an algorithm based on Viterbi inference. The advantage of the presented model is that training of the model is done jointly, i.e., the sentence-level decisions affect the document-level decisions and vice versa. However, the proposed method optimizes a loss function that is composed of sentence-level and document-level parts, which can potentially hurt document-level performance in order to compensate for poor sentence-level performance. Also the proposed model requires

training data with sentence-level and document-level sentiment labels, which might be expensive to acquire.

### 2.2.2 Annotator Rationales

The notion of *"annotator rationales"* was first introduced by work of Zaidan et al. (2007). The key idea proposed in that work was to use a different kind of training label: document-level sentiment annotation with "rationales". *The annotator rationales are in essence those text spans, highlighted by the human annotators in support of (i.e., as an explanation of) the sentiment label they select for a document as a whole.* Since these annotator rationales can provide performance gains only if incorporated somehow into a learning framework capable of exploiting them, Zaidan et al. (2007) propose to modify a standard SVM classifier by incorporating the annotator rationales in the form of additional constraints for the learning procedure. The role of the additional constraints is to ensure that the final classifier is more confident in a document from the training set than the same document but without the rationales (i.e., the rationale text spans are deleted). The authors conduct an extensive annotation study of rationales by providing the annotation guidelines and exploring inter-annotator agreements, and conclude that annotator rationales can be helpful regardless of the fact that different annotators might identify different text spans as supporting the document label; the number of annotator rationales could vary as well. The authors argue that their framework is useful for all text categorization domains, however they only consider the sentiment categorization task for evaluation. In this thesis, we use the proposed approach to handle automatically discovered rationales (Chapter 3).

In their follow-up work Zaidan and Eisner (2008) propose a generative model for learning from annotator rationales, that aims not only to predict

document-level sentiment but also to generate rationales. Though their approach learns to predict the document label and the rationales jointly, it assumes that the human annotator is consistent in marking rationales.

**Annotator rationales for other tasks**. Interestingly, there has been some research in using annotator rationales in computer vision (Donahue and Grauman (2011)), more specifically for the image classification task.

## 2.3 Phrase-level Sentiment Classification

In this section we will discuss work on the phrase-level sentiment classification task. Since a lot of work on computing phrase-level sentiment relies on word-level sentiment, we start with a brief overview of research on constructing sentiment lexica — lists of words with their corresponding polarities — and then continue with the discussion of previous approaches to phrase-level sentiment classification.

**Sentiment lexica.** There has been a lot of research in determining the sentiment of words and constructing sentiment dictionaries (e.g., Hatzivassiloglou and McKeown (1997), Turney and Littman (2003), Rao and Ravichandran (2009), Mohammad et al. (2009), Velikovich et al. (2010)). Some of the proposed approaches for sentiment lexicon construction (e.g., Wilson et al. (2005b), Esuli and Sebastiani (2006)) rely on manually built lexical resources such as General Inquirer (Stone et al. (1966))[1], WordNet (Fellbaum (1998)), etc. Mohammad et al. (2009) constructed a high-coverage sentiment lexicon using a Roget-like thesaurus and affix rules. Other approaches to sentiment lexicon induction build it from corpora typically using machine learning methods and starting with

[1] `http://www.wjh.harvard.edu/~inquirer/`

a small set of seed words with known sentiments (e.g., Hatzivassiloglou and McKeown (1997), Turney and Littman (2003), Rao and Ravichandran (2009), Velikovich et al. (2010)).

**Sentiment of a phrase: accounting for word composition.** Work by Polanyi and Zaenen (2004) was one of the first in computational linguistics to point out that local interactions between words are very important for identifying the sentiment of a text snippet, while most previous research (e.g., Hatzivassiloglou and McKeown (1997), Turney and Littman (2003)) focused on identifying sentiment overall sentiment by considering individual lexical items. Polanyi and Zaenen (2004) considered various lexical phenomena that can change the valence of lexical items, such as sentence-based and discourse-based contextual valence shifters, and proposed a way of calculating the final sentiment of the text that accounts for local interactions. The authors showed on a few examples that their proposed method can lead to improved performance over simple counting of positive and negative words.

Accounting for local interactions between lexical items in automatic sentiment classifiers was done by using features such as bigrams, trigams, etc. as well as features derived by from syntactic or semantic patterns (e.g., Kennedy and Inkpen (2006), Shaikh et al. (2007), Wilson et al. (2005b)). One of the drawbacks of these methods is that they heavily rely on heuristically defined interactions and sentiment lexica. Another drawback is that though these models can account for certain word interactions, the final model still uses the flat bag-of-words feature representation and thus the structural nature of the interactions may not be accounted for.

In their influential work "Sentiment composition", Moilanen and Pulman (2007) proposed to account for the structural nature of word composition.

The authors rely on the Principle of Compositionality (Frege (1892), Montague (1974), Dowty et al. (1981)) to compute the sentiment of a phrase. They develop rules based on syntactic dependency parse trees to compute the resulting sentiment of a phrase or a sentence in a bottom-to-the-top manner by starting from the sentiments of the individual lexical items and subsequently computing sentiment values in the intermediate nodes of the dependency tree and, finally, in the root. While the proposed method accounts for the structural nature of interactions, the rules used in their system are hand-written, which might require a significant effort and time of a domain expert to develop.

Choi and Cardie (2008) proposed a learning-based method for binary phrase-level sentiment classification that is also based on ideas from compositional semantics. The developed model starts with a sentiment lexicon that contains prior (out of context) polarities of words; a set of hand-written compositional rules; and a set of sentiment-labeled phrases. The proposed feature-based algorithm with compositional inference identifies the sentiment of the phrase by learning the appropriate assignments of intermediate hidden variables given the features, including the prior polarities of lexical items in a phrase and the phrase-level sentiment label. Though their method can account for complex structural interactions in a phrase or sentence, it, too, relies on a set of hand-written rules.

Nakagawa et al. (2010) introduced a learning-based model that uses compositional inference similar to Choi and Cardie (2008), but also learns rules for sentiment composition from the data. The authors proposed the Tree-CRF model — a model that uses conditional random fields (CRFs) with hidden variables, where the structure of the model is defined by the dependency tree of a phrase or a sentence. The polarities of the dependency sub-trees are represented as

hidden variables, since they are not observed at the training time, the only observed sentiment label is a phrase-level sentiment. The advantage of the model proposed by Nakagawa et al. (2010), is that it does not require hand-written rules for sentiment composition and can learn the rules from the data; however, it heavily relies on sentiment lexica and various carefully hand-crafted features, which might be expensive to compute during inference.

**Different levels of sentiment.** Though most of the work in the sentiment analysis area has considered binary sentiment labels (positive and negative), in real-world settings, sentiment values stretch out across a polarity spectrum. Some of the previous work (e.g., Pang and Lee (2005), Goldberg and Zhu (2006)) considered the task of predicting star ratings at the document level. Wilson et al. (2004) tackles the problem of classifying phrases from the MPQA dataset according to their subjective strength but not polarity. In this thesis, we propose to use a single ordinal sentiment scale that combines both the polarity and the strength annotations from the MPQA corpus.

Work by Liu and Seneff (2009) considers the task of classifying the reviews on a five-level sentiment scale. It models the compositional effects of combining adverbs, adjectives and negators. The authors suggest ways of computing the sentiment of adjectives from data; and compute the effect of combining an adjective with an adverb as a multiplicative effect; and the effect of combining adjective with a negator as an additive effect. The proposed method requires the knowledge of part-of-speech tag for each word, the list of negators (since the negator is an adverb as well), and it models only very specific compositions.

Taboada et al. (2011) considered ten levels for word-level sentiments and proposed a lexicon-based method for binary document-level sentiment classification. The proposed method develops a lexicon for the words with various

part-of-speech tags and handles compositional effects for certain syntactic patterns using predefined compositional rules. Similar to work by Liu and Seneff (2009), it models negation as an additive effect rather than a polarity flip. The drawback of the proposed method is that it relies on the creation of the extensive hand-ranked dictionaries.

In this thesis we propose a compositional matrix-space model for phrase-level ordinal sentiment classification that does not rely on sentiment lexica or hand-written compositional rules.

## 2.4 Summary of the Chapter

In this chapter we gave an overview of the related work in sentiment analysis. We started our discussion by describing the sentiment analysis research in various domains motivated by real-world applications. Then we describe in more detail the related work on document-level sentiment classification. We started by describing the conventional approaches to this task. We continued with an overview of work that goes beyond conventional models and incorporates knowledge about the structure of the document in statistical classification models. Then we describe work on "annotator rationales" that goes beyond conventional sentiment classification models by relying on additional information from human annotators. Finally, we describe related work on phrase-level sentiment classification.

CHAPTER 3

# USING AUTOMATICALLY DISCOVERED INFORMATIVE SENTENCES TO IMPROVE DOCUMENT-LEVEL SENTIMENT CLASSIFICATION

As described in Chapter 1, the task of document-level sentiment classification — automatically identifying whether a given document has an overall positive or overall negative sentiment — can be treated as standard text categorization task (Pang et al. (2002)). One of the central challenges in sentiment-based text categorization, however, is that not every portion of a given document is equally informative for inferring its overall sentiment. More specifically, (1) subjective documents are often comprised of objective and subjective parts (Pang and Lee (2004)) and (2) the subjective parts may consist of sentences with polarities opposite that of the document (Pang et al. (2002)). These issues complicate the task of sentiment classification (see example in Table 1.1).

Pang and Lee (2004) address (1) by employing the minimum cut algorithm to mitigate the effect of potentially objective sentences for document-level sentiment classification. More specifically, they suggest a two-stage approach: first, to filter out objective sentences and keep only the *subjective* ones; then use the resulting documents as the training examples for the classifier. One advantage of the approach proposed by Pang and Lee (2004) is that it does not require explicit manual annotations to filter out the objective sentences. However, the resulting subjective extracts could be noisy, and might not always lead to performance gains for document-level sentiment classification task.

Zaidan et al. (2007) address both (1) and (2) by asking human annotators to mark (at least some of) the relevant text spans that *support (or explain) each document-level sentiment decision*. The text spans of these "rationales" (or infor-

mative text spans)[1] are then used to construct additional training examples that can guide the learning algorithm toward better categorization models (we provide the details in Section 3.1).

*But could we perhaps enjoy the performance gains of rationale-enhanced learning models without any additional human effort whatsoever (beyond the document-level sentiment label)?* We hypothesize that in the area of sentiment analysis, where there has been a great deal of recent research attention given to various aspects of the task (Pang and Lee (2008), Liu (2012)), this might be possible: using existing resources for sentiment analysis, we might be able to automatically identify the informative segments. In this chapter, we explore a number of methods to automatically acquire informative segments for document-level sentiment classification. *For simplicity, we consider informative text spans only at the sentence-level.* In particular, we investigate the use of off-the-shelf sentiment analysis components and lexicons for this purpose. Our approaches for acquiring informative sentences can be viewed as *mostly unsupervised* in that we do not require manually annotated informative text spans for training.

**Roadmap of the Chapter.** The work described in this chapter is based on Yessenalina et al. (2010a). The rest of this chapter is organized as follows. We first briefly summarize the SVM-based learning approach of Zaidan et al. (2007) that allows the incorporation of informative text spans (Section 3.1). We next introduce three methods for the automatic acquisition of informative sentences (Section 3.2). The experimental results are presented in Section 3.3, followed by related work (Section 3.4) and summary of contributions (Section 3.5).

[1] In this chapter we will use the term ”rationales” and ”informative text spans” interchangeably.

## 3.1 Background: Incorporating Informative Text Spans as Additional Constraints

Zaidan et al. (2007) first introduced the notion of *informative text spans* (annotator rationales) — text spans highlighted by human annotators as support or evidence for each document-level sentiment decision. These spans, of course, are only useful if the sentiment categorization algorithm can be extended to exploit them effectively. With this in mind, Zaidan et al. (2007) propose the following extension to the standard SVM learning algorithm[2] (Joachims (1997)). They assume that the documents of interest are movie reviews. They also assume a standard text categorization approach in which each document $x_i$ is represented as a bag-of-words feature vector, that has 1 if a certain word from the active lexicon is present in a document, and 0 otherwise.

Let $\vec{x}_i$ be movie review $i$, and let $\{\vec{r}_{ij}\}$ be the set of *annotator rationales* that support the positive or negative sentiment decision for $\vec{x}_i$. For each such rationale $\vec{r}_{ij}$ in the set, construct a *contrast training example* $\vec{v}_{ij}$, by removing the text span associated with the rationale $\vec{r}_{ij}$ from the original review $\vec{x}_i$. Intuitively, the contrast example $\vec{v}_{ij}$ should not be as "easy" for the learning algorithm as the original review $\vec{x}_i$, since one of the supporting regions identified by the human annotator has been deleted. That is, the *correct* learned model should be *less confident* of its classification of a contrast example vs. the corresponding original example, and the classification boundary of the model should be modified accordingly.

Zaidan et al. (2007) formulate exactly this intuition as SVM constraints as follows:

$$(\forall i, j) : \; y_i \; (\vec{w}\vec{x}_i - \vec{w}\vec{v}_{ij}) \geq \mu(1 - \xi_{ij})$$

[2]We assume that the reader is familiar with SVM learning.

where $y_i \in \{-1, +1\}$ is the negative/positive sentiment label of document $i$, $\vec{w}$ is the weight vector, $\mu \geq 0$ controls the size of the margin between the original examples and the contrast examples, and $\xi_{ij}$ are the associated slack variables. After some re-writing of the equations, the resulting objective function and constraints for the SVM are as follows:

$$\frac{1}{2}||\vec{w}||^2 + C \sum_i \xi_i + C_{contrast} \sum_{ij} \xi_{ij} \tag{3.1}$$

subject to constraints:

$$(\forall i): \quad y_i \ \vec{w} \cdot \vec{x}_i \geq 1 - \xi_i, \quad \xi_i \geq 0$$

$$(\forall i, j): \quad y_i \ \vec{w} \cdot \vec{x}_{ij} \geq 1 - \xi_{ij} \quad \xi_{ij} \geq 0$$

where $\xi_i$ and $\xi_{ij}$ are the slack variables for $\vec{x}_i$ (the original examples) and $\vec{x}_{ij}$ ($\vec{x}_{ij}$ are named as *pseudo examples* and defined as $\vec{x}_{ij} = \frac{\vec{x}_i - \vec{v}_{ij}}{\mu}$), respectively. Intuitively, the pseudo examples ($\vec{x}_{ij}$) represent the difference between the original examples ($\vec{x}_i$) and the contrast examples ($\vec{v}_{ij}$), weighted by a parameter $\mu$. $C$ and $C_{contrast}$ are parameters to control the trade-offs between training errors and margins for the original examples $\vec{x}_i$ and pseudo examples $\vec{x}_{ij}$ respectively. As noted in Zaidan et al. (2007), $C_{contrast}$ values are generally smaller than $C$ for noisy rationales.

We will similarly employ the extension by Zaidan et al. (2007) to SVM learning to incorporate automatically, rather than manually, identified rationales for document-level sentiment categorization.

## 3.2 Automatically Acquiring Informative Sentences

Our goal is to automatically acquire informative sentences that will approximate human annotator rationales. For this, we rely on the following two assumptions:

(1) Regions marked as informative sentences are more subjective than unmarked regions.

(2) The sentiment of each informative sentence coincides with the document-level sentiment.

Note that assumption (1) was not observed in the Zaidan et al. (2007) work: annotators were asked only to mark a few rationales, leaving other (also subjective) rationale sections unmarked.

And at first glance, assumption (2) might seem too obvious. But it is important to include as there can be subjective regions with seemingly conflicting sentiment in the same document (Pang et al. (2002)). For instance, an author for a movie review might express a positive sentiment toward the movie, while also discussing a negative sentiment toward one of the fictional characters appearing in the movie. This implies that not all subjective regions will be relevant for the document-level sentiment classification — rather only those regions whose polarity matches that of the document should be considered.

In order to extract regions that satisfy the above assumptions, we first look for subjective regions in each document, then filter out those regions that exhibit a sentiment value (i.e., polarity) that conflicts with polarity of the document.

Because our ultimate goal is to reduce human annotation effort as much as possible, we do not employ supervised learning methods to directly learn to identify good rationales from human-annotated rationales. Instead, we opt for methods that make use of only the document-level sentiment and off-the-shelf utilities that were trained for slightly different sentiment classification tasks using a corpus from a different domain and of a different genre. Although such utilities might not be optimal for our task, we hypothesized that these basic resources from the research community would constitute an adequate source of

sentiment information for our purposes.

We next describe three methods for the automatic acquisition of rationales.

### 3.2.1 Contextual Polarity Classification

The first approach employs OpinionFinder (Wilson et al. (2005a)), an off-the-shelf opinion analysis utility.[3] In particular, OpinionFinder identifies phrases expressing positive or negative opinions. Because OpinionFinder models the task as a word-based classification problem rather than a sequence tagging task, most of the identified opinion phrases consist of a single word. In general, such short text spans cannot fully incorporate the contextual information relevant to the detection of subjective language (Wilson et al. (2005b)). Therefore, we conjecture that good rationales should extend beyond short phrases.[4] For simplicity, we choose to extend OpinionFinder phrases to sentence boundaries.

In addition, to be consistent with our second operating assumption, we keep only those sentences whose polarity coincides with the document-level polarity. In sentences where OpinionFinder marks multiple opinion words with opposite polarities we perform a simple voting — if words with positive (or negative) polarity dominate, then we consider the entire sentence as positive (or negative). We ignore sentences with a tie. Each selected sentence is considered as a separate rationale.

[3] www.cs.pitt.edu/mpqa/opinionfinderrelease/

[4] This conjecture is indirectly confirmed by the fact that human-annotated rationales are rarely a single word.

### 3.2.2 Polarity Lexicons

Unfortunately, domain shift as well as task mismatch could be a problem with any opinion utility based on supervised learning. It is worthwhile to note that OpinionFinder is trained on a newswire corpus whose prevailing sentiment is known to be negative (Wiebe et al. (2005)). Therefore, we next consider an approach that does not rely on supervised learning techniques but instead explores the use of a manually constructed polarity lexicon. In particular, we use the lexicon constructed for Wilson et al. (2005b), which contains about 8000 words. Each entry is assigned one of three polarity values: positive, negative, neutral. We construct rationales from the polarity lexicon for every instance of positive and negative words in the lexicon that appear in the training corpus. As in the OPINIONFINDER rationales, we extend the words found by the POLARITYLEXICON approach to sentence boundaries to incorporate potentially relevant contextual information. We retain as rationales only those sentences whose polarity coincides with the document-level polarity as determined via the voting scheme of Section 3.2.1.

### 3.2.3 Random Rationales

Finally, we acquire informative sentences randomly, selecting 25% of the sentences from each document and treating each as a separate rationale. We chose the value of 25% to match the percentage of sentences per document, on average, that contain human-annotated rationales in our dataset (24.7%). Note, that the percent of the informative sentences found by the OPINIONFINDER, POLARITYLEXICON, RANDOM RATIONALES are 22.8% 38.7% and 25.0% respectively.

Table 3.1: Comparison of Automatically Acquired Informative Sentences vs. Human-annotated Rationales.

| | Precision | | | Recall | | | F-Score | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | All | Pos | Neg | All | Pos | Neg | All | Pos | Neg |
| OPINIONFINDER | 54.9 | 56.1 | 54.6 | 45.1 | 22.3 | 65.3 | 49.5 | 31.9 | 59.5 |
| POLARITYLEXICON | 45.2 | 42.7 | 48.5 | 63.0 | 71.8 | 55.0 | 52.6 | 53.5 | 51.6 |
| RANDOM RATIONALES | 28.9 | 26.0 | 31.8 | 25.9 | 24.9 | 26.7 | 27.3 | 25.5 | 29.0 |

### 3.2.4 Comparison of Automatically Acquired Informative Sentences vs. Human Annotated Sentences

Before evaluating the performance of the automatically acquired informative sentences, we summarize in Table 3.1 the differences between automatic vs. human-annotated rationales. All computations were performed on the same movie review dataset of Pang and Lee (2004) used in Zaidan et al. (2007). Note that the Zaidan et al. (2007) annotation guidelines did not insist that annotators mark **all** rationales, only that some were marked for each document. Nevertheless, we report precision, recall, and F-score based on overlap with the human-annotated rationales of Zaidan et al. (2007), so as to demonstrate the degree to which the proposed approaches align with human intuition. Overlap measures were also employed by Zaidan et al. (2007).

As shown in Table 3.1, the annotator rationales found by OPINIONFINDER (F-score 49.5%) and POLARITYLEXICON (F-score 52.6%) match the human rationales much better than those found by RANDOM RATIONALES (F-score 27.3%). Also as expected, OPINIONFINDER's positive rationales match the human rationales at a significantly lower level (F-score 31.9%) than negative rationales (59.5%). This is due to the fact that OpinionFinder is trained on a dataset biased toward negative sentiment (see Section 3.2.2). In contrast, all other approaches

show a balanced performance for positive and negative rationales vs. human rationales.

## 3.3 Experiments

For our experiments with contrast examples we use $SVM^{light}$ (Joachims (1999)). We evaluate the usefulness of automatically acquired informative sentences on five different datasets. The first is the movie review data of Pang and Lee (2004), which was manually annotated with rationales by Zaidan et al. (2007)[5]; the remaining are four product review datasets from Blitzer et al. (2007).[6] Only the movie review dataset contains human annotator rationales. We replicate the same feature set and experimental set-up as in Zaidan et al. (2007) to facilitate comparison with their work.

- We use binary unigram features corresponding to the unstemmed words or punctuation marks with count greater or equal to 4 in the full 2000 documents, then we normalize the examples to the unit length. When computing the pseudo examples $\vec{x}_{ij} = \frac{\vec{x}_i - \vec{v}_{ij}}{\mu}$ we first compute $(\vec{x}_i - \vec{v}_{ij})$ using the binary representation. As a result, features (unigrams) that appeared in both vectors will be zeroed out in the resulting vector. We then normalize the resulting vector to a unit vector.

As discussed in Section 3.1 the framework for learning with contrast examples introduced in Zaidan et al. (2007) requires three parameters: ($C$, $\mu$, $C_{contrast}$), where $C$ and $C_{contrast}$ are parameters to control the trade-off between training error and margins for the original examples and pseudo examples respectively; $\mu$ controls the size of a margin between the original examples and the contrast

[5]Available at http://www.cs.jhu.edu/∼ozaidan/rationales/.
[6]http://www.cs.jhu.edu/∼mdredze/datasets/sentiment/.

examples. To set the parameters, we use a grid search with step 0.1 for the range of values of each parameter around the point (1,1,1). In total, we try around 3000 different parameter triplets for each type of rationales.

### 3.3.1 Experiments with Movie Review Data

We follow Zaidan et al. (2007) for the training/test data splits. The top half of Table 3.2 shows the performance of a system trained with **no annotator rationales** vs. two variations of human annotator rationales. The NORATIONALES system is trained on the full text for each document.

Table 3.2: Experimental results for the movie review data.

| Method | Accuracy |
|---|---|
| NORATIONALES | 88.56 |
| HUMANR | 91.61$^{\bullet}$ |
| HUMANR@SENTENCE | 91.33$^{\bullet\ \dagger}$ |
| OPINIONFINDER | 91.78$^{\bullet\ \dagger}$ |
| POLARITYLEXICON | 91.39$^{\bullet\ \dagger}$ |
| RANDOM RATIONALES | 90.00$^{*}$ |
| OPINIONFINDER+HUMANR@SENTENCE | 92.50$^{\bullet\ \triangle}$ |

– The numbers marked with $^{\bullet}$ (or $^{*}$) are statistically significantly better than NORATIONALES according to a paired t-test with $p < 0.001$ (or $p < 0.01$).
– The numbers marked with $^{\triangle}$ are statistically significantly better than HUMANR according to a paired t-test with $p < 0.01$.
– The numbers marked with $^{\dagger}$ are ***not*** statistically significantly worse than the human rationales (HUMANR) according to a paired t-test with $p > 0.1$.

HUMANR treats each rationale in the same way as Zaidan et al. (2007). HUMANR@SENTENCE extends the human annotator rationales to sentence boundaries, and then treats each such sentence as a separate rationale. As shown in Table 3.2, we get almost the same performance from these two variations (91.33% and 91.61%).[7] This result demonstrates that locking rationales to sen-

[7]The performance of HUMANR reported by Zaidan et al. (2007) is 92.2% which lies between

tence boundaries was a reasonable choice.

Among the approaches that make use of only automatic rationales (bottom half of Table 3.2), the best is OPINIONFINDER, reaching 91.78% accuracy. This result is slightly better than results exploiting human rationales (91.33-91.61%), although the difference is not statistically significant. This result demonstrates that automatically generated rationales are just as good as human rationales in improving document-level sentiment classification. Similarly strong results are obtained from the POLARITYLEXICON as well.

Rather unexpectedly, RANDOM RATIONALES also achieves statistically significant improvement over NORATIONALES (90.0% vs. 88.56%). However, notice that the performance of RANDOM RATIONALES is statistically significantly lower than those based on human rationales (91.33-91.61%).

In our experiments so far, we observed that some of the automatic rationales are just as good as human rationales in improving the document-level sentiment classification. Could we perhaps achieve an even better result if we combine the automatic rationales with human rationales? The answer is yes! The accuracy of OPINIONFINDER+HUMANR@SENTENCE reaches 92.50%, which is statistically significantly better than HUMANR (91.61%). In other words, not only can our automatically generated rationales replace human rationales, but they can also improve upon human rationales when they are available.

### 3.3.2 Experiments with Product Reviews

We next evaluate our approaches on datasets for which human annotator rationales do not exist. For this, we use some of the product review data from Blitzer et al. (2007): reviews for Books, DVDs, Videos and Kitchen appliances.

the performance we get (91.61%) and the oracle accuracy we get if we knew the best parameters for the test set (92.67%).

Each dataset contains 1000 positive and 1000 negative reviews. The reviews, however, are substantially shorter than those in the movie review dataset: the average number of sentences in each review is 9.20/9.13/8.12/6.37 respectively vs. 30.86 for the movie reviews. We perform 10-fold cross-validation, where 8 folds are used for training, 1 fold for tuning parameters, and 1 fold for testing. Table 3.3 shows the results.

Table 3.3: Experimental results for Product Review data.

| Method | Books | DVDs | Videos | Kitchen |
|---|---|---|---|---|
| NoRationales | 80.20 | 80.95 | 82.40 | 87.40 |
| OPINIONFINDER | 81.65$^*$ | 82.35$^*$ | 84.00$^*$ | 88.40 |
| POLARITYLEXICON | 82.75$^\bullet$ | 82.85$^\bullet$ | 84.55$^\bullet$ | 87.90 |
| RANDOM RATIONALES | 82.05$^\bullet$ | 82.10$^\bullet$ | 84.15$^\bullet$ | 88.00 |

– The numbers marked with $^\bullet$ (or $^*$) are statistically significantly better than NORATIONALES according to a paired t-test with $p < 0.05$ (or $p < 0.08$).

Rationale-based methods perform statistically significantly better than NORATIONALES for all but the Kitchen dataset. An interesting trend in product review datasets is that RANDOM RATIONALES rationales are just as good as other more sophisticated rationales. We suspect that this is because product reviews are generally shorter and more focused than the movie reviews, thereby any randomly selected sentence is likely to be a good rationale. Quantitatively, subjective sentences in the product reviews amount to 78% (McDonald et al. (2007)), while subjective sentences in the movie review dataset constitute only about 25% (Mao and Lebanon (2006)).

### 3.3.3 Examples of Automatically Acquired Informative Sentences

In this section, we examine an example to compare the automatically generated rationales (using OPINIONFINDER) with human annotator rationales for the movie review data. In the following positive document snippet, automatic rationales are underlined, while **human-annotated rationales** are in bold face.

> ...But a little niceness goes a long way these days, and **there's no denying the entertainment value** of that thing you do! **It's just about impossible to hate**. **It's an inoffensive, enjoyable piece** of nostalgia that is **sure to leave audiences smiling and humming, if not singing**, "that thing you do!" — quite possibly for days...

Notice that, although OPINIONFINDER misses some human rationales, it avoids the inclusion of "impossible to hate", which contains only negative terms and is likely to be confusing for the learning framework with contrast examples.

## 3.4 Related Work

In broad terms, automatically constructing rationales and using them to formulate contrast examples can be viewed as learning with prior knowledge (e.g., Schapire et al. (2002), Wu and Srihari (2004)). In our task, the prior knowledge corresponds to our operational assumptions given in Section 3.2: for the document-level sentiment classification task, good rationales are likely to be subjective, and their sentiments should match the document-level sentiment.

Our operational assumptions can further be loosely connected to recognizing and exploiting discourse structure. Taboada et al. (2009) investigate this

aspect more directly, by categorizing each paragraph as either "formal" or "functional", and further dividing the functional paragraphs into "description" or "comment". Then the authors make use of such discourse information to improve document-level sentiment classification. The work of Taboada et al. (2009), however, requires human annotation for the discourse information. In contrast, our approaches do not make use of human annotations at the sentence or paragraph level. The work of Pang and Lee (2004) recognizes and exploits discourse structure implicitly, without requiring extra human annotation. The main difference from our approach is that Pang and Lee (2004) incorporate the discourse information at inference time using the minimum cut algorithm, while we make use of it at training time using the learning framework with contrast examples.

## 3.5 Summary of the Chapter

In this chapter, we explored methods to automatically acquire informative sentences for document-level sentiment classification. Our study is motivated by the desire to retain the performance gains of rationale-enhanced learning models while eliminating the need for additional human annotation effort. By employing existing resources for sentiment analysis, we automatically discovered informative sentences that are as good as human annotator rationales in improving document-level sentiment classification.

CHAPTER 4

# MULTI-LEVEL STRUCTURED MODELS FOR DOCUMENT-LEVEL SENTIMENT CLASSIFICATION

Chapter 1 suggested that all parts of a document should not be treated equally in document-level sentiment classification: some parts are more indicative of the sentiment label of the document than others. As the movie review from Table 1.1 shows, objective sentences are often interleaved with subjective ones; moreover, an overall positive review might still include some negative opinions about an actor or the plot. This makes the sentiment classification task harder for machine learning methods using bag-of-words representations, that treat all words in the document in the same way, and ignore sentence structure. In particular, the positive (negative) words can potentially appear in positive (negative) documents as well as in negative (positive) ones. *In this chapter, as in Chapter 3, we continue exploiting the sentence structure of the document for document-level sentiment classification; however, we incorporate the structure differently.* We develop a two-level structured model for document-level sentiment classification that jointly learns to predict the document-level sentiment and the set of informative sentences that explain the label of the document.

As discussed in Chapter 2, early research on document-level sentiment classification employed conventional machine learning techniques for text categorization (Pang et al. (2002)). These methods, however, assume that documents are represented via a flat feature vector (e.g., a bag-of-words). As a result, their ability to identify and exploit subjectivity (or other useful) information at the sentence-level is limited.

And although researchers subsequently proposed methods for incorporating sentence-level subjectivity information, existing techniques have some un-

desirable properties. First, they typically require gold standard sentence-level annotations (McDonald et al. (2007), Mao and Lebanon (2006)). But the cost of acquiring such labels can be prohibitive. Second, some solutions for incorporating sentence-level information lack mechanisms for controlling error propagation from the subjective sentence identification subtask to the main document classification task (Pang and Lee (2004)). Finally, solutions that attempt to handle the error propagation problem have done so by explicitly optimizing for the best *combination* of document-level and sentence-level classification accuracy (McDonald et al. (2007)). Optimizing for this compromise, when the real goal is to maximize only the document-level accuracy, can potentially hurt document-level performance.

In this chapter, we propose a joint two-level model to address the aforementioned concerns. We formulate our training objective to directly optimize for document-level accuracy. Further, we do not require gold standard sentence-level labels for training. Instead, our training method treats sentence-level labels as hidden variables and *jointly learns* to predict the document label and those *informative* sentences that best "explain" it, thus controlling the propagation of incorrect sentence labels. And by directly optimizing for document-level accuracy, our model learns to solve the informative sentence extraction subtask only to the extent required for accurately classifying document sentiment. Empirical evaluations on movie reviews and U.S. Congressional floor debates show improved performance over previous approaches.

**Roadmap of the Chapter.** The material described in this chapter is based on Yessenalina et al. (2010b). In the rest of this chapter, we will discuss related work (Section 4.1), motivate (Section 4.2) and describe our model (Section 4.3). Then we present an empirical evaluation of our model on movie reviews and U.S.

Congressional floor debates datasets (Section 4.4). We close this chapter with discussion (Section 4.5) and conclusions (Section 4.6).

## 4.1 Related Work

Pang and Lee (2004) first showed that sentence-level extraction can improve document-level performance (see Chapter 2 for more details). One advantage of their two-stage approach is that it avoids the need for explicit subjectivity annotations. However, it employs a cascaded approach in which the output of an earlier (sentence-level) stage is consumed as input to the subsequent (document-level) stage. And like other cascaded approaches to sentiment classification (e.g., Thomas et al. (2006), Mao and Lebanon (2006)), it can be difficult to control error propagation. from the sentence-level subtask to the main document classification task.

Instead of taking a cascaded approach, one can directly modify the training of flat document classifiers using lower-level information. For instance, Zaidan et al. (2007) used human annotators to mark the "annotator rationales", which are text spans that support the document's sentiment label. These rationales are then used to formulate additional constraints during SVM training to ensure that the resulting document classifier is less confident in classifying a document that does not contain the rationale versus the original document. In Chapter 3 we extended their approach to use automatically generated rationales.

A natural approach to avoid the pitfalls associated with cascaded methods is to use joint two-level models that simultaneously solve the sentence-level and document-level tasks (e.g., McDonald et al. (2007), Zaidan and Eisner (2008)) Since these models are trained jointly, the sentence-level predictions affect the document-level predictions and vice-versa. However, such approaches typi-

cally require sentence-level annotations during training, which can be expensive to acquire. Furthermore, the training objectives are usually formulated as a compromise between sentence-level and document-level performance. If the goal is to predict well at the document-level, then these approaches are solving a much harder problem that is not exactly aligned with maximizing document-level accuracy.

Recently, researchers within both Natural Language Processing (e.g., Petrov and Klein (2007), Chang et al. (2010), Clarke et al. (2010)) and other fields (e.g., Felzenszwalb et al. (2008), Yu and Joachims (2009)) have analyzed joint multi-level models (i.e., models that simultaneously solve the main prediction task along with important subtasks) that are trained using limited or no explicit lower-level annotations. Similar to our approach, the lower-level labels are treated as hidden or latent variables during training. Although the training process is non-trivial (and in particular requires a good initialization of the hidden variables), it avoids the need for human annotations for the lower-level subtasks. Some researchers have also recently applied hidden variable models to sentiment analysis, but they were focused on classifying either phrase-level (Choi and Cardie (2008)) or sentence-level polarity (Nakagawa et al. (2010)).

## 4.2 Extracting Latent Explanations

In this chapter, we take the view that each document has a subset of sentences that best explains its sentiment. Consider the "annotator rationales" generated by human judges for the movie reviews dataset (Zaidan et al. (2007)). Each rationale is a text span that was identified to support (or explain) its parent document's sentiment. Thus, these rationales can be interpreted as (something close to) a ground truth labeling of the explanatory segments. Using a dataset

where each document contains only its rationales, cross validation experiments using an SVM classifier yield 97.44% accuracy — as opposed to 86.33% accuracy when using the full text of the original documents. Clearly, extracting the best supporting segments can offer a tremendous performance boost.

We are interested in settings where human-extracted explanations such as annotator rationales might not be readily available, or are imperfect. As such, we will formulate the set of extracted sentences as latent or hidden variables in our model. Viewing the extracted sentences as latent variables will pose no new challenges during prediction, since the model is expected to predict all labels at test time. We will leverage recent advances in training latent variable SVMs (Yu and Joachims (2009)) to arrive at an effective training procedure.

## 4.3 Model: Structural SVMs for Sentiment Classification with Latent Explanations (*SVM*$^{sle}$)

In this section, we present a two-level document classification model. Although our model makes predictions at both the document and sentence levels, it will be trained (and evaluated) only with respect to document-level performance. We begin by presenting the feature structure and inference method. We will then describe a supervised training algorithm based on structural SVMs, and finally discuss some extensions and design decisions.

Let $x$ denote a document, $y = \pm 1$ denote the sentiment (for us, a binary positive or negative polarity) of a document, and $s$ denote a subset of explanatory sentences in $x$. Let $\Psi(x, y, s)$ denote a joint feature map that outputs features describing the quality of predicting sentiment $y$ using explanation $s$ for document $x$. We focus on linear models, so given a (learned) weight vector $\vec{w}$, we can write

the quality of predicting $y$ (with explanation $s$) as

$$F(x, y, s; \vec{w}) = \vec{w}^T \Psi(x, y, s), \tag{4.1}$$

and a document-level sentiment classifier as

$$h(x; \vec{w}) = \operatorname*{argmax}_{y=\pm 1} \max_{s \in S(x)} F(x, y, s; \vec{w}), \tag{4.2}$$

where $S(x)$ denotes the collection of feasible explanations (e.g., subsets of sentences) for $x$.

Let $x^j$ denote the $j$-th sentence of $x$. We propose the following instantiation of (4.1),

$$\vec{w}^T \Psi(x, y, s) = \frac{1}{N(x)} \sum_{j \in s} y \cdot \vec{w}_{pol}^T \psi_{pol}(x^j) + \vec{w}_{subj}^T \psi_{subj}(x^j), \tag{4.3}$$

where the first term in the summation captures the quality of predicting polarity $y$ on sentences in $s$, the second term captures the quality of predicting sentences in $s$ as the subjective sentences, and $N(x)$ is a normalizing factor (which will be discussed in more detail in Section 4.3.3). We represent the weight vector as

$$\vec{w} = \begin{bmatrix} \vec{w}_{pol} \\ \\ \vec{w}_{subj} \end{bmatrix}, \tag{4.4}$$

and $\psi_{pol}(x^j)$ and $\psi_{subj}(x^j)$ denote the polarity and subjectivity features of sentence $x^j$, respectively. Note that $\psi_{pol}$ and $\psi_{subj}$ are disjoint by construction, i.e., $\psi_{pol}^T \psi_{subj} = 0$. We will present extensions in Section 4.3.5.

For example, suppose $\psi_{pol}$ and $\psi_{subj}$ were both bag-of-words feature vectors. Then we might learn a high weight for the feature corresponding to the word "think" in $\psi_{subj}$ since that word is indicative of the sentence being subjective (but not necessarily indicating positive or negative polarity).

**Algorithm 1** Inference Algorithm for (4.2)

Input: $x$
Output: $(y, s)$
$s_+ \leftarrow \text{argmax}_{s \in S(x)} \vec{w}^T \Psi(x, +1, s)$
$s_- \leftarrow \text{argmax}_{s \in S(x)} \vec{w}^T \Psi(x, -1, s)$
**if** $\vec{w}^T \Psi(x, +1, s_+) > \vec{w}^T \Psi(x, -1, s_-)$ **then**
    Return $(+1, s_+)$
**else**
    Return $(-1, s_-)$
**end if**

### 4.3.1 Making Predictions

Algorithm 1 describes our inference procedure. Recall from (4.2) that our hypothesis function predicts the sentiment label that maximizes (4.3). To do this, we compare the best set of sentences that explains a positive polarity prediction with the best set that explains a negative polarity prediction.

We now specify the structure of $S(x)$. In this chapter, we use a cardinality constraint,

$$S(x) = \{s \subseteq \{1, \ldots, |x|\} \ : \ |s| \leq f(|x|)\}, \tag{4.5}$$

where $f(|x|)$ is a function that depends only on the number of sentences in $x$. For example, a simple function is $f(|x|) = |x| \cdot 0.3$, indicating that at most 30% of the sentences in $x$ can be informative (explain the sentiment label of the document).

Using this definition of $S(x)$, we can then compute the best set of informative sentences for each possible $y$ by computing the joint subjectivity and polarity score of each sentence $x^j$ in isolation,

$$y \cdot \vec{w}_{pol}^T \psi_{pol}(x^j) + \vec{w}_{subj}^T \psi_{subj}(x^j),$$

and selecting the top $f(|x|)$ as $s$ (or fewer, if there are fewer than $f(|x|)$ that have positive joint score).

### 4.3.2 Training

For training, we will use an approach based on latent variable structural SVMs (Yu and Joachims (2009)).

**Optimization Problem (OP) 1.**

$$\min_{\vec{w},\xi\geq 0} \frac{1}{2}\|\vec{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \tag{4.6}$$

*s.t.* $\forall i$ :

$$\max_{s\in S_i} \vec{w}^T\Psi(x_i, y_i, s) \geq \max_{s'\in S(x_i)} \vec{w}^T\Psi(x_i, -y_i, s') + 1 - \xi_i \tag{4.7}$$

OP 1 optimizes the standard SVM training objective for binary classification. Each training example has a corresponding constraint (4.7). This constraint ensures that the score of the highest scoring explanation for the training polarity label is larger than score of the highest scoring explanation for the opposite polarity label. Note, that we never observe the true explanation for the training labels; they are the hidden or latent variables. The hidden variables are also ignored in the objective function.

As a result, one can interpret OP 1 to be directly optimizing a trade-off between model complexity (as measured using the 2-norm) and document-level classification error in the training set. This has two main advantages over related training approaches. First, it solves the multi-level problem jointly as opposed to separately, which avoids introducing difficult to control propagation errors. Second, it does not require solving the sentence-level task perfectly, and also does not require precise sentence-level training labels. In other words, our goal is to learn to identify the informative sentences that best explain the training labels to the extent required for good document classification performance.

OP 1 is non-convex because of the constraints (4.7). To solve OP 1, we use a combination of the CCCP algorithm (Yuille and Rangarajan (2003)) with cutting

**Algorithm 2** Training Algorithm for OP 1

1: Input: $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ *//training data*
2: Input: $C$ *//regularization parameter*
3: Input: $(s_1, \ldots, s_N)$ *//initial guess*
4: $\vec{w} \leftarrow$ *SSVMSolve*$(C, \{(x_i, y_i, s_i)\}_{i=1}^N)$
5: **while** $\vec{w}$ not converged **do**
6: **for** $i = 1, \ldots, N$ **do**
7: $s_i \leftarrow \text{argmax}_{s \in S(x_i)} \vec{w}^T \Psi(x_i, y_i, s)$
8: **end for**
9: $\vec{w} \leftarrow$ *SSVMSolve*$(C, \{(x_i, y_i, s_i)\}_{i=1}^N)$
10: **end while**
11: Return $\vec{w}$

plane training of structural SVMs (Joachims et al. (2009)), as proposed by Yu and Joachims (2009). Suppose each constraint (4.7) is replaced by

$$\vec{w}^T \Psi(x_i, y_i, s_i) \geq \max_{s' \in S(x_i)} \vec{w}^T \Psi(x_i, -y_i, s') + 1 - \xi_i,$$

where $s_i$ is some fixed explanation (e.g., an initial guess of the best explanation). Then OP 1 reduces to a standard structural SVM, which can be solved efficiently (Joachims et al. (2009)). Algorithm 2 describes our training procedure. Starting with an initial guess $s_i$ for each training example, the training procedure alternates between solving an instance of the resulting structural SVM (called *SSVMSolve* in Algorithm 2) using the currently best known explanations $s_i$ (Line 9), and making a new guess of the best explanations (Line 7). Yu and Joachims (2009) showed that this alternating procedure for training latent variable structural SVMs is an instance of the CCCP procedure (Yuille and Rangarajan (2003)), and so is guaranteed to converge to a local optimum.

For our experiments, we do not train until convergence, but instead use performance on a validation set to choose the halting iteration. Since OP 1 is non-convex, a good initialization is necessary. To generate the initial explanations, one can use an off-the-shelf sentiment classifier such as the OpinionFinder system (Wilson et al. (2005b)) introduced in Chapter 3. For some datasets, there ex-

ist documents with annotated sentences, which we can treat either as the ground truth or another (very good) initial guess of the explanatory sentences.

### 4.3.3 Feature Representation

Like any machine learning approach, we must specify a useful set of features for the $\psi$ vectors described above. We will consider two types of features.

**Bag-of-words.** Perhaps the simplest approach is to define $\psi$ using a bag-of-words feature representation, with one feature corresponding to each word in the active lexicon of the corpus. Using such a feature representation might allow us to learn which words have high polarity (e.g., "great") and which are indicative of subjective sentences (e.g., "opinion").

**Sentence properties.** We can incorporate many useful features to describe sentence subjectivity. For example, subjective sentences might densely populate the end of a document, or exhibit spatial coherence (so features describing previous sentences might be useful for classifying the current sentence). Such features cannot be compactly incorporated into flat models that ignore the document structure.

For our experiments, we normalize each $\psi_{subj}$ and $\psi_{pol}$ to have unit 2-norm.

**Joint Feature Normalization.** Another design decision is the choice of normalization $N(x)$ in (4.3). Two straightforward choices are $N(x) = f(|x|)$ and $N(x) = \sqrt{f(|x|)}$, where $f(|x|)$ is the size constraint as described in (4.5). In our experiments we tried both and found the square root normalization to work better in practice; therefore all the experimental results are reported using $N(x) = \sqrt{f(|x|)}$. We suggest an analysis that sheds light on when square

root normalization can be useful.

**Analysis.** Recall that all the $\psi_{subj}$ and $\psi_{pol}$ vectors have unit 2-norm, which is assumed here to be desirable. We now show that using $N(x) = \sqrt{f(|x|)}$ achieves a similar property for $\Psi(x, y, s)$. We can write the squared 2-norm of $\Psi(x, y, s)$ as

$$
\begin{aligned}
|\Psi(x,y,s)|^2 &= \frac{1}{N(x)^2}\left[\sum_{j\in s} y\cdot\psi_{pol}(x^j)+\psi_{subj}(x^j)\right]^2 \\
&= \frac{1}{f(|x|)}\left[\left(\sum_{j\in s}\psi_{pol}(x^j)\right)^2+\left(\sum_{j\in s}\psi_{subj}(x^j)\right)^2\right],
\end{aligned}
$$

where the last equality follows from the fact that

$$\psi_{pol}(x^j)^T\psi_{subj}(x^j) = 0,$$

due to the two vectors using disjoint feature spaces by construction. The summation of the $\psi_{pol}(x^j)$ terms is written as

$$
\begin{aligned}
\left(\sum_{j\in s}\psi_{pol}(x^j)\right)^2 &= \sum_{j\in s}\sum_{i\in s}\psi_{pol}(x^j)^T\psi_{pol}(x^i) \\
&\approx \sum_{j\in s}\psi_{pol}(x^j)^T\psi_{pol}(x^j) \qquad (4.8) \\
&= \sum_{j\in s} 1 \le f(|x|),
\end{aligned}
$$

where (4.8) follows from the sparsity assumption that

$$\forall i \neq j:\ \psi_{pol}(x^j)^T\psi_{pol}(x^i) \approx 0.$$

A similar argument applies for the $\psi_{subj}(x^j)$ terms. Thus, by choosing $N(x) = \sqrt{f(|x|)}$ the joint feature vectors $\Psi(x, y, s)$ will have approximately equal magnitude as measured using the 2-norm.

### 4.3.4 Incorporating Proximity Information

As mentioned in Section 4.3.3, it is possible (and likely) for subjective sentences to exhibit spatial coherence (e.g., they might tend to group together). To exploit this structure, we will expand the feature space of $\psi_{subj}$ to include both the words of the current and previous sentence as follows,

$$\psi_{subj}(x, j) = \begin{bmatrix} \psi_{subj}(x^j) \\ \\ \psi_{subj}(x^{j-1}) \end{bmatrix}.$$

The corresponding weight vector can be written as

$$\vec{w}'_{subj} = \begin{bmatrix} \vec{w}_{subj} \\ \\ \vec{w}_{prevSubj} \end{bmatrix}.$$

By adding these features, we are essentially assuming that the words of the previous sentence are predictive of the subjectivity of the current sentence.

Alternative approaches include explicitly accounting for this structure by treating informative (explanatory) sentence extraction as a sequence-labeling problem, such as in McDonald et al. (2007). Such structure formulations can be naturally encoded in the joint feature map. Note that the inference procedure in Algorithm 1 is still tractable, since it reduces to comparing the best sequence of informative/non-informative sentences that explains a positive sentiment versus the best sequence that explains a negative sentiment. For this study, we chose not to examine this more expressive yet more complex structure.

### 4.3.5 Extensions

Though our initial model (4.3) is simple and intuitive, performance can depend heavily on the quality of latent variable initialization and the quality of the feature structure design. Consider the case where the initialization contains only

objective sentences that do not convey any sentiment. Then all the features initially available during training are generated from these objective sentences and are thus useless for sentiment classification. In other words, too much useful information has been suppressed for the model to make effective decisions. To hedge against learning poor models due to using a poor initialization and/or a suboptimal feature structure, we now propose extensions that incorporate information from the entire document.

We identify the following desirable properties that any such extended model should satisfy:

(A) The model should be linear.

(B) The model should be trained jointly.

(C) The component that models the entire document should influence which sentences are extracted.

The first property stems from the fact that our approach relies on linear models. The second property is desirable since joint training avoids error propagation that can be difficult to control. The third property deals with the information suppression issue.

**Regularizing Relative to a Prior: *SVM*$^{sle}$ with Prior**

We first consider a model that satisfies properties (A) and (C). Using the representation in (4.4), we propose a training procedure that regularizes $\vec{w}_{pol}$ relative to a prior model. Suppose we have a weight vector $\vec{w}_0$ which indicated the a priori guess of the contribution of each corresponding feature, then we can train our model using OP 2,

**Optimization Problem (OP) 2.**

$$\min_{\vec{w},\xi\geq 0} \frac{1}{2}\|\vec{w}-\vec{w}_0\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \forall i:$$

$$\max_{s\in S_i} \vec{w}^T\Psi(x_i,y_i,s) \geq \max_{s'\in S(x_i)} \vec{w}^T\Psi(x_i,-y_i,s') + 1 - \xi_i$$

For our experiments, we use

$$\vec{w}_0 = \begin{bmatrix} \vec{w}_{doc} \\ \\ 0 \end{bmatrix},$$

where $\vec{w}_{doc}$ denotes a weight vector trained to classify the polarity of entire documents. Then one can interpret OP 2 as enforcing that the polarity weights $\vec{w}_{pol}$ not be too far from $\vec{w}_{doc}$. Note that $\vec{w}_0$ must be available before training. Therefore this approach does not satisfy property (B).

**Extended Feature Space: *SVM*$^{sle}$ with Feature Smoothing (*SVM*$^{sle}_{fs}$)**

One simple way to satisfy all three aforementioned properties is to jointly model not only polarity and subjectivity of the extracted sentences, but also polarity of the entire document. Let $\vec{w}_{doc}$ denote the weight vector used to model the polarity of entire document $x$ (so the document polarity score is then $\vec{w}^T_{doc}\psi_{pol}(x)$). We can also incorporate this weight vector into our structured model to compute a smoothed polarity score of each sentence via $\vec{w}^T_{doc}\psi_{pol}(x^j)$. Following this intuition, we propose the following structured model,

$$\begin{aligned}\vec{w}^T\Psi(x,y,s) = \\ \frac{y}{N(x)}\left(\sum_{j\in s}\left(\vec{w}^T_{pol}\psi_{pol}(x^j) + \vec{w}^T_{doc}\psi_{pol}(x^j)\right)\right) \\ + \frac{1}{N(x)}\left(\sum_{j\in s}\vec{w}^T_{subj}\psi_{subj}(x^j)\right) + y\cdot\vec{w}^T_{doc}\psi_{pol}(x)\end{aligned}$$

where the weight vector is now

$$\vec{w} = \begin{bmatrix} \vec{w}_{pol} \\ \vec{w}_{subj} \\ \vec{w}_{doc} \end{bmatrix}.$$

Training this model via OP 1 achieves that $\vec{w}_{doc}$ is (1) used to model the polarity of the entire document, and (2) used to compute a smoothed estimate of the polarity of the extracted sentences. This satisfies all three properties (A), (B), and (C), although other approaches are also possible.

## 4.4 Experiments

We empirically evaluate the models proposed in the previous section. We start by describing the datasets and the experimental setup in Section 4.4.1 and discuss the experimental results in Section 4.4.2.

### 4.4.1 Experimental Setup

We evaluate our methods using the Movie Reviews and U.S. Congressional Floor Debates datasets, following the setup used in previous work for comparison purposes.[1]

**Movie Reviews.** We use the movie reviews dataset from Zaidan et al. (2007) (originally released by Pang and Lee (2004)), that contains annotated rationales for each review. We use those annotated rationales to generate an additional

[1] A software implementation of our method is publicly available `http://projects.yisongyue.com/svmsle/`. Datasets in the required format for *SVM*$^{sle}$ are available at `http://www.cs.cornell.edu/~ainur/data.html`

initialization during training (described below). We follow exactly the experimental setup used in Zaidan et al. (2007). In particular, since the rationale annotations are available for nine out of 10 folds, we used the 10-th fold as the blind test set. We trained nine different models on subsets of size eight, used the remaining fold as the validation set, and then measured the average performance on the final test set.

**U.S. Congressional Floor Debates.** We also use the U.S. Congressional floor debates transcripts from Thomas et al. (2006). The data was extracted from GovTrack (http://govtrack.us), which has all available transcripts of U.S. floor debates in the House of Representatives in 2005. As in previous work, only debates with discussions of "controversial" bills were considered (where the losing side had at least 20% of the speeches). The goal is to predict the vote ("yea" or "nay") for the speaker of each speech segment. For our experiments, we evaluate our methods using the speaker-based speech-segment classification setting as described in Thomas et al. (2006).[2]

Since our training procedure solves a non-convex optimization problem, it requires an initial guess of the explanatory sentences. We use an explanatory set size (4.5) of 30% of the number of sentences in each document, $L = \lceil 0.3 \cdot |x| \rceil$, with a lower cap of 1. We generate initialization using OpinionFinder (Wilson et al. (2005b)), which was shown to be a reasonable substitute for human annotations in the Movie Reviews dataset in Chapter 3 of this dissertation. We select all sentences whose majority vote of word-level polarities predicted by OpinionFinder matches the document's sentiment. If there are fewer than $L$ sentences, we add sentences starting from the end of the document. If there are

[2]In the other setting described in Thomas et al. (2006) (segment-based speech-segment classification), around 39% of the documents in the whole dataset contain only 1-3 sentences, making it an uninteresting setting to analyze with our model.

more, we remove sentences starting from the beginning of the document.

We consider two additional (baseline) methods for initialization: using a random set of sentences, and using the last 30% of sentences in the document. In the Movie Reviews dataset, we also use sentences containing human annotator rationales as a final initialization option. No such manual annotations are available for the Congressional Debates.

### 4.4.2 Experimental Results

We evaluate three versions of our model: the initial model (4.3) which we call $SVM^{sle}$ (**SVM**s for **S**entiment classification with **L**atent **E**xplanations), $SVM^{sle}$ regularized relative to a prior as described in Section 4.3.5 which we refer to as $SVM^{sle}$ w/ Prior,[3] and the feature smoothing model described in Section 4.3.5 which we call $SVM^{sle}_{fs}$. Due to the difficulty of selecting a good prior, we expect $SVM^{sle}_{fs}$ to exhibit the most robust performance.

Tables 4.1 and 4.2 show a comparison of our proposed methods on the two datasets. We observe that $SVM^{sle}_{fs}$ provides both strong and robust performance. The performance of $SVM^{sle}$ is generally better when trained using a prior than not in the Movie Reviews dataset. Both extensions appear to hurt performance in the U.S. Congressional Floor Debates dataset. Using OpinionFinder to initialize our training procedure offers good performance across both datasets, whereas the baseline initializations exhibit more erratic performance behavior.[4] Unsurprisingly, initializing using human annotations (in the Movie Reviews dataset) can offer further improvement. Adding proximity features (as

[3]We either used the same value of $C$ to train both standard SVM model and $SVM^{sle}$ w/ Prior or used the best standard SVM model on the validation set to train $SVM^{sle}$ w/ Prior. We chose the combination that works the best on the validation set.

[4]Using the random initialization on the U.S. Congressional Floor Debates dataset offers surprisingly good performance.

Table 4.1: Summary of the experimental results for the Movie Reviews datasets using $SVM^{sle}$, $SVM^{sle}$ w/ Prior and $SVM^{sle}_{fs}$ with and without proximity features.

| Methods | Random 30% | Last 30% | OpinionFinder | Annot. Rationales |
|---|---|---|---|---|
| $SVM^{sle}$ | 87.22 | 89.72 $*$ | 91.28 $\star$ | 91.61 $\star$ |
| $SVM^{sle}$+ Prox.Feat. | 85.44 | 88.83 | 90.89 $\star$ | 92.00 $\star$ |
| $SVM^{sle}$w/ Prior | 87.61 | 90.50 $*$ | 91.72 $\star$ | 92.67 $\star$ |
| $SVM^{sle}$+ Prox.Feat. | 87.56 | 90.00 $*$ | 93.22$\star$ | 92.00 $\star$ |
| $SVM^{sle}_{fs}$ | 89.50 | 91.06 $\star$ | 92.50$\star$ | 92.39 $\star$ |
| $SVM^{sle}_{fs}$+ Prox.Feat. | 88.22 | 91.22 $\star$ | 92.39$\star$ | 93.22 $\star$ |

– For Movie Reviews, the SVM baseline accuracy is 88.56%. A $\star$ (or $*$) indicates statically significantly better performance than baseline according to the paired t-test with $p < 0.001$ (or $p < 0.05$).

Table 4.2: Summary of the experimental results for the U.S. Congressional Floor debates datasets using $SVM^{sle}$, $SVM^{sle}$ w/ Prior and $SVM^{sle}_{fs}$ with and without proximity features.

| Methods | Random 30% | Last 30% | OpinionFinder |
|---|---|---|---|
| $SVM^{sle}$ | 78.84 | 73.26 | 77.33 |
| $SVM^{sle}$+ Prox.Feat. | 73.14 | 73.95 | 79.53 |
| $SVM^{sle}$w/ Prior | 78.49 | 71.51 | 77.09 |
| $SVM^{sle}$+ Prox.Feat. | 76.40 | 73.60 | 78.60 |
| $SVM^{sle}_{fs}$ | 77.33 | 67.79 | 77.67 |
| $SVM^{sle}_{fs}$+ Prox.Feat. | 73.84 | 73.37 | 77.09 |

– For U.S. Congressional Floor Debates, the SVM baseline accuracy is 70.00%. Statistical significance cannot be calculated because the data comes in train/validation/test split, not folds.

described in Section 4.3.4) in general seems to improve performance when using a good initialization, and hurts performance otherwise.

Tables 4.3 and 4.4 show a comparison of $SVM^{sle}_{fs}$ with previous work on the Movie Reviews and U.S. Congressional Floor Debates datasets, respectively. For

Table 4.3: Comparison of $SVM_{fs}^{sle}$ with previous work on the Movie Reviews dataset. We considered two settings: when human annotations are available (Annot. Labels), and when they are unavailable (No Annot. Labels).

| | METHOD | ACC |
|---|---|---|
| Baseline | SVM | 88.56 |
| Annot. Labels | Zaidan et al. (2007) | 92.20 |
| | $SVM_{fs}^{sle}$ | 92.28 |
| | $SVM_{fs}^{sle}$+ Prox.Feat. | 93.22 |
| No Annot. Labels | Yessenalina et al. (2010a) | 91.78 |
| | $SVM_{fs}^{sle}$ | 92.50 |
| | $SVM_{fs}^{sle}$+Prox.Feat. | 92.39 |

Table 4.4: Comparison of $SVM_{fs}^{sle}$ with previous work on the U.S. Congressional Floor Debates dataset for the speaker-based segment classification task.

| | METHOD | ACC |
|---|---|---|
| Baseline | SVM | 70.00 |
| Prior work | Thomas et al. (2006) | 71.28 |
| | Bansal et al. (2008) | 75.00 |
| Our work | $SVM_{fs}^{sle}$ | 77.67 |
| | $SVM_{fs}^{sle}$+ Prox.Feat. | 77.09 |

the Movie Reviews dataset, we considered two settings: when human annotations are available, and when they are not (in which case we initialized using OpinionFinder). For the U.S. Congressional Floor Debates dataset we used only the latter setting, since there are no annotations available for this dataset. In all cases we observe $SVM_{fs}^{sle}$ showing improved performance compared to previous results.

**Training details.** We tried around 10 different values for $C$ parameter, and selected the final model based on the validation set. The training procedure al-

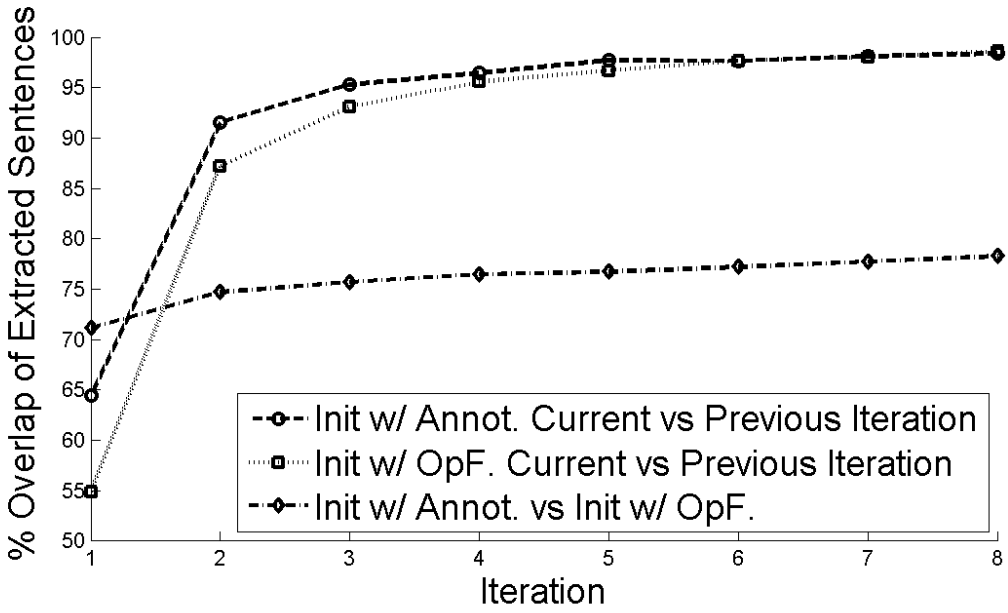


Figure 4.1: Overlap of extracted sentences from different $SVM_{fs}^{sle}$ models on the Movie Reviews training set.

ternates between training a standard structural SVM model and using the subsequent model to re-label the latent variables. We selected the halting iteration of the training procedure using the validation set. When initializing using human annotations for the Movie Reviews dataset, the halting iteration is typically the first iteration, whereas the halting iteration is typically chosen from a later iteration when initializing using OpinionFinder.

Figure 4.1 shows the per-iteration overlap of extracted sentences from $SVM_{fs}^{sle}$ models initialized using OpinionFinder and human annotations on the Movie Reviews training set. We can see that training has approximately converged after about 10 iterations.[5] We can also see that both models iteratively learn to extract sentences that are more similar to each other than their respective initializations (the overlap between the two initializations is 57%). This is an

[5]The number of iterations required to converge is an upper bound on the number of iterations from which to choose the halting iteration (based on a validation set).

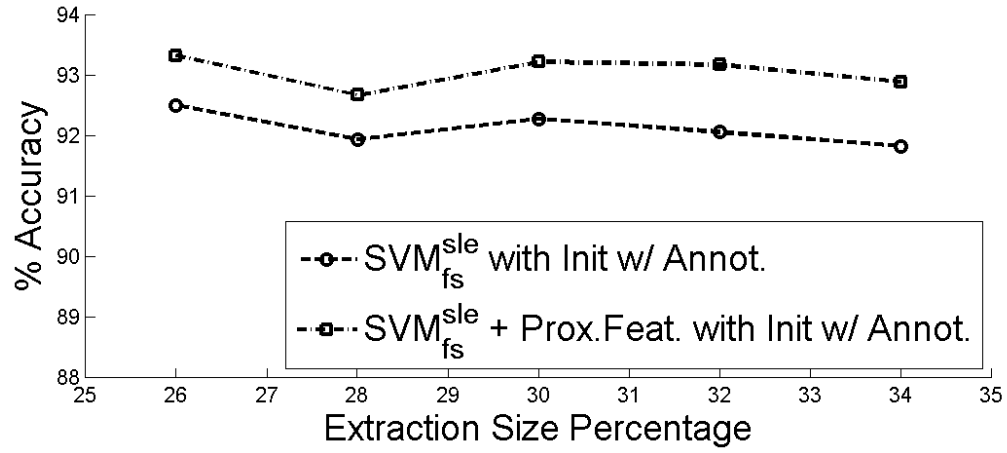


Figure 4.2: Test accuracy on the Movie Reviews dataset for $SVM_{fs}^{sle}$ while varying extraction size.

indicator that our learning problem, despite being non-convex and having multiple local optima, has a reasonably large "good" region that can be approached using different initialization methods.

**Varying the extraction size.** Figure 4.2 shows how accuracy on the test set of $SVM_{fs}^{sle}$ changes on the Movie Reviews dataset as a function of varying the extraction size $f(|x|)$ from (4.5). We can see that performance changes smoothly[6] (and so is robust), and that one might see further improvement from more careful tuning of the size constraint.

**Examining an example prediction.** Our proposed methods are not designed to extract interpretable explanations, but examining the extracted explanations might still yield meaningful information. Table 4.5 contains an example speech from the U.S. Congressional Floor Debates test set, with Latent Explanations found by $SVM_{fs}^{sle}$ highlighted in boldface. This speech was made in support of the Stem Cell Research Enhancement Act. For comparison, Table 4.5 also shows

[6]The smoothness will depend on the initialization.

Table 4.5: "yea" speech with ***Latent Explanations*** from the U.S. Congressional Floor Debates dataset predicted by $SVM_{fs}^{sle}$ with OpinionFinder initialization. Latent Explanations are preceded by solid circles with numbers denoting their preference order (1 being most preferred by $SVM_{fs}^{sle}$). The five least subjective sentences are preceded by circles with numbers denoting the subjectivity order (1 being least subjective according to $SVM_{fs}^{sle}$).

❷ ***Mr. Speaker, I am proud to stand on the house floor today to speak in favor of the Stem Cell Research Enhancement Act, legislation which will bring hope to millions of people suffering from disease in this nation.*** ❸ ***I want to thank Congresswoman Degette and Congressman Castle for their tireless work in bringing this bill to the house floor for a vote.***

① The discovery of embryonic stem cells is a major scientific breakthrough. ⑤ Embryonic stem cells have the potential to form any cell type in the human body. This could have profound implications for diseases such as Alzheimer's, Parkinson's, various forms of brain and spinal cord disorders, diabetes, and many types of cancer. ② According to the Coalition for the Advancement of Medical Research, there are at least 58 diseases which could potentially be cured through stem cell research.

That is why more than 200 major patient groups, scientists, and medical research groups and 80 Nobel Laureates support the Stem Cell Research Enhancement Act. ③ They know that this legislation will give us a chance to find cures to diseases affecting 100 million Americans. I want to make clear that I oppose reproductive cloning, as we all do. I have voted against it in the past. ❹ ***However, that is vastly different from stem cell research and as an ovarian cancer survivor, I am not going to stand in the way of science.***

Permitting peer-reviewed Federal funds to be used for this research, combined with public oversight of these activities, is our best assurance that research will be of the highest quality and performed with the greatest dignity and moral responsibility. The policy President Bush announced in August 2001 has limited access to stem cell lines and has stalled scientific progress.

As a cancer survivor, I know the desperation these families feel as they wait for a cure. ④ This congress must not stand in the way of that progress. ❺ ***We have an opportunity to change the lives of millions, and I hope we take it.*** ❶ ***I urge my colleagues to support this legislation.***

the five least subjective sentences according to $SVM_{fs}^{sle}$, that are underlined. Notice that most of these "objective" sentences can plausibly belong to speeches made in opposition to bills that limit stem cell research funding. That is, they

do not clearly indicate the speaker's stance towards the specific bill in question. We can thus see that our approach can indeed learn to infer sentences that are essential to understanding the document-level sentiment.

## 4.5 Discussion

Making good structural assumptions simplifies the development process. Compared to methods that modify the training of flat document classifiers (e.g., Zaidan et al. (2007)), our approach uses fewer parameters, leading to a more compact and faster training stage. Compared to methods that use a cascaded approach (e.g., Pang and Lee (2004)), our approach is more robust to errors in the lower-level subtask due to being a joint model.

Introducing latent variables makes the training procedure more flexible by not requiring lower-level labels, but does require a good initialization (i.e., a reasonable substitute for the lower-level labels). We believe that the widespread availability of off-the-shelf sentiment lexicons and software, despite being developed for a different domain, makes this issue less of a concern, and in fact creates an opportunity for approaches like ours to have real impact.

One can incorporate many types of sentence-level information that cannot be directly incorporated into a flat model. Examples include scores from another sentence-level classifier (e.g., from Nakagawa et al. (2010)) or combining phrase-level polarity scores (e.g., from Choi and Cardie (2008)) for each sentence, or features that describe the position of the sentence in the document.

Most prior work on the U.S. Congressional Floor Debates dataset focused on using relationships between speakers such as agreement (Thomas et al. (2006), Bansal et al. (2008)), and used a global min-cut inference procedure. However, they require all test instances to be known in advance (i.e., their formulations

are transductive). Our method is not limited to the transductive setting, and instead exploits a different and complementary structure: the latent explanation (i.e., only some sentences in the speech are indicative of the speaker's vote).

In a sense, the joint feature structure used in our model is the simplest that could be used. Our model makes no explicit structural dependencies between sentences, so the choice of whether to extract each sentence is essentially made independently of other sentences in the document. More sophisticated structures can be used if appropriate. For instance, one can formulate the sentence extraction task as a sequence labeling problem similar to McDonald et al. (2007), or use a more expressive graphical model such as in Pang and Lee (2004), Thomas et al. (2006). So long as the global inference procedure is tractable or has a good approximation algorithm, then the training procedure is guaranteed to converge with rigorous generalization guarantees (Finley and Joachims (2008)). Since any formulation of the extraction subtask will suppress information for the main document-level task, one must take care to properly incorporate smoothing if necessary.

Another interesting direction is training models to predict not only sentiment polarity, but also whether a document is objective. For example, one can pose a three class problem ("positive", "negative", "objective"), where objective documents might not necessarily have a good set of informative (explanatory) sentences, similar to Chang et al. (2010).

## 4.6 Summary of the Chapter

In this chapter we presented latent variable structured models for the document-level sentiment classification task. These models do not rely on sentence-level annotations, and are trained jointly (over both the document and

sentence levels) to directly optimize document-level accuracy. Experiments on two standard sentiment analysis datasets showed improved performance over previous results.

Our approach can, in principle, be applied to any classification task that is well modeled by jointly solving an extraction subtask. However, as evidenced by our experiments, proper training does require a reasonable initial guess of the extracted informative sentences, as well as ways to mitigate the risk of the extraction subtask suppressing too much information (such as via feature smoothing).

CHAPTER 5

# COMPOSITIONAL MATRIX-SPACE MODELS FOR PHRASE-LEVEL SENTIMENT CLASSIFICATION

In this chapter we consider the task of phrase-level sentiment classification. As described in Chapter 1, humans use quite different intuitions for deciding the sentiment of a phrase or a sentence, compared to deciding the sentiment of a document. In this chapter, we exploit the compositional semantic structure of phrases to improve phrase-level sentiment classification. More specifically we learn matrix-space word representations that are compositional in nature and model composition as matrix multiplication.

As described in Chapter 2, work in the sentiment analysis area ranges from identifying the sentiment of individual words to determining the sentiment of phrases, sentences and documents. The bulk of previous research, however, models just positive vs. negative sentiment, collapsing positive (or negative) words, phrases and documents of differing intensities into just one positive (or negative) class. For word-level sentiment, therefore, these methods would not recognize a difference in sentiment between words like "good" and "great", which have the same direction of polarity (i.e., positive) but different intensities. At the phrase level, the methods will fail to register compositional effects in sentiment brought about by intensifiers like "very", "absolutely", "extremely", etc. "Happy" and "very happy", for example, will both be considered simply "positive" in sentiment. In real-world settings, on the other hand, sentiment values extend across a polarity spectrum — from very negative, to neutral, to very positive. Recent research has shown, in particular, that modeling intensity at the phrase level is important for real-world natural language processing tasks including question answering and textual entailment (de Marneffe et al. (2010)).

This chapter describes a general approach for phrase-level sentiment analysis that takes these real-world requirements into account: *we adopt a five-level ordinal sentiment scale and present a learning-based method that assigns ordinal sentiment scores to phrases*. Importantly, our approach will also be explicitly *compositional*[1] in nature so that it can accurately account for critical interactions among the words in each sentiment-bearing phrase.

The vast majority of methods for phrase-level and sentence-level sentiment analysis do not tackle the task compositionally: they, instead, employ a bag-of-words representation and, at best, incorporate additional features to account for negators, intensifiers, and for contextual valence shifters, which can change the sentiment over neighboring words (e.g., Polanyi and Zaenen (2004), Wilson et al. (2005b), Kennedy and Inkpen (2006), Shaikh et al. (2007)).

A notable exception is work by Moilanen and Pulman (2007), who propose a compositional semantic approach to assign a positive or negative sentiment to newspaper article titles. However, their knowledge-based approach presupposes the existence of a sentiment lexicon and a set of symbolic compositional rules.

But learning-based compositional approaches for sentiment analysis also exist. Choi and Cardie (2008), for example, propose an algorithm for phrase-based sentiment analysis that learns proper assignments of intermediate sentiment decision variables given the *a priori* (i.e., out of context) polarity of the words in the phrase and the (correct) phrase-level polarity. As in Moilanen and Pulman (2007), semantic inference is based on (a small set of) hand-written compositional rules. In contrast, Nakagawa et al. (2010) use a dependency parse tree to guide the learning of compositional effects. Each of the above, however, uses a

[1] As described in Chapter 1 and Chapter 2, the *Principle of Compositionality* asserts that the meaning of a complex expression is a function of the meanings of its constituent expressions and the rules used to combine them.

binary rather than an ordinal sentiment scale.

In contrast, our proposed method for phrase-level sentiment analysis is inspired by recent work on distributional approaches to compositionality. In particular, Baroni and Zamparelli (2010) tackle adjective-noun compositions using a vector representation for nouns and *learning* a matrix representation for each adjective. The adjective matrices are then applied as functions over the meanings of nouns — via matrix-vector multiplication — to derive the meaning of adjective-noun combinations. Rudolph and Giesbrecht (2010) show theoretically, that multiplicative matrix-space models are a general case of vector-space models and furthermore exhibit desirable properties for semantic analysis: they take into account word order and are reasonable from algebraic and cognitive perspectives. Their work, however, does not present an algorithm for learning such models; nor does it provide empirical evidence in favor of matrix-space models over vector-space models.

In this chapter, we propose a *learning-based* approach to assign *ordinal sentiment scores* to sentiment-bearing phrases using a general *compositional matrix-space model of language*. All words are modeled as matrices, independent of their part-of-speech, and compositional inference is uniformly modeled as matrix multiplication. To predict an ordinal scale sentiment value, we employ Ordered Logistic Regression, introducing a novel training algorithm to accommodate our compositional matrix-space representations. To our knowledge, this is the first such algorithm for learning matrix-space models for semantic composition. We evaluate the approach on a standard sentiment corpus (Wiebe et al. (2005)), making use of its manually annotated phrase-level annotations for polarity and intensity, and compare our approach to the more commonly employed bag-of-words model. We show that our matrix-space model significantly outperforms

a bag-of-words model for the ordinal scale sentiment prediction task.

**Roadmap of the Chapter.** The work described in this chapter is based on Yessenalina and Cardie (2011). We start by describing the compositional effects in sentiment analysis tasks (Section 5.1). We present the model in Section 5.2 and experimental methodology in Section 5.3, then proceed with the discussion of the experimental results in Section 5.4. Section 5.5 describes related work in distributional similarity and compositionality. We discuss the model in Section 5.6 and summarize the chapter in Section 5.7.

## 5.1 Compositional Effects in Sentiment Analysis

To motivate our compositional model for phrase-level sentiment classification, we start by describing the compositional effects in sentiment analysis that we would like to model. We discuss compositional effects in sentiment analysis using a few examples of combining polar adjectives with adverbs, including negators.

First, consider combining an adverb like "very" with a polar adjective like "good". "Good" has an *a priori* positive sentiment, so "very good" should be considered **more** positive even though "very", on its own, does not bear sentiment. Combining "very" with a negative adjective, like "bad", results in a phrase ("very bad") that should be characterized as more negative than the original adjective. Thus, it is convenient to think of the effect of combining an intensifying adverb with a polar adjective as being *multiplicative* in nature, if we assume the adjectives ("good" and "bad") to have positive and a negative sentiment scores, respectively.

Next, let us consider adverbial negators, e.g., "not", combined with polar

adjectives. When modeling only positive and negative labels for sentiment, negators are generally treated as flipping the polarity of the adjective it modifies (Choi and Cardie (2008), Nakagawa et al. (2010)). However, recent work (Taboada et al. (2011), Liu and Seneff (2009)) suggests that the effect of the negator when ordinal sentiment scores are employed is more akin to dampening the adjective's polarity rather than flipping it. For example, if "perfect" has a strong positive sentiment, then the phrase "not perfect" is still positive, though to a lesser degree. And while "not terrible" is still negative, it is less negative than "terrible". For these cases, it is convenient to view "not" as shifting polarity to the opposite side of polarity scale by some value, which is essentially is an *additive* effect.

There are, of course, more interesting examples of compositional semantic effects on sentiment: e.g., *prevent cancer*, *ease the burden*. Here, the verbs *prevent* and *ease* act as content-word negators (Choi and Cardie (2008)) in that they modify the negative sentiment of their direct object arguments so that the phrase as a whole is perceived as somewhat positive.

We want to model both additive and multiplicative compositional effects for phrase-level sentiment classification task. Our proposed matrix-space model accounts for both of these effects.

## 5.2 The Model for Ordinal Scale Sentiment Prediction

As described above, our task is to predict an ordinal scale sentiment label for a phrase. To this end, we employ a sentiment scale with five ordinal values: VERY NEGATIVE, NEGATIVE, NEUTRAL, POSITIVE and VERY POSITIVE. Given a set of phrases with their gold standard ordinal sentiment labels as training examples, we then use an Ordered Logistic Regression (OLogReg) model for prediction.

Unfortunately, our matrix-space representation precludes doing this directly.

We have chosen OLogReg, as opposed to say PRanking (Crammer and Singer (2001)), because optimization of the former is more attractive: the objective (likelihood) is smooth and the gradients are continuous. As will become clear shortly, learning our models is not trivial and it is important to use sophisticated off-the-shelf optimizers such as L-BFGS.

For a bag-of-words model, OLogReg learns one weight for each word and a set of thresholds by maximizing the likelihood of the training data. Typically, this is accomplished by using an optimizer like L-BFGS whose interface needs the value and gradient of the likelihood with respect to the parameters at their current values. In the next subsections, we instantiate OLogReg for our sentiment prediction task using a matrix-space word model (Sections 5.2.1 and 5.2.2) and a bag-of-words model (Sections 5.2.3). The learning formulation of bag-of-words OLogReg is convex therefore we will get to the global optimum; in contrast, the optimization problem for matrix-space model is non-convex, it is important to initialize the model well. Initialization of the matrix-space model is discussed in Section 5.2.4.

### 5.2.1 Notation

In the subsequent subsections we will use the following notation. Let $n$ be the number of phrases in the training set and let $d$ be the number of words in the dictionary. Let $x^i$ be the $i$-th phrase and $y^i$ would be the label of $x^i$, where $y^i$ takes $r$ different values $y^i \in \{0, \ldots, r-1\}$. Then $|x^i|$ will denote the length of the phrase $x^i$, and the words in $i$-th phrase are: $x^i = x^i_1, x^i_2, \ldots, x^i_{|x^i|}$; $x^i_j, 1 \leq j \leq |x^i|$ is the $j$-th word of $i$-th phrase; where $x^i_j$ is from the dictionary: $1 \leq x^i_j \leq d$.

In the case of the bag-of-words model, $\Phi(x^i) \in \mathbb{R}^d$ is the representation of the $i$-th phrase. $\Phi_j(x^i)$ counts the number of times the $j$-th word from the dictionary appears in the $i$-th phrase. Given a $w \in \mathbb{R}^d$ it assigns a score $\xi_i$ to a phrase $x^i$ by

$$\xi_i = w^T \Phi(x^i) = \sum_{j=1}^{|x^i|} w_{x_j^i} \tag{5.1}$$

In the case of the matrix-space model the $\Phi(x^i) \in \mathbb{R}^{|x^i| \times d}$ is the representation of the $i$-th phrase. $\Phi_{jk}(x^i)$ is 1, if $x_j^i$ is the $k$-th word in the dictionary, and zero otherwise. Given $u, v \in \mathbb{R}^m$ and a set of matrices $\{W_p \in \mathbb{R}^{m \times m}\}_{p=1}^d$, one for each word, it assigns a score $\xi_i$ to a phrase $x^i$ by

$$\begin{aligned} \xi_i &= u^T \left( \prod_{j=1}^{|x^i|} \sum_{k=1}^{d} W_k \Phi_{jk}(x^i) \right) v \\ &= u^T \left( \prod_{j=1}^{|x^i|} W_{x_j^i} \right) v \end{aligned} \tag{5.2}$$

where $\prod_{j=1}^{|x^i|} W_{x_j^i} = W_{x_1^i} W_{x_2^i} \cdots W_{x_{|x^i|}^i}$ in **exactly this order**. We choose to map matrices to the real numbers by using vectors $u$ and $v$ from $\mathbb{R}^{m \times 1}$; so that $\xi = u^T M v$, where $M \in \mathbb{R}^{m \times m}$, which is sensitive to the order of matrices, i.e., $u^T M_1 M_2 v \neq u^T M_2 M_1 v$. Note, that care must be taken in choosing how to map a matrix to a real number. For example, another way to map matrices to the real numbers is to use the determinant of a matrix; however, the determinant is not sensitive to the word order: $det(M_1 M_2) = det(M_1) det(M_2) = det(M_2 M_1)$; which is not desirable for a model like ours that needs to account for word order.

**Modeling composition.** A $m \times m$ matrix, representing a word, can be considered as a linear function, mapping from $\mathbb{R}^m$ to $\mathbb{R}^m$. Composition of words is modeled by function composition, in our case composition of linear functions, i.e., matrix multiplication. Note, that unlike bag-of-words model, the matrix-

space model takes word order into account, since matrix multiplication is not commutative operation.

### 5.2.2 Ordered Logistic Regression

Now we will describe our objective function for OLogReg and its derivatives. OLogReg has $r-1$ thresholds ($\kappa_0, \dots \kappa_{r-2}$), so introducing $\kappa_{-1} = -\infty$ and $\kappa_{r-1} = \infty$ leads to the unified expression for posterior probabilities for all values of $k$:

$$\begin{aligned} P(y^i = k|x) &= P(\kappa_{k-1} < \xi_i \leq \kappa_k) \\ &= F(\kappa_k - \xi_i) - F(\kappa_{k-1} - \xi_i) \end{aligned}$$

$F(x)$ is an inverse-logit function

$$F(x) = \frac{e^x}{1 + e^x}$$

this is its derivative:

$$\frac{dF(x)}{dx} = F(x)(1 - F(x))$$

Therefore the negative loglikelihood of the training data will look like the following (Hardin and Hilbe (2007)):

$$L = -\sum_{i=1}^{n} \sum_{k=0}^{r-1} \ln(F(\kappa_k - \xi_i) - F(\kappa_{k-1} - \xi_i)) I(y^i = k)$$

where $r$ is the number of ordinal classes, $\xi_i$ is the score of $i$-th phrase, $I$ is the indicator function that is equal to 1, when $y^i = k$, and zero otherwise. We need to minimize the objective $L$ with respect to the following constraints:

$$\kappa_{k-1} \leq \kappa_k, \;\; 1 \leq k \leq r-2 \tag{5.3}$$

(The constraints are similar to the ones in PRank algorithm). For ease of optimization we parametrize our model via $\kappa_0$, and $\tau_j, 1 \leq j \leq r-2$:

$\kappa_{-1} = -\infty$,

$\kappa_0$,

$\kappa_1 = \kappa_0 + \tau_1$,

$\kappa_2 = \kappa_0 + \sum_{j=1}^{2} \tau_j$,

$\ldots$,

$\kappa_{r-2} = \kappa_0 + \sum_{j=1}^{r-2} \tau_j$

$\kappa_{r-1} = \infty$,

where $\tau_1$, ..., $\tau_{r-2}$ are non-negative values, that represent how far the corresponding thresholds are from each other. Then the constraints (5.3) would be:

$$\tau_j \geq 0, \;\; 1 \leq j \leq r-2 \tag{5.4}$$

To simplify the equations we can rewrite the negative loglikelihood as follows:

$$L = -\sum_{i=1}^{n} \sum_{k=0}^{r-1} \ln(A_{ik} - B_{ik}) I(y^i = k) \tag{5.5}$$

where

$$A_{ik} = \begin{cases} F(\kappa_0 + \sum_{j=1}^{k} \tau_j - \xi_i), & \text{if } k = 0, \ldots, r-2 \\ 1, & \text{if } k = r-1 \end{cases}$$

$$B_{ik} = \begin{cases} 0, & \text{if } k = 0 \\ F(\kappa_0 + \sum_{j=1}^{k-1} \tau_j - \xi_i), & \text{if } k = 1, \ldots, r-1 \end{cases}$$

Let's introduce $L_{ik} = -\ln(A_{ik} - B_{ik}) I(y^i = k)$ and then the derivative of $L_{ik}$ with respect to $\kappa_0$ will be:

$$\begin{aligned} \frac{\partial L_{ik}}{\partial \kappa_0} &= \frac{-[A_{ik}(1 - A_{ik}) - B_{ik}(1 - B_{ik})]}{A_{ik} - B_{ik}} I(y^i = k) \\ &= (A_{ik} + B_{ik} - 1) I(y^i = k) \end{aligned}$$

For $j = y^i$:

$$\frac{\partial L_{ik}}{\partial \tau_j} = \frac{-A_{ik}(1 - A_{ik})}{A_{ik} - B_{ik}} I(y^i = k)$$

For all $j < y^i$:

$$\frac{\partial L_{ik}}{\partial \tau_j} = (A_{ik} + B_{ik} - 1) I(y^i = k)$$

For all $j > y^i$: $\frac{\partial L_{ik}}{\partial \tau_j} = 0$. The derivative with respect to the score $\xi_i$ is:

$$\frac{\partial L_{ik}}{\partial \xi_i} = (-A_{ik} - B_{ik} + 1) I(y^i = k) \tag{5.6}$$

**Matrix-Space Word Model**

Here we compute the derivatives with respect to a word. For the OLogReg model with matrix-space word representations, we have:

$$\frac{\partial L}{\partial W_{x_j^i}} = \frac{\partial L}{\partial \xi_i} \cdot \frac{\partial \xi_i}{\partial W_{x_j^i}}$$

The expression for $\frac{\partial L}{\partial \xi_i}$ is given in (5.6); we will derive $\frac{\partial \xi_i}{\partial W_{x_j^i}}$ from (5.2). In the case of the Matrix-Space word model, each word is represented as an $m \times m$ affine matrix $W$:

$$W = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \tag{5.7}$$

We choose the class of **affine matrices** since for affine matrices matrix multiplication represents both operations: linear transformation and translation. Linear transformation is important for modeling changes in sentiment, translation is also useful (we also make use of a translation vector during initialization, see Section 5.2.4). In this thesis we consider $m \geq 3$ since we want the matrix $A$ from (5.7) to represent rotation and scaling. Applying the affine transformation $W$ to vector $[x, 1]^T$ is equivalent to applying linear transformation $A$ and translation $b$ to $x$. [2]

[2]
$$\begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} Ax + b \\ 1 \end{pmatrix}$$
where $A$ is a linear transformation, $b$ is a translation vector.
Also the product of affine matrices is an affine matrix.

Though vectors $u$ and $v$ can be learned together with word matrices $W_j$, **we choose to fix $u$ and $v$.** The main intuition behind fixing $u$ and $v$ is *to reduce the degrees of freedom of the model*: different assignments of $u$, $v$ and $W_j$-s can lead to the same score $\xi$, i.e., there exist $\hat{u}$, $\hat{v}$ and $\hat{W}_j$-s different from $u$, $v$ and $W_j$-s respectively, such that $\xi(u, v, W)$ would be equal to $\xi(\hat{u}, \hat{v}, \hat{W})$. The specific choice of $u$ and $v$ leads to an equivalent model for all $\hat{u}$ and $\hat{v}$ such that $\hat{u} = M^T u$, $\hat{v} = M^{-1} v$, where $M$ is any invertible transformation (i.e., $\hat{u}$, $\hat{v}$ are derived from $u$, $v$ by applying linear transformations $M^T$, $M^{-1}$ respectively):

$$\begin{aligned} u^T W_1 W_2 v &= (u^T M)(M^{-1} W_1 M)(M^{-1} W_2 M)(M^{-1} v) \\ &= \hat{u}^T \hat{W}_1 \hat{W}_2 \hat{v} \end{aligned}$$

The derivative of the phrase $\xi_i$ with respect to $j$-th word $W_j$ would be (for brevity we drop the phrase index and $W_j$ refers to $W_{x_j^i}$ and $p$ refers to $|x_i|$):

$$\begin{aligned} \frac{\partial \xi_i}{\partial W_j} &= \left( \frac{\partial u^T W_1 W_2 \dots W_p v}{\partial W_j} \right) \\ &= \left[ (u^T W_1 \dots W_{j-1})^T (W_{j+1} \dots W_p v)^T \right] \\ &= \left[ (W_{j-1}^T \dots W_1^T)(u v^T)(W_p^T \dots W_{j+1}^T) \right] \end{aligned}$$

(see Petersen and Pedersen (2008)).

In case if a certain word appears multiple times in the phrase, the derivative with respect to that word would be a sum of derivatives with respect to each appearance of a word, while all other appearances are fixed. For example,

$$\left( \frac{\partial u^T W W_1 W v}{\partial W} \right) = u(W_1 W v)^T + (u^T W W_1)^T v^T$$

where $W$ is a representation of a word that is repeated.

So given the expression (5.6) for $\frac{\partial L}{\partial \xi_i}$, the derivative with respect to each word can be computed. Notice that the update for the $j$-th word in a sentence depends on the order of the words, which is in line with our desire to account for word order.

### Optimization

The goal of the training procedure is for the $i$-th phrase with $p$ words $x_1 x_2 \dots x_p$ to learn word matrices $W_1$, $W_2$, ..., $W_p$ and thresholds $\kappa_0$, $\tau_1$, ... $\tau_{r-2}$ such that resulting $\xi_i$-s will lead to the lowest negative loglikelihood. So, given the negative loglikelihood and the derivatives with respect $\kappa_0$ and $\tau_j$-s and word matrices $W$, we optimize objective (5.5) subject to $\tau_j \geq 0$. We use L-BFGS-B (Large-scale Bound-constrained Optimization) by Byrd et al. (1995) as an optimizer.

### Regularization in Matrix-Space Model

In order to make sure that the L-BFGS-B updates do not cause numerical issues we perform the following regularization to the resulting matrices. An $m$ by $m$ matrix $W_j$ that can be represented as:

$$W_j = \begin{pmatrix} A_{11} & a_{12} \\ a_{21}^T & a_{22} \end{pmatrix}$$

where $A_{11} \in \mathbb{R}^{m-1 \times m-1}$, $a_{12}, a_{21} \in \mathbb{R}^{m-1 \times 1}$, $a_{22} \in \mathbb{R}$. First make the matrix affine by updating the last row, then the updated matrix will look like:

$$\hat{W}_j = \begin{pmatrix} A_{11} & a_{12} \\ 0 & 1 \end{pmatrix}$$

It can be proven that such a projection returns the closest affine matrix in Frobenius norm.

However, we also want to regularize the model to avoid ill-conditioned matrices. Ill-conditioned matrices represent transformations whose output is very sensitive to small changes in the input and therefore they have a similar effect to having large weights in a bag-of-words model. To perform such a regularization we "shrink" the singular values of $A_{11}$ towards one. More specifically,

**Algorithm 3** Training Algorithm for Matrix-Space OLogReg

```
Input: {(x_1, y_1), ..., (x_n, y_n)} //training data
Input: h //projection parameter
Input: T //number of iterations
Input: W, κ_0 and τ_j //initial values
for t = 1, ..., T do
    (W, κ_0, τ_j)=minimize L using L-BFGS-B
    for i = 1, ..., d do
        W_i=Project(W_i, h)
    end for
end for
Return W, κ_0, τ_j
```

we first use the Singular Value Decomposition (SVD) of the $A_{11}$: $U\Sigma V^T = A_{11}$, where $U$ and $V$ are orthogonal matrices, $\Sigma$ is a matrix with singular values on the diagonal. Then we update singular values in the following way to get $\tilde{\Sigma}$: $\tilde{\Sigma}_{ii} = \Sigma_{ii}^h$, where $h$ is a parameter between $0$ and $1$. If $h = 1$, then $\Sigma_{ii}$ remains the same. In the other extreme case, if $h = 0$, then $\Sigma_{ii}^h = 1$. For intermediate values of $h$ the singular values of $A_{11}$ would be brought closer to one. Finally, we recompute $\tilde{A}_{11}$: $\tilde{A}_{11} = U\tilde{\Sigma}V^T$. So, $\tilde{W}_j$ would be :

$$\tilde{W}_j = \begin{pmatrix} \tilde{A}_{11} & a_{12} \\ 0 & 1 \end{pmatrix}$$

**Learning in the Matrix-Space Model**

We use Algorithm 3 to learn the matrix-space model. What essentially happens is that we iterate two steps: optimizing the $W$ matrices using L-BFGS-B and the projection step. L-BFGS-B returns a solution that is not necessarily an affine matrix. After projecting to the space of affine matrices we start L-BFGS-B from a better initial point. In practice, the first few iterations lead to large decrease in negative loglikelihood.

### 5.2.3 Bag-Of-Words Model

In the bag-of-words model the score of the $i$-th phrase is given in (5.1). Therefore, the partial derivative with respect to $j$-th word in $i$-th phrase $\frac{\partial \xi_i}{\partial w_{x_j^i}}$ is equal to the number $c_j$ of times $x_i^j$ appears in $x^i$, so:

$$\frac{\partial L}{\partial w_{x_j^i}} = \frac{\partial L}{\partial \xi_i} \cdot c_j$$

**Optimization.** We minimize negative loglikelihood using L-BFGS-B subject to $\tau_j \geq 0$.

**Regularization.** To prevent overfitting for bag-of-words model we regularize $w$. The $L_2$-regularized negative loglikelihood will consist of the expression in (5.5) and an additional term $\frac{\lambda}{2}||w||_2^2$, where $||\cdot||_2$ is the $L_2$-norm of a vector. The derivative of the additional term with respect to $w$ will be:

$$\frac{\partial \frac{\lambda}{2}||w||_2^2}{\partial w} = \lambda w$$

Hence the partial derivative with respect to $w_{x_j^i}$ will have an additional term $\lambda w_{x_j^i}$.

### 5.2.4 Initialization

**Initialization of bag-of-words OLogReg.** We initialize the weight for each word with zero and $\kappa_0$ with a random number and $\tau_j$-s with non-negative random numbers. Since the learning problem for bag-of-words OLogReg is convex, we will get the global optimum.

**Better Initialization of Matrix-Space Model.** Preliminary experiments showed that the Matrix-Space model needs a good initialization. Initializing

with different random matrices reaches different local minima and the quality of local minima depends on initialization. Therefore, it is important to initialize the model with a good starting point. *One way to initialize the Matrix-Space model is to use the weights learned by the bag-of-words model.* We use the following intuition. As noted in Section 5.2.2 applying transformation $A$ of affine matrix $W$ can model a linear transformation, while vector $b$ represents a translation. Since the matrix-space model can encode a vector-space model (Rudolph and Giesbrecht (2010)), we can initialize the matrices to exactly mimic the bag-of-words model. In order to do that we place the weight, learned by the bag-of-words model in the first component of $b$. Let's assume that $w_{x_1}$ and $w_{x_2}$ are the weights learned for two distinct words $x_1$ and $x_2$ respectively. To compute the polarity score of a phrase $x_1\ x_2$, the bag-of-words model sums the weights of these two words: $w_{x_1}$ and $w_{x_2}$. Now we want to have the same effect in matrix-space model. Here we assume $m = 3$.

$$Z = \begin{pmatrix} 1 & 0 & w_{x_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & w_{x_2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & w_{x_1} + w_{x_2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finally, there is a step of mapping matrix $Z$ to a number using $u$ and $v$, such that $\xi(Z) = w_{x_1} + w_{x_2}$. We also want vector $u$ and $v$ to be such that:

$$u^T \begin{pmatrix} 1 & 0 & w_{x_1} + w_{x_2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} v = w_{x_1} + w_{x_2} \tag{5.8}$$

The last equation can help us construct $u$ and $v$. We also set $u$ and $v$ to be orthogonal: $u^T v = 0$. So, we arbitrarily choose two orthogonal vectors for which

Table 5.1: Mapping of combination of polarities and intensities from MPQA dataset to our ordinal sentiment scale.

| Polarity | Intensity | Ordinal label |
|---|---|---|
| negative | high, extreme | 0 |
| negative | medium | 1 |
| neutral | high, extreme, medium | 2 |
| positive | medium | 3 |
| positive | high, extreme | 4 |

equation (5.8) holds: $u = [1, \sqrt{2}, 1]^T$ and $v = [1, -\sqrt{2}, 1]^T$.[3]

## 5.3 Experimental Methodology

For experimental evaluation of the proposed method we use the publicly available Multi-Perspective Question Answering (MPQA)[4] corpus (Wiebe et al. (2005)) version 1.2, which contains 535 newswire documents that are manually annotated with phrase-level subjectivity and intensity. We use the expression-level boundary markings in MPQA to extract phrases. We evaluate on positive, negative and neutral opinion expressions that have intensities "medium", "high" or "extreme".[5] The schematic mapping of phrase polarity and intensity values on ordinal sentimental scale is shown in Table 5.1.

### 5.3.1 Training Details

We perform 10-fold cross-validation on phrases extracted from the MPQA corpus: eight folds for training; one as a validation set; and one as test set. In total

[3] If $m > 3$, $u$ and $v$ can be set using the same intuition.
[4] http://www.cs.pitt.edu/mpqa/
[5] We ignored low-intensity phrases similar to Choi and Cardie (2008), Nakagawa et al. (2010).

there were 8022 phrases. Before training, we extract lemmas for each word. For evaluation we use $L_1$ loss: $\frac{1}{n}\sum_i |\hat{y}^i - y^i|$, where $\hat{y^i}$ is the prediction.

**Choice of dimensionality $m$.** The reported experiments are done by setting $m = 3$. Preliminary experiments with higher values of $m$ (5, 20, 50), did not lead to a better performance and increased the training time; therefore we did not use those values in our final experiments.

### 5.3.2 Methods

**PRank.** For each of the folds, we run 500 iterations of PRank and choose an early stopping iteration using a model that led to the lowest $L_1$ loss on the validation set; afterwards report the average performance on respective test sets.

**Bag-of-words OLogReg.** To prevent overfitting we search for the best regularization parameter among the following values of $\lambda$: $10^i$, from $10^{-4}$ to $10^4$. The lowest negative log-likelihood value on the validation set is attained for[6] $\lambda = 0.1$. With this value of $\lambda$ fixed, the final model is the one with the lowest negative loglikelihood on the training set.

**Matrix-space OLogReg+RandInit.** First, we initialized matrices with random numbers from normal distribution $N(0, 0.1)$ and set $u$ and $v$ as in Section 5.2.4, $T$ is set to 25. We run with two different random seeds and three different values for the parameter $h$: [0.1, 0.5, 0.9] and report the performance of the model that had the lowest negative loglikelihood on the validation set. The setting of $h$ that lead to the best model was $0.9$.

[6]We pick single $\lambda$ that gives best average validation set performance, and then use it to compute the average test set performance.

Table 5.2: $L_1$ loss for vector-space Ordered Logistic Regression and Matrix-Space Logistic Regression. $^\dagger$ Stands for a significant difference w.r.t. the Bag-Of-Words OLogReg model with p-value less than 0.001 ($p < 0.001$).

| Method | $L_1$ loss |
|---|---|
| PRank | 0.7808 |
| Bag-of-words OLogReg | 0.6665 |
| Matrix-space OLogReg+RandInit | 0.7417 |
| Matrix-space OLogReg+BowInit | $0.6375^\dagger$ |

**Matrix-space OLogReg+BowInit.** For the matrix-space models we initialize the model with the output of the regularized Bag-of-words OLogReg as described in Section 5.2.4, $T$ is set to 25. Then we use the training procedure described in Algorithm 3. We consider three different values for the parameter $h$ [0.1, 0.5, 0.9] and choose the model with the lowest validation set negative log-likelihood. The best setting of $h$ was $0.1$.

## 5.4 Results

We report $L_1$ loss for the four models in Table 5.2. The worst performance (denoted by the highest $L_1$ loss value) is obtained by PRank, followed by matrix-space OLogReg with random initialization. Bag-of-words OLogReg obtains quite good performance, and matrix-space OLogReg, initialized using the bag-of-words model performs the best, showing statistically significant improvements over the bag-of-words OLogReg model according to a paired t-test.

To see what the bag-of-word and matrix-space models are learning we performed inference on a few examples. In Table 5.3 we show the sentiment scores of the best performing bag-of-words **OLogReg** model and the best performing model based on matrices **Matrix-space OLogReg+BowInit**. By sentiment score, we mean equation (5.1) of Bag-of-words OLogReg and equation (5.2) of Matrix-

Table 5.3: Phrase and the sentiment scores of the phrase for 2 models Matrix-space OLogReg+BowInit and Bag-of-words OLogReg respectively. Notice that **relative ranking order what matters**.

| Phrase | Matrix-space OLogReg+BowInit | Bag-of-words OLogReg |
|---|---|---|
| not | -0.83 | -0.42 |
| very | 0.23 | 0.04 |
| good | 2.81 | 1.51 |
| very good | 3.53 | 1.55 |
| not good | -0.16 | 1.09 |
| not very good | 0.66 | 1.13 |
| bad | -1.67 | -1.42 |
| very bad | -2.01 | -1.38 |
| not bad | -0.54 | -1.85 |
| not very bad | -1.36 | -1.80 |

space OLogReg+BowInit.

Here we choose two popular adjectives "good" and "bad" that appeared in the training data, and examine the effect of applying the intensifier "very" on the sentiment score. As we can see, the matrix-space model learns a matrix for "very" that correctly intensifies both "bad" and "good" on the sentiment scale, i.e., $\xi(good) < \xi(very\ good)$ and $\xi(bad) < \xi(very\ bad)$, while the bag-of-words model gets the sentiment of "very bad" wrong: it is more positive than "bad". We also looked at the effect of combining "not" with these adjectives. The matrix-space model correctly encodes the effect of the negator for both positive and negative adjectives, such that $\xi(not\ good) < \xi(good)$ and $\xi(bad) < \xi(not\ bad)$. For the interesting case of applying a negator to a phrase with an intensifier, $\xi(not\ good)$ should be less than $\xi(not\ very\ good)$ and $\xi(not\ very\ bad)$ should be less than $\xi(not\ bad)$.[7] As shown in Table 5.3, these are predicted correctly by the matrix-space model, but the bag-of-words model misses the case of "bad".

[7]See the detailed discussion in Taboada et al. (2011) and Liu and Seneff (2009).

Also notice that since in the matrix-space model each word is represented as a function, more specifically a linear operator, and the function composition defined as matrix multiplication, we can think of ”not very” being an operator itself, that is a composition of operator ”not” and operator ”very”.

## 5.5 Related Work

The related work in the sentiment analysis area is discussed in Chapter 2.3. In this section we briefly overview related work in distributional semantics and compositionality.

**Distributional Semantics and Compositionality.** Research in the area of distributional semantics in NLP and Cognitive Science has looked at different word representations and different ways of combining words. Mitchell and Lapata (2010) propose a framework for vector-based semantic composition. They define composition as an additive or multiplicative function of two vectors and show that compositional approaches generally outperform non-compositional approaches that treat the phrase as the union of single lexical items.

Work by Baroni and Zamparelli (2010) models nouns as vectors in some semantic space and adjectives as matrices. It shows that modeling adjectives as linear transformations and applying those linear transformations to nouns results in final vectors for adjective-noun compositions that are close in semantic space to other similar phrases. The authors argue that modeling adjectives as a linear transformation is a better idea than using additive vector-space models. In their work, a separate matrix for each adjective is *learned* using the Partial Least Squares method in a completely unsupervised way. The recent work by Rudolph and Giesbrecht (2010), described in the introduction to this chapter, ar-

gues for plausibility of *multiplicative matrix-space* models. In contrast to work in semantics, our work is concerned with a specific dimension of word meaning — sentiment. Our techniques, however, are quite general and should be applicable to other problems in lexical semantics.

## 5.6 Discussion

Though in our model the order of composition is the same as the word order, we believe that a linguistically informed order of composition can give us further performance gains. For example, one can use the output of a dependency parser to guide the order of composition, similar to Nakagawa et al. (2010). Another possibility for improvement is to use the information about the scope of negation. In this thesis we assume the scope of negation to be the expression following the negation; in reality, however, determining the scope of negation is a complex linguistic phenomenon (Moilanen and Pulman (2007)). So the proposed model can benefit from identifying the scope of negation, similar to Councill et al. (2010).

Another possibility is to explore various ways of initialization of the matrix-space model. One interesting direction to explore might be to use non-negative matrix factorization (Lee and Seung (2001)), co-clustering techniques (Dhillon (2001)) to better initialize words that share similar contexts. The other possible direction is to use existing sentiment lexica and employing a "curriculum learning" strategy (Bengio et al. (2009), Kumar et al. (2010)) for our learning problem.

## 5.7 Summary of the Chapter

In the current chapter we presented a novel matrix-space model for ordinal scale sentiment prediction and an algorithm for learning such a model. The proposed model learns a matrix for each word; the composition of words is modeled as iterated matrix multiplication. The matrix-space framework with iterated matrix multiplication defines an elegant framework for modeling composition; it is also quite general. We use the matrix-space framework in the context of sentiment prediction, a domain where interesting compositional effects can be observed. The main focus of this chapter was to exploit compositional structure of the phrase by learning matrix-space word representations. One of the benefits of the proposed approach is that by learning matrices for words, the model can handle unseen word compositions (e.g., unseen bigrams) when the unigrams involved have been seen.

However, it is not trivial to learn a matrix-space model. Since the final optimization problem is non-convex, the initialization has to be done carefully. Here the weights learned in bag-of-words model come to rescue and provide good initial point for optimization procedure. The final model outperforms the bag-of-words based model, which suggests that this research direction is very promising.

CHAPTER 6

# CONCLUSIONS

In this thesis we addressed two important tasks in the sentiment analysis area: document-level and phrase-level sentiment classification. Here we summarize the contributions of our work and discuss future directions.

## 6.1 Summary of Contributions

**Incorporating automatically discovered informative sentences to improve document-level sentiment classification.** Informative sentences for document-level sentiment classification are those sentences that exhibit the same sentiment as the document, thus explain or support the document's sentiment label. We showed that informative sentences discovered automatically using sentiment analysis resources improve document-level sentiment classification, when incorporated in the form of additional constraints for an SVM classifier.

**Two-level joint structured model for document-level sentiment classification.** We further used automatically discovered informative sentences as latent variables in joint structured models. We explored two-level joint structured models for document-level sentiment classification; our final model does not require sentence-level sentiment annotations and directly optimizes document-level sentiment classification accuracy, using sentence-level information only to the extent necessary for solving the classification task. Our proposed model demonstrates improved performance on two publicly available datasets.

**Compositional matrix-space models for phrase-level sentiment classification.** We presented an algorithm for learning matrix-space models for phrase-level sentiment classification. The resulting model learns matrix-space word representations that are explicitly compositional and the composition is modeled as matrix multiplication. Our proposed model outperforms bag-of-words representation for phrase-level sentiment classification task.

## 6.2 Future Work

There are many different directions to extend our work. In this section we describe some of these.

**Rationales.** Using rationales could be beneficial for other classification tasks. For example, one interesting task to consider is deception detection (Ott et al. (2011)) — the task of classifying whether a review is deceptive or not. Ott et al. (2011) created a dataset of reviews with gold standard labels for this task. We hypothesize that after careful consideration of those reviews, human annotators could find explanatory text segments (rationales) that support the deception label of the review. Can we automatically identify rationales for this task?

**Structured models for document-level sentiment classification.** The structure used in the model that we proposed in Chapter 4 is a set of informative sentences. Instead, one can propose to represent the sentences in a document as a linear chain of sentence-level sentiment variables connected to the respective sentences, then identify informative sentences as in standard sequence-labeling task. This will capture interactions between neighboring sentences by using a more expressive graphical model as opposed to using proximity features as we

did in Chapter 4. The inference in such a model will be tractable since the Viterbi algorithm could be used to predict the best sequence of informative/non-informative sentences; and then the best sequence of informative sentences could be used as a structure for document-level sentiment classification.

Work by Thomas et al. (2006) on the U.S. Congressional floor debates dataset exploits the speaker agreement structure of the debates. In Chapter 4, we developed a model that exploits the structure of informative sentences, which is orthogonal to the speaker agreement structure used by Thomas et al. (2006). Potentially one can combine these orthogonal and complementary structures to further improve performance on the task of classifying speeches.

Another interesting research question is to consider the sentiment rating prediction task: instead of predicting just positive or negative label, predict an ordinal sentiment label. Will the structure of informative sentences as we defined it in this thesis be useful for this task? Or should we define informative sentences differently for this task?

One might also consider the task of classifying objective vs. positive vs. negative documents. In this setting the objective documents, as opposed to subjective ones, might not have a good set of informative explanatory sentences. Could we develop a model for this setting?

**Compositional matrix-space model for phrase-level sentiment classification.** In Chapter 5, we proposed a compositional matrix-space model for phrase-level sentiment classification. However, our learned model potentially can combine any words. It has been known that some word combinations are more probable than others. One way to extend our model is to incorporate language modeling as part of the learning objective, so that more plausible word combinations will get a higher score than less plausible ones.

An interesting research direction is to explore ways of reducing the number of parameters in the model. One way is to consider matrices that have some structure, therefore have fewer parameters. Another way is to have part-of-speech-dependent word representations, where words with certain part-of-speech tags will have many fewer parameters; for example, matrices for adjectives and vectors for nouns (similar to Baroni and Zamparelli (2010)), etc.

Another possibility is to investigate different ways of initializing the matrix-space model. It might be possible to use existing sentiment lexica to develop better ways to initialize word matrices. Perhaps, word clustering techniques (e.g., Brown et al. (1992)) could be used to initialize words such that words appearing in similar contexts initialized with similar matrices.

The other possible research direction is to employ a "curriculum learning" strategy (Bengio et al. (2009)), to learn our proposed matrix-space model, by learning from shorter phrases at first, and gradually moving to longer phrases.

Finally, our proposed model could be used for other tasks in lexical semantics such as paraphrase detection, if we consider the matrix-space semantic representation of a phrase, rather than its sentiment value. Then, it might be possible to formulate a learning objective that enforces similarities between the matrix representations of similar phrases.

## BIBLIOGRAPHY

[Bansal et al.2008] Mohit Bansal, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *International Conference on Computational Linguistics (COLING)*.

[Baroni and Zamparelli2010] Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Morristown, NJ, USA. Association for Computational Linguistics.

[Bautin et al.2008] Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Bengio et al.2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09. ACM.

[Bethard et al.2004] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*.

[Blair-Goldensohn et al.2008] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. Reis, and J. Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era (NLPIX)*.

[Blitzer et al.2007] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June. Association for Computational Linguistics.

[Breck et al.2007] Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India.

[Brown et al.1992] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, and J.C.

Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

[Byrd et al.1995] R. H. Byrd, P. Lu, and J. Nocedal. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, pages 1190–1208.

[Cardie et al.2003] Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane Litman. 2003. Combining low-level and summary representations of opinions for multi-perspective question answering. In *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*, pages 20–27.

[Chang et al.2010] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[Chesley et al.2006] Paula Chesley, Bruce Vincent, Li Xu, and Rohini Srihari. 2006. Using verbs and adjectives to automatically classify blog sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 27–29.

[Choi and Cardie2008] Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Choi et al.2005] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

[Choi et al.2006] Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Morristown, NJ, USA. Association for Computational Linguistics.

[Clarke et al.2010] James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *ACL Conference on Natural Language Learning (CoNLL)*.

[Councill et al.2010] Isaac G. Councill, Ryan McDonald, and Leonid Velikovich. 2010. What's great and what's not: learning to classify the scope of negation

for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Crammer and Singer2001] Koby Crammer and Yoram Singer. 2001. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press.

[Crammer and Singer2003] K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991.

[Das and Chen2001] Sanjiv Das and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*.

[Das et al.2005] Sanjiv Ranjan Das, Peter Tufano, and Francisco de Asis Martinez-Jerez. 2005. eInformation: A clinical study of investor discussion and sentiment. *Financial Management*, 34(3):103–137.

[Dave et al.2003] K. Dave, S. Lawrence, and D.M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

[de Marneffe et al.2010] Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2010. Was it good? It was provocative. learning the meaning of scalar adjectives. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, July 11–16. ACL.

[Devitt and Ahmad2007] Ann Devitt and Khurshid Ahmad. 2007. Sentiment analysis in financial news: A cohesion-based approach. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 984–991.

[Dhillon2001] I. S. Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*.

[Donahue and Grauman2011] Donahue and Grauman. 2011. Annotator rationales for visual recognition.

[Dowty et al.1981] D. Dowty, R. Wolf, and S. Peters. 1981. Introduction to montague semantics.

[Esuli and Sebastiani2006] Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

[Fellbaum1998] Christiane Fellbaum. 1998. WordNet An Electronic Lexical Database.

[Felzenszwalb et al.2008] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. 2008. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Finley and Joachims2008] Thomas Finley and Thorsten Joachims. 2008. Training structural svms when exact inference is intractable. In *International Conference on Machine Learning (ICML)*.

[Frege1892] Frege. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100.

[Glorot et al.2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classidication: A deep learning approach. In *International Conference on Machine Learning (ICML)*.

[Godbole et al.2007] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Goldberg and Zhu2006] Andrew B. Goldberg and Jerry Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*.

[Hardin and Hilbe2007] James W. Hardin and Joseph Hilbe. 2007. *Generalized Linear Models and Extensions*. Stata Press.

[Hatzivassiloglou and McKeown1997] Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *EACL*, pages 174–181.

[Hu and Liu2004a] Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177.

[Hu and Liu2004b] Minqing Hu and Bing Liu. 2004b. Mining opinion features in customer reviews. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 755–760.

[Jiang et al.2011] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 151–160.

[Joachims et al.2009] Thorsten Joachims, Thomas Finley, and Chun-Nam Yu. 2009. Cutting plane training of structural svms. *Machine Learning*, 77(1):27–59.

[Joachims1997] T. Joachims. 1997. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, Universität Dortmund, LS VIII-Report.

[Joachims1999] Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. pages 169–184.

[Johansson and Moschitti2011] Richard Johansson and Alessandro Moschitti. 2011. Extracting opinion expressions and their polarities – exploration of pipelines and joint models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 101–106, Portland, United States.

[Kale et al.2007] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Tim Finin, and Anupam Joshi. 2007. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*. Short paper.

[Kennedy and Inkpen2006] Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2, Special Issue on Sentiment Analysis)):110–125.

[Kim and Hovy2005] Soo-Min Kim and Eduard Hovy. 2005. Identifying opinion holders for question answering in opinion texts. In *Proceedings of the AAAI Workshop on Question Answering in Restricted Domains*.

[Kumar et al.2010] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23*. NIPS.

[Lee and Seung2001] D. Lee and H. Seung. 2001. Algorithms for non-negative matrix factorization. In *NIPS*.

[Liu and Seneff2009] Jingjing Liu and Stephanie Seneff. 2009. Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 161–169, Singapore, August. Association for Computational Linguistics.

[Liu2012] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*.

[Mao and Lebanon2006] Yi Mao and Guy Lebanon. 2006. Isotonic conditional random fields and local sentiment flow. In *Neural Information Processing Systems (NIPS)*.

[McDonald et al.2007] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Mei et al.2007] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of WWW*, pages 171–180, New York, NY, USA. ACM Press.

[Mitchell and Lapata2010] Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

[Mohammad et al.2009] Saif Mohammad, Cody Dunne, and Bonnie Dorr. 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*, pages 599–608, Singapore, August. Association for Computational Linguistics.

[Moilanen and Pulman2007] Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, pages 378–382, September 27-29.

[Montague1974] Montague. 1974. Formal philosophy: Selected papers of richard montague.

[Morinaga et al.2002] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 341–349. Industry track.

[Murray and Carenini2009] G. Murray and G. Carenini. 2009. Predicting subjectivity in multimodal conversations. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1348–1357. Association for Computational Linguistics.

[Murray and Carenini2011] G. Murray and G. Carenini. 2011. Subjectivity detection in spoken and written conversations. *Natural Language Engineering*, 17(03):397–418.

[Nakagawa et al.2010] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[Niu et al.2005] Yun Niu, Xiaodan Zhu, Jianhua Li, and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *Proceedings of the American Medical Informatics Association 2005 Annual Symposium*.

[O'Connor et al.2010] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[Ott et al.2011] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Pang and Lee2004] Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271, Morristown, NJ, USA. Association for Computational Linguistics.

[Pang and Lee2005] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Pang and Lee2008] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

[Pang et al.2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Petersen and Pedersen2008] K. B. Petersen and M. S. Pedersen. "2008". *The Matrix Cookbook*. "Technical University of Denmark", "oct". "Version 20081110".

[Petrov and Klein2007] Slav Petrov and Dan Klein. 2007. Discriminative log-linear grammars with latent variables. In *Neural Information Processing Systems (NIPS)*.

[Polanyi and Zaenen2004] Livia Polanyi and Annie Zaenen. 2004. Contextual lexical valence shifters. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.

[Rao and Ravichandran2009] Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 675–682, Athens, Greece, March. Association for Computational Linguistics.

[Rudolph and Giesbrecht2010] Sebastian Rudolph and Eugenie Giesbrecht. 2010. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 907–916, Morristown, NJ, USA. Association for Computational Linguistics.

[Schapire et al.2002] Robert E. Schapire, Marie Rochery, Mazin G. Rahim, and Narendra Gupta. 2002. Incorporating prior knowledge into boosting. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 538–545, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Shaikh et al.2007] Mostafa Shaikh, Helmut Prendinger, and Ishizuka Mitsuru. 2007. Assessing sentiment of text by semantic dependency and contextual

valence analysis. *Affective Computing and Intelligent Interaction*, pages 191–202.

[Snyder and Barzilay2007] Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*, pages 300–307.

[Somasundaran et al.2007] Swapna Somasundaran, Theresa Wilson, Janyce Wiebe, and Veselin Stoyanov. 2007. QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Stone et al.1966] P.J. Stone, D.C. Dunphy, and M.S. Smith. 1966. The general inquirer: A computer approach to content analysis.

[Stoyanov and Cardie2008] Veselin Stoyanov and Claire Cardie. 2008. Annotating topics of opinions. In *International Conference on Language Resources and Evaluation (LREC)*.

[Stoyanov and Cardie2011] Veselin Stoyanov and Claire Cardie. 2011. Automatically creating general-purpose opinion summaries from text. In *Recent Advances in Natural Language Processing (RANLP)*.

[Stoyanov et al.2005] Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the OpQA corpus. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 923–930, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

[Stoyanov2009] Veselin Stoyanov. 2009. *Opinion Summarization: Automatically Creating Useful Representations of the Opinions Expressed in Text*. Ph.D. thesis, Cornell University.

[Taboada et al.2009] Maite Taboada, Julian Brooke, and Manfred Stede. 2009. Genre-based paragraph classification for sentiment analysis. In *Proceedings of the SIGDIAL 2009 Conference*, pages 62–70, London, UK, September. Association for Computational Linguistics.

[Taboada et al.2011] Maite Taboada, Julian Brooke, Milan Tofiloskiy, and Kim-

berly Vollz. 2011. Lexicon-based methods for sentiment analysis. In *Computational Linguistics*.

[Takamura et al.2005] Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientation of words using spin model. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 133–140.

[Tateishi et al.2001] Kenji Tateishi, Yoshihide Ishiguro, and Toshikazu Fukushima. 2001. Opinion information retrieval from the Internet. *Information Processing Society of Japan (IPSJ) SIG Notes*, 2001(69(20010716)):75–82. Also cited as "A reputation search engine that gathers people's opinions from the Internet", IPSJ Technical Report NL-14411. In Japanese.

[Thomas et al.2006] Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Titov and McDonald2008] Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 308–316, Columbus, Ohio.

[Tong2001] Richard M. Tong. 2001. An operational system for detecting and tracking opinions in on-line discussion. In *Proceedings of the Workshop on Operational Text Classification (OTC)*.

[Tsochantaridis et al.2004] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, pages 104–112.

[Turney and Littman2003] Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

[Turney2002] Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA. Association for Computational Linguistics.

[Velikovich et al.2010] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexi-

cons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, California, June. Association for Computational Linguistics.

[Wang and Liu2011] Dong Wang and Yang Liu. 2011. A pilot study of opinion summarization in conversations. In *ACL*.

[Wiebe et al.2003] Janyce Wiebe, Eric Breck, Christopher Buckley, Claire Cardie, Paul Davis, Bruce Fraser, Diane Litman, David Pierce, Ellen Riloff, Theresa Wilson, David Day, and Mark Maybury. 2003. Recognizing and organizing opinions expressed in the world press. In *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*.

[Wiebe et al.2005] Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

[Wiebe2000] Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *AAAI*, pages 735–740.

[Wilson et al.2004] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? In *AAAI*.

[Wilson et al.2005a] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005a. Opinionfinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35, Morristown, NJ, USA. Association for Computational Linguistics.

[Wilson et al.2005b] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Wu and Srihari2004] Xiaoyun Wu and Rohini Srihari. 2004. Incorporating prior knowledge with weighted margin support vector machines. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333, New York, NY, USA. ACM.

[Yessenalina and Cardie2011] Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the*

*Conference on Empirical Methods in Natural Language Processing*, pages 172–182. Association for Computational Linguistics.

[Yessenalina et al.2010a] Ainur Yessenalina, Yejin Choi, and Claire Cardie. 2010a. Automatically generating annotator rationales to improve sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Yessenalina et al.2010b] Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010b. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1046–1056. Association for Computational Linguistics.

[Yu and Joachims2009] Chun-Nam Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*.

[Yuille and Rangarajan2003] Alan L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Computation*, 15(4):915–936, April.

[Zaidan and Eisner2008] Omar F. Zaidan and Jason Eisner. 2008. Modeling annotators: a generative approach to learning from annotator rationales. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Zaidan et al.2007] Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.