Saarland University

Department of Computational Linguistics

Master Thesis:

Fine-grained Sentiment Analysis with Discourse Structure

Author:
Yudong Zhou
Matriculation: 2546248

Supervisors:
Prof. Manfred Pinkal
Dr. Alexis M. Palmer
Annemarie Friedrich

14th October 2013

Abstract

Sentiment analysis refers to the task of natural language processing to determine whether a piece of text contains some subjective information and what subjective information it expresses, i.e., whether the attitude behind this text is positive, negative or neutral. Understanding the opinions behind user-generated content automatically is of great help for commercial and political use, among others. The task can be conducted on different levels, classifying the polarity of words, sentences or entire documents.

In this thesis, we propose and investigate a method of elementary discourse unit (EDU) level sentiment analysis using discourse features. Following prior work, we hypothesize that when we want to predict the sentiment of a certain EDU, we can use the sentiments of other EDUs which stand in some discourse relation with the current one. For example, a Contrast relation is likely to signal that the sentiments of two arguments of this relation are different. Once we know one of the sentiments, we can try to use this information to predict the other one. Some discourse theories have been applied in this context, but no prior work compares relative influence of different discourse theories on the performance of sentiment analysis. To the best of our knowledge, this is the first work comparing different discourse theories in a principled way in the task of sentiment analysis, making use of state-of-the-art discourse parsers.

To discover the discourse relations we employ two discourse parsers based on two different discourse theories, respectively Rhetorical Structure Theory (RST) and the The Penn Discourse Treebank (PDTB). We propose several models to represent the discourse structures and test them on the task of EDU-level sentiment prediction. In the discourse representations the relations are determined by the discourse parsers and sentiment values of connected EDUs are taken from our gold-standard data. We also implemented two baselines using lexical features and features concerning adjacent EDUs. The classification accuracy increases if discourse features are added, and RST-based discourse features outperforms PDTB-based discourse features.

Contents

List of Figures

List of Tables

1 Introduction

Sentiment analysis[1] has become an attractive research area in the field of natural language processing, with the aim of automatically determining the attitude or sentiment expressed by a text. With a rapid growth of user generated content on the Web, especially in online review sites, it is found an interesting and useful task to find out what other people think. In a survey conducted by [Horrigan (2008)] on over 2000 American adults, it is shown that among readers of online reviews of restaurants, hotels, and various services (e.g., travel agencies or doctors), between 73% and 87% report that reviews had a significant influence on their purchase.

In this thesis we propose and investigate a method of elementary discourse unit (EDU) level sentiment analysis using discourse features. An EDU is a short piece of text with a complete and continuous opinion. (See Chapter 2 for more details). Following prior work, we hypothesize that when we want to predict the polarity of a particular EDU, we can use the polarities of other EDUs which stand in some discourse relation with the current one. To discover the discourse relations, we employ two discourse parsers based on two different theories of discourse, Rhetorical Structure Theory (RST) and the lexically-based approach of the Penn Discourse Treebank (PDTB). We propose several ways to extract feature representations from the discourse structures and test them on the task of EDU-level sentiment prediction. We also implement two baselines using lexical features and features concerning adjacent EDUs. The details of the four sets of features (Lexicon-based, Adjacency-based, RST-based and PDTB-based) are discussed in Chapter 4.

The goal of sentiment analysis is to determine the sentiment or polarity[2] of a piece of text. Sentiment analysis can be done at different granularities, depending on the length of text. For example, document level sentiment analysis assigns a polarity label for the whole document, while polarities borne by shorter text within one review or article will not be differentiated.

Aspect-based[3] sentiment analysis goes one step beyond general sentiment analysis. An aspect is an abstract set of related things that people can judge. For example, service is a commonly used aspect in restaurant reviews and the contents under this aspect could be the attitude of the waiters, waiting time, etc. For an aspect-based approach, a fine-grained analysis is made in order to identify sentiment orientations at the level of aspects or features. In such studies, a procedure of aspect extraction is conducted before

[1] The task is also called *Opinion Mining* in some literatures.

[2] While the term *polarity* is associated with several linguistic phenomena, it will be used to refer to sentiment polarity in this thesis.

[3] It is also referred as *feature-based* sometimes, and an *aspect* is called a *topic*. *Aspect* will be used in this thesis.

sentiment polarity prediction, and sentiments of each aspect are summarized afterwards. This task meets a practical demand for some online review sites such that, on a website that sells cellphones, reviews with distinctions between aspects like sound, battery, screen or camera would be useful.

There are some more fine-grained levles of sentiment analysis, such as sentence-level, phrase-level and word-level sentiment analysis. Our focus in this thesis is sentiment analysis on the elementary discourse unit (EDU) level. An EDU is a minimal block of a discourse analysis which is meaningful and continuous expression. It could be a sentence, a clause or a phrase. We think that the polarity within one EDU trends to be continuous, which makes EDU-level analysis important to understand the content precisely. If we regard an article or a paragraph consisting of minimal units that bear a uniform opinion, mining these uniform opinions would be an essential and important task for understanding the whole article or paragraph. Consider the following example from TripAdvisor[4]:

Example 1.1. *Different opinions within one sentence*

We had a great dining experience at Rave to celebrate my son 's exam success, ***but unfortunately the second time of visiting was not so good,*** *when we just wanted a drink before going out for a meal.*

This is one sentence consisting of three EDUs (distinguished by normal, **bond** and *italic* font) with different topics and sentiments: the first EDU talks about a previous experience and the polarity is positive; the second EDU is about the current experience and is negative; the last one describes the time and is neutral. If we consider the overall sentiment of the whole text span, assigning only one sentiment label will lose certain mentioned information, resulting in an incomplete understanding of the text.

EDUs are linked together in an article via discourse relations. We hypothesize that some discourse relations will indicate a shift in either aspect, polarity, or both aspect and polarity. Motivated by this, in this thesis, we use some discourse features to predict sentiments of EDUs, aiming to understand sentiment-bearing text more specifically.

Discourse information could tell the shifts of aspects or/and polarity to some degree, and might be an interesting feature to improve polarity prediction. Here are two examples of aspect or/and polarity changes between two EDUs connected by a contrast relation, taken from TripAdvisor.

Example 1.2. *Shift of Polarity and/or Aspect*

(a) *[Although the appearance of the hotel front pales in comparison with the other 4 neighbouring hotel,]*$_{\mathrm{EDU1}}$ *[but the room was surprising roomy by NYC stds , clean and well-equipped.]*$_{\mathrm{EDU2}}$

[4] http://www.tripadvisor.com/

(b) [At some points there were large queues at check-in which we saw,]$_{\text{EDU1}}$ *[but what can you expect with a hotel with 1700 rooms!]*$_{\text{EDU2}}$

The contrast relation in (a) co-occurs with both an aspect change (*location* → *rooms*) and a polarity change (*neg* → *pos*), while in (b) there is only a polarity shift (*neg* → *neu*).

Different discourse relations may also have different influences on polarity and/or aspect shift. As our primary goal is to predict polarity of an EDU [5], we will use data with aspect annotations to make it simple and clear. Our main focus is to measure different dis-course relations' influence on polarity classification.

This mentioned method could be useful when there isn't enough lexical information to classify the sentiment, or when the words' sentiments within one EDU is contradictory. The latter scenario is likely due to the fact that natural language expression could be ambiguous, while the sentiment expressed in one EDU is definable. It is also a proper method to understand some sentences or some parts of sentences by understanding context.

We take text files from two datasets, use two different style discourse parsers to parse these files into EDUs and extract discourse relations among EDUs. These relations, together with adjacent relations and polarity scores from additional lexical resources, are used to build our model to represent these EDUs. Then we use cross-validation method to train and test our model. Figure 1 illustrates the work flow.

The contributions of this thesis include: apply rich discourse features to EDU level sentiment prediction with consideration of aspects; two state-of-the-art discourse parsers based on different theories are tested and compared. To the best of our knowledge, this work is the first comparing different discourse theories in a principled way in the task of sentiment analysis, making use of state-of-the-art discourse parsers. The classification accuracy of the EDU level sentiment analysis increases if discourse features are added, and out RST-based discourse features outperforms PDTB-based discourse features.

The contents of this thesis are organized as follows,

Chapter 2 introduces the background of related research work, including different discourse modelling theories and previous approaches to sentiment analysis.

Chapter 3 introduces additional resources employed in this thesis, including two datasets and three lexical resources.

Chapter 4 explains features we use to classify sentiment, and how we extract these features from additional lexical resources and parsers' outputs.

[5] We refer the EDU we want to predict to *the current EDU* in the rest of this thesis

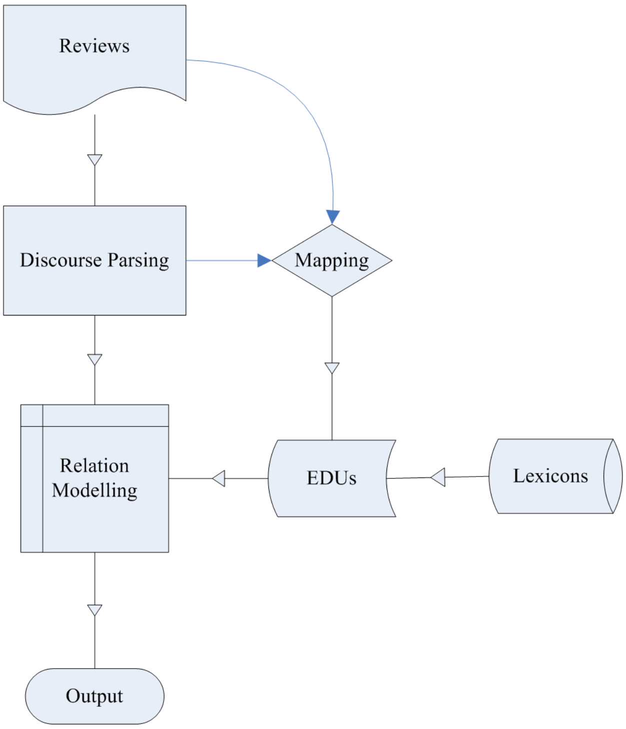


Figure 1. Work flow

Chapter 5 presents the implementation of our methods and evaluation of the results, followed by comparisons and analysis on these results.

Chapter 6 concludes with a brief summary of this thesis and possible future work.

2 Background

This chapter introduces the background of theories and research work relevant to this thesis. They are divided into three sections: the first section is about two discourse theories, the second section introduces two state-of-the-art discourse parsers corresponding to mentioned two theories, the third section introduces various research topics and approaches in the field of sentiment analysis.

2.1 Theories of Discourse Structure

Discourse analysis is a kind of text analysis beyond the sentence level. Generally a discourse consists of a sequence of sentences, although structure within one sentence also matters sometimes. These sentences, or shorter text spans (for example, sub-sentences), are the basic components of discourse. Discourse structure, as stated in Webber et al. (2012), are the patterns that one sees in multi-sentence (multi-clausal) texts. Analysing this structure is important to understand information in the text.

There are several theories proposed to model discourse structure. Two of them are involved in this thesis, both of which try to use links known as discourse relations to connect different sentences/clauses/EDUs within the same text. Rhetorical Structure Theory (RST) [Mann and Thompson (1988)] was originally developed as part of studies of computer-based text generation, by a team at Information Sciences Institute (part of University of Southern California). It has been commonly used in numerous Natural Language Processing (NLP) tasks, such as Information Extraction, Auto Summarization, etc. The Penn Discourse Treebank (PDTB) [Prasad et al. (2008)] is a corpus annotated with information related to discourse structure by Institute for Research in Cognitive Science, University of Pennsylvania.

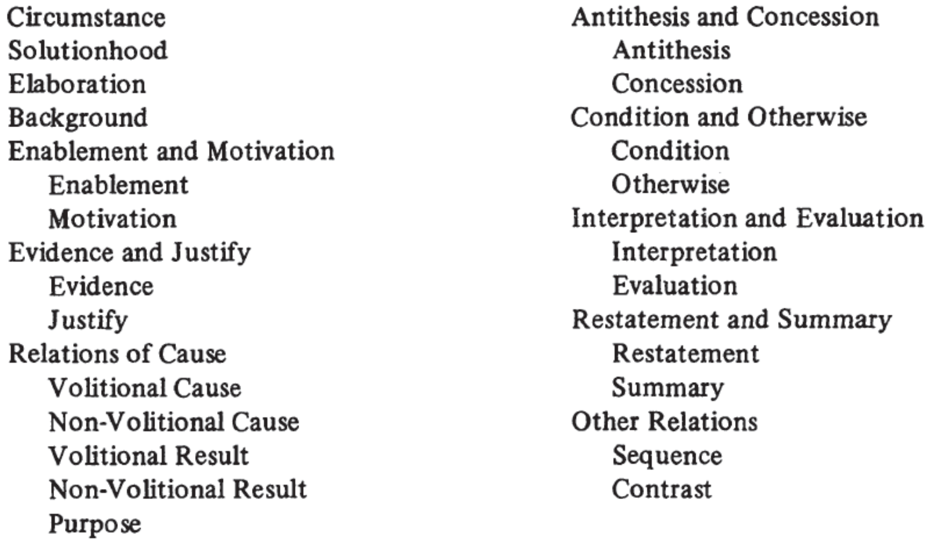
2.1.1 Rhetorical Structure Theory (RST)

Rhetorical Structure Theory (RST) is a theory describing discourse relations among discourse segments. As introduced in Chapter 1, an EDU is a minimal block of a discourse analysis which is a meaningful, continuous expression and has independent functional integrity. EDUs are organised by some rules to constitute a document. These rules, or regularities explain how different text spans forms a document. RST is a theory to explain the coherence of texts: each EDU has an evident role for supporting the whole document, these roles are organised through discourse relations.

In RST, two non-overlapping spans of text are connected by a discourse relation. The essential span is called the *nucleus* and the supporting one is called the *satellite*. For instance, in a *Background* relation, one text span is the focus of writer's opinion and a second text span provides additional information for understanding this opinion. Text whose understanding is being facilitated is recognized as *Nucleus* and text for facilitating understanding is recognized as *Satellite.* As described in [Mann and Thompson (1988)], the definition of a discourse relation has four parts:

1. *Constraints on the Nucleus,*
2. *Constraints on the Satellite,*
3. *Constraints on the combination of Nucleus and Satellite,*
4. *The Effect.*

For example, in the definition of the EVIDENCE relation, *constraints on N* include "Reader might not believe N to a degree satisfactory to Writer"; *constraints on S* require "The reader believes S or will find it credible"; *constraints on the N + S combination* says "R's comprehending S increases R's belief of N"; and *the effect* is "R's belief of N is increased". These definitions of relations are the principles for identifying the relations together with the spans in text. A list of relations defined in RST con be found in Figure 2, including 12 groups of relations and possible sub-classes.



- Circumstance
- Solutionhood
- Elaboration
- Background
- Enablement and Motivation
  - Enablement
  - Motivation
- Evidence and Justify
  - Evidence
  - Justify
- Relations of Cause
  - Volitional Cause
  - Non-Volitional Cause
  - Volitional Result
  - Non-Volitional Result
  - Purpose
- Antithesis and Concession
  - Antithesis
  - Concession
- Condition and Otherwise
  - Condition
  - Otherwise
- Interpretation and Evaluation
  - Interpretation
  - Evaluation
- Restatement and Summary
  - Restatement
  - Summary
- Other Relations
  - Sequence
  - Contrast

Figure 2. RST relation types

In most cases a relation links two text spans (EDUs), usually adjacent but not necessarily so. As introduced, each EDU is categorised as either nucleus or satellite. If the relation doesn't have a particular preference, it could be multinuclear also where two text spans are treated equally. For example, Contrast is a multinuclear relation.

Possible RST structures are defined by schemas which are patterns consisting of some text spans, a specification of relations between them, and how nuclei are related to the whole collection. Five kinds of schemas recognized by RST are illustrated in Figure 3. Horizontal lines stand for text spans, the curves indicate relations with the arrow pointing to nuclei and the straight lines also identify nuclei. For example in the top-left relation in Figure 3, there are two EDUs (horizontal lines); the straight line on the second EDU indicates that this EDU is a nucleus and the arrow indicates a circumstance relation, pointing from the satellite to the nucleus. There is one kind of schema that is not mentioned here, which is a single relation with nucleus and satellite.

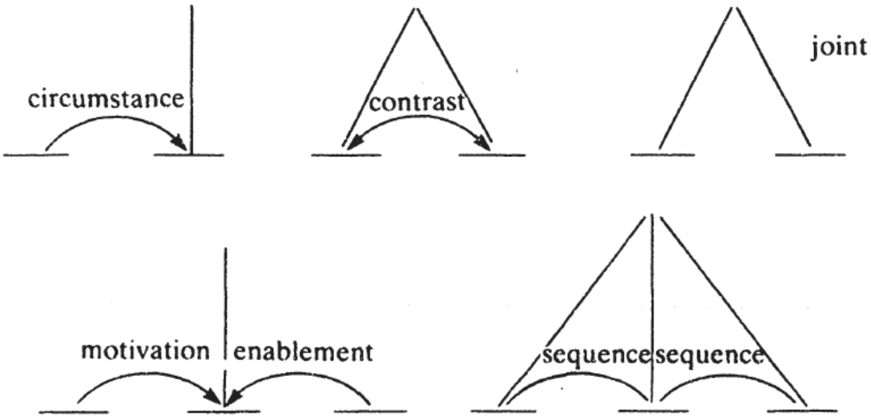


Figure 3. RST schemas

A global RST tree structure is constructed for each document/text. A typical RST tree is constructed as following: leaves of the tree are EDUs; adjacent leaves (usually two, though exception can be found) can be connected by a RST discourse relation with nuclei and satellite distinguished and this relation is the node that connects these two EDUs; a node may connect to another node or a leaf; nodes keep joining until one root node is constructed. Figure 4[6] illustrates how RST-style parsing result is organized and represented.

2.1.2 The Penn Discourse Treebank (PDTB)

The Penn Discourse Treebank (PDTB) is a corpus annotated with information related to discourse structure and discourse semantics, by Institute for Research in Cognitive Science, University of Pennsylvania. The annotation was done on the Wall Street Journal (WSJ) Corpus which contains more than one million words. The PDTB focuses on encoding discourse relations. The PDTB follows a lexically-grounded approach to discovering explicit discourse relations. *Discourse connectives* are the necessary lexical items that

[6] From INTRO TO RST, http://www.sfu.ca/rst/01intro/intro.html

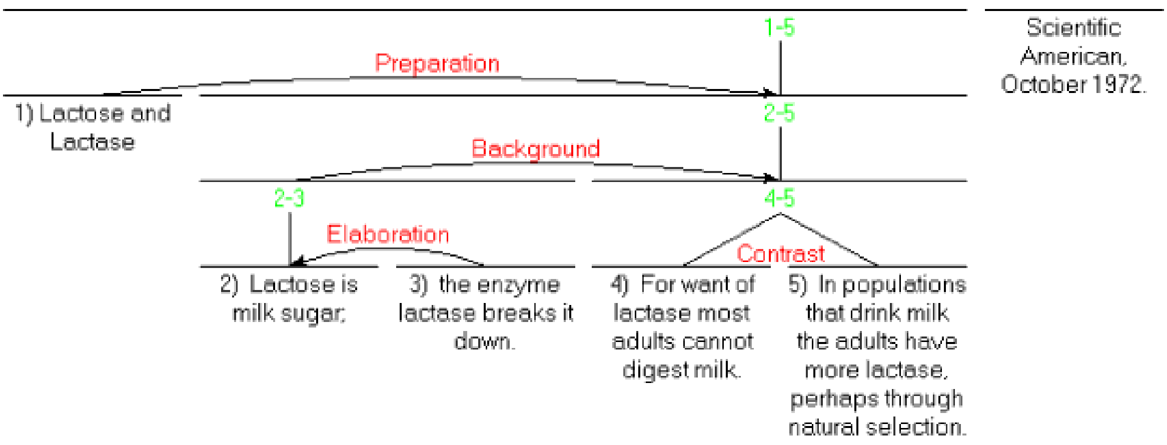

1-5
Scientific American, October 1972.
Preparation
1) Lactose and Lactase
2-5
Background
2-3
4-5
Elaboration
Contrast
2) Lactose is milk sugar;
3) the enzyme lactase breaks it down.
4) For want of lactase most adults cannot digest milk.
5) In populations that drink milk the adults have more lactase, perhaps through natural selection.

Figure 4. Rhetorical Structure Theory (RST) Tree

imply or help identify the discourse relations. Consider the following example from the Penn Discourse Treebank:

Example 2.1. Because *mutual fund trades dont take effect until the market closed,* these shareholders effectively stayed put.

For this example, the Cause relation can be annotated by marking the discourse connective *Because*. Whether a word is a connective or what sense of relation a connective could indicate depends on the content of the sentence.

PTDB takes a binary predicate-argument view of discourse relations. A discourse connective is regarded as a predicate that takes two spans of text as its arguments, namely Arg1 and Arg2. Compared to RST, PDTB annotates local relations instead of tree-level long-distance relations. There is no commitment that a high-level structure for a document can be built based on PDTB annotations of relations and arguments. The PDTB provides a three level hierarchical schema for relations, as shown in Figure 5. In PDTB there are 14 relation tags, constituting 4 parent classes: Temporal, Comparison, Contingency, Expansion.

Discourse relations realized by discourse connectives that are drawn from syntactically well-defined classes are called *Explicit* relations and these connectives are also called Explicit connectives. There is no constraint for the arguments of explicit connectives: they could be anywhere in the text. Consider the following example from Penn Discourse Treebank. Arg1 is in italics, and Arg2 is in bold. We can observe that Arg1 can appear embedded in Arg2.

Example 2.2. As an indicator of the tight grain supply situation in the U.S., market analysts said that **late Tuesday the Chinese government**, *which often buys U.S. grains in quantity,* **turned** <u>instead</u> **to Britain to buy 500,000 metric tons of wheat.**

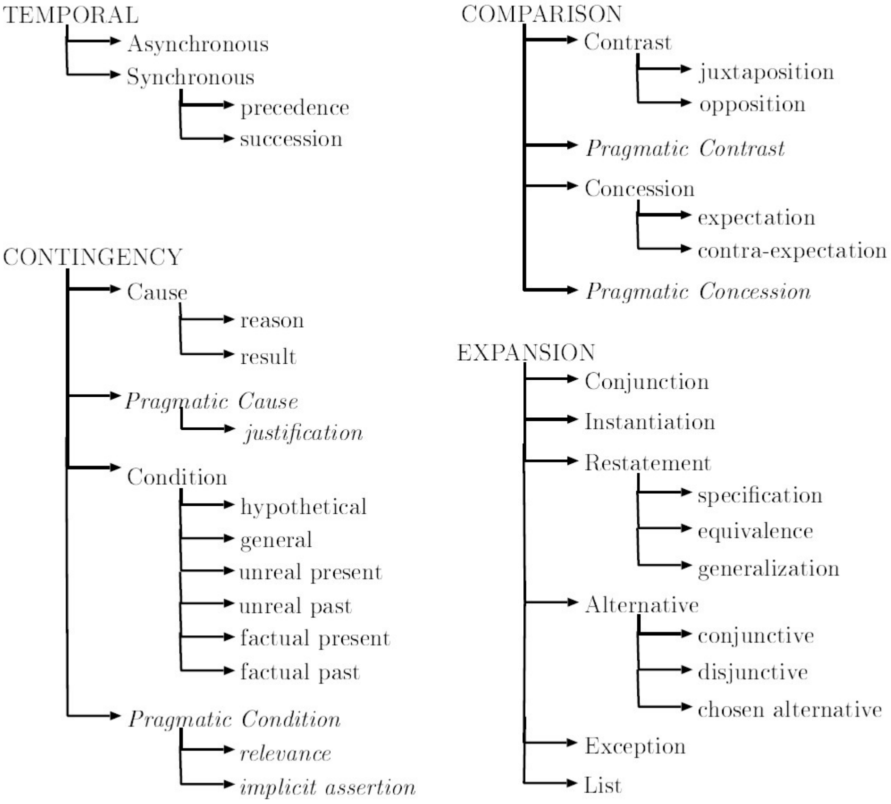


Figure 5. PDTB relation types

Explicit relations are distinguished from *Implicit* relations which hold between two adjacent sentences (or EDUs) in the absence of explicit connectives. There might be a implicit discourse connective connecting two text spans or none. Consider the following example of an implicit Restatement relation without any connectives connecting two arguments. Arg1 is *italicized* and Arg2 is **bolded**.

Example 2.3. *My wife and I stayed at the Empire Hotel for 8 nights on our honeymoon.* **We booked a superior hotel and even though it was supposed to be for honeymooners they placed us on the 3rd floor.**

For both explicit and implicit relations, there exist two and only two arguments. For explicit relations, Arg2 is the argument to which the connective is syntactically bound and Arg1 is the other argument. As for implicit relations, Arg1 and Arg2 are labelled by their linear order in the text.

2.2 Discourse Parsers

In this section we introduce two discourse parsers used in this thesis and how the parsers work on our datasets. There are several discourse parsers publicly available, such as

HILDA discourse parser [Hernault et al. (2010)], The Feng-Hirst parser [Feng and Hirst (2012)], the Lin parser [Lin et al. (2010)]. The first two parsers are based on RST theory and the last one is an end-to-end PDTB-style parser. We choose the Feng-Hirst parser and the Lin parser for our discourse parsing task for the following reasons:

1. They represent two most popular discourse analysis theories or styles: RST and PDTB, which allows us to make comparisons on the performance of our sentiment analysis task between these two discourse theories.
2. HILDA parser is the first fully-implemented feature-based RST-style discourse parser that works at the full text level. The Feng-Hirst parser takes HILDA as its basis and reports an improvement over HILDA parser on several features. According to this the Feng-Hirst parser can be regarded as a state-of-the-art RST parser.

The Feng-Hirst parser is a RST-style text-level discourse parser developed by Wei Feng and Graeme Hirst in Department of Computer Science, University of Toronto. It incorporates rich-linguistic features such as semantic similarity and cue phrases. The parser is trained and tested on the RST Discourse Treebank (RST-DT). RST-DT is a RST-style annotated corpus consisting of 385 documents from the *Wall Street Journal.*

The Feng-Hirst parser takes a text T as input, outputs a segmentation S of EDUs of T (User specified segmentation is accepted) and a discourse parse tree based on T and S. The Feng-Hirst parser was trained and tested on the RST Discourse Treebank (RST-DT) [Carlson et al. (2003)]. There are two steps in the work flow: EDU segmentation and Tree building. Previous parser HILDA has achieved a F-score of 93.8% on the EDU segmentation task and The Feng-Hirst parser took the EDU segmentation results from HILDA. For the tree building task the Feng-Hirst parser reported a 95.64% accuracy and 89.51% F-score, both significantly outperform HILDA parser.

The Lin parser was developed by Ziheng Lin, Hwee Tou Ng and Min-Yen Kan from Department of Computer Science, National University of Singapore. It is the first full end-to-end discourse parser in the PDTB style. It takes a text T as input and outputs a discourse structure of T. The components of this parser consist of the connective classifier, the argument labeller, the explicit classifier, the non-explicit classifier, and the attribution span labeller. Figure 6 shows the work flow of this parser. The relations the Lin parser tries to discover are the Level 2 relations in the PDTB discourse relation hierarchy (Recall Figure 5 for details).

The training and testing of the Lin parser was based on PDTB, using Sec. 0221 for training, Sec. 22 for development, and Sec. 23 for testing as suggested in [Prasad et al. (2007)]. They report an F score of 86.77% over the baseline (uses only the connectives as features, obtain an F score of 86.00%) for their explicit classifier; an F score of 39.63%

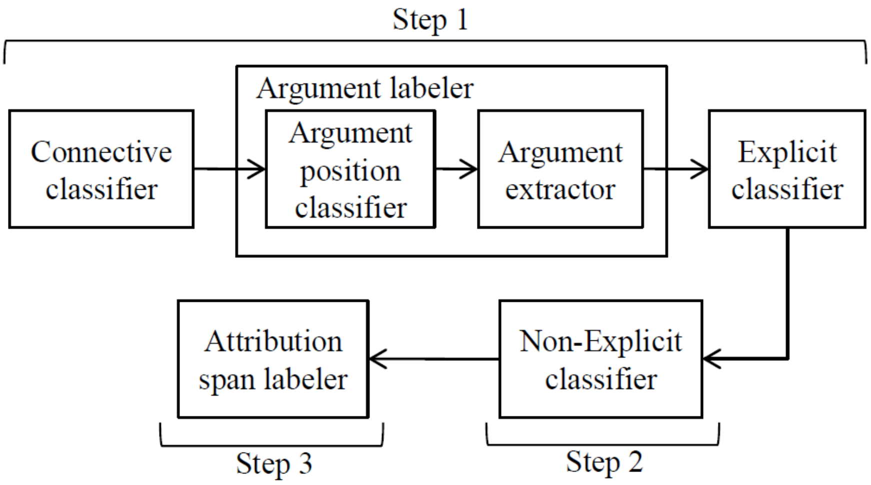


Figure 6. The work flow of the Lin parser

for the implicit classifier. Given the fact that the results of implicit relation identification isn't so sound and might introduce noise to our later sentiment classification task, we take only parsing results of explicit relations and their arguments in our work.

2.3 Sentiment Analysis

Work which deals with computational models of opinion, sentiment, and subjectivity in texts (most common), speeches and other forms of natural language, is known as opinion mining or sentiment analysis. The term "sentiment analysis" is now used to refer to computational analysis which automatically extracts, evaluates and predicts/determines the judgement/attitude in given texts. Early work appears during 2001 such as [Das and Chen (2001)] and [Tong (2001)] whose focus was market or business use. It is still true that work in the sentiment analysis field has potential on marketing applications on the Web. This is also one of the reasons that most work in this field is about online reviews such as product reviews, hotels reviews, restaurant reviews, etc.

In the following sections we first briefly introduce general approaches to sentiment analysis, and then introduce work with consideration of discourse and aspect.

2.3.1 Sentiment Analysis: Approaches

Sentiment analysis could be carried out at different levels based on the length of input text, from word level to review level. Word-level and phrase-level sentiment analysis take advantage of previous work in *Distributional Semantics*, with the objective of determining the polarities of unseen words or phrase. This is out of the range of this thesis, since our major task is to test the influence of discourse structure on EDU-level analysis.

There are two common approaches to sentiment analysis: language model and using additional sentiment annotated dictionaries. Language model is useful to classify whether a span of text is subjective to a certain degree. It is commonly used as a basic feature for classification. (for instance, in [Taboada et al. (2009)] n-gram features are used.); Additional lexical resources are built as knowledge bases with polarity annotated tokens/words/concepts. Given a new text, the algorithm looks up the polarity-bearing words in the dictionary and calculates a sentiment score for the whole text. The classifier then classifies the text with this score.

There are many such lexical resources for English and we use three of them in this work for a wider coverage: AFINN[7] [Nielsen (2011)], A list collected by Minqing Hu and Bing Liu (referred to Hu-Liu)[8] since their work [Hu and Liu (2004)], Lexicon of OpinionFinder system[9] [Wilson et al. (2005)]. These lexical resources are used to build a lexicon-based baseline.

2.3.2 Sentiment Analysis with Discourse Features

In the field of sentiment analysis, much work has been done based on local information of sentences (say, without considering relations or connections between sentences). [Polanyi and Zaenen (2006)] argues that local concentration is incomplete and often gives the wrong results when implemented directly, and polarity calculation is affected by some lexical and discourse context. Discourse information is therefore considered as an important feature for polarity prediction.

Instead of considering each sentence equally, people have tried to measure different degree of contributions of sentences or sub-sentences to the overall sentiment expressed in a document. There are two main approaches to this discourse-sensitive document level sentiment analysis task. One approach is rule-based. [Somasundaran et al. (2009)] employs a constraint-based approach to restrict opinion prediction such that text spans targeted at the same entity with relations in their schema will be constrained with same polarity. [Zhou et al. (2011)] defines a discourse schema with discourse constraints on polarity to discover intra-sentence level discourse relations for eliminating polarity ambiguities. One example of these discourse constraints is that two text spans connected by a Contrast relation should hold opposite polarities.

[7] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[8] http://www.cs.uic.edu/ liub/FBS/opinion-lexicon-English.rar

[9] http://mpqa.cs.pitt.edu/opinionfinder/

Another approach is weight-based. After extracting discourse structures from a document, text spans are assigned different weights for contributing to the polarity score according to their roles in discourse relations. [Taboada et al. (2008)] hypothesizes several weighting schemes, and achieves the best results on “1.5-0.5” weight schema, which gives a weight of 1.5 to words expressed in nuclei and 0.5 to satellites. Later work such as [Wu and Qiu (2012)] assumes that nuclei and satellites of different relations should also be weighted differently, and some relations should have higher weights than others. They train a linear optimizer to find the best weights. Both approaches reported improvements over a lexicon-based baseline that does not take discourse structure into account.

The focus of previous work has been document level polarity prediction using discourse information. To extract sentiment polarity on a more fine-grained level, some work has been done on sentence/sub-sentence level polarity prediction. [Zirn et al. (2011)] use discourse relations for sub-sentence level sentiment prediction. Our intuition is similar to this work, but we consider a broader range of discourse relations while they only distinguish Contrast relations and non-Contrast relations. Moreover, we take into consideration the influence of discourse relations on aspect shift rather than on polarity alone.

Works mentioned above detect discourse relations by looking for explicit discourse connectives, or training classifiers on annotated data. Some works take advantages of state-of-the-art discourse parsers. For example, [Taboada et al. (2008)] and [Heerschop et al. (2011)] use some additional tools like Sentence-level PArsing of DiscoursE (SPADE) [Soricut and Marcu (2003)] to indicate discourse relations. Using discourse parsers may bring in noise, but will increase the coverage of discourse information.

The term discourse structure was used by all works mentioned, while in some work it just refers to the usage of discourse phenomenoa, in some other works RST-style discourse structure is referred to. The discourse parsers we use in this work implement not only RST-style structure, but also PDTB-style structure, and hence we uniformly use the term discourse structure for these two specific theories.

2.3.3 Aspect-based Sentiment Analysis

We take aspect information into consideration in our work for the following reasons. First, in order to predict the “relevant” polarity expressed by an EDU, we shall know the topic or aspect of review this EDU is about. By only focusing on content relevant to the current aspect, the prediction result we get should be more realistic, since there is some content that contains polarity-bearing words that are not relevant to the concerned aspect. (Details can be found in Chapter 4) Second, from Example 1.2, we can see that discourse structure could indicate not only information about polarity continuity, but also some other information, one of which is aspect information. We add features about aspect in our work and could test whether aspect information has an influence on polarity continuity through the whole text.

There are several works concerning aspect-based sentiment analysis. [Wang et al. (2010)] and [Brody and Elhadad (2010)] use the Latent Dirichlet Allocation (LDA) topic modelling method to gather content about same aspect and use some state-of-art features to calculate the polarity of that aspect and each aspects influence on the overall sentiment rating. They also calculate the influence of different aspects on the overall polarity of a review.

In [Lazaridou et al. (2013)] they use an unsupervised method to induce discourse relations and find that discourse relations play the role of opinion and aspect shifter. By using discourse relations as features, they report an improvement on the EDU-level prediction of both sentiment and aspect.

Our implementation of aspect extraction (will be discussed in Chapter 4) is simple since this is not prior task. We consider most aspect information are expressed in nouns and noun phrases. So we implemented a method clustering all mentioned noun phrases for each aspect and select those beyond a threshold as the *representatives* that could reflect one aspect. Aspect shifts from EDU to EDU are tested, to examine whether this information could help our sentiment prediction task.

2.4 Summary

In this chapter we introduced two different theories of discourse structure. We took one state-of-the-art discourse parser for each of the corresponding theories. The Feng-Hirst parser implements RST theory, which constructs a tree structure for each text, while the Lin parser is PDTB style so that it focuses more on local connections. We also discussed different approaches to sentiment analysis, pointing out our major research problem is EDU level sentiment prediction, with simple lexical features and features extracted from discourse structures.

3 Datasets and resources

In this chapter we introduce external resources used in this thesis, including two datasets and three lexical resources. The datasets consist of reviews from different domains such as hostels and products (including cellphones, food and kitchen housewares), with sentiment annotations on the EDU level. We use the data to test our discourse structure models on the task of sentiment analysis. The three lexical resources contain lists of words and their corresponding sentiments. They are used to calculate the lexical baselines in the sentiment prediction task.

3.1 Datasets

We extract two annotated datasets from [Lazaridou et al. (2013)] and [Zirn et al. (2011)]. The former one consists of annotations of both sentiment and aspect for each EDU, while the latter one contains EDU level sentiment annotations only. All annotations were done by teams of respective authors.

3.1.1 TripAdvisor Dataset

[Lazaridou et al. (2013)] implemented an unsupervised method, so the original dataset consists of an unlabelled part and a labelled part which was used as the gold standard for evaluation. For our purpose in this work, we take only the labelled part, referred to TripAdvisor Dataset later.

This TripAdvisor dataset was retrieved from TripAdvisor.com. It consists of 65 reviews (1541 EDUs, EDUs segmented by SLSEG software package[10]). 9 annotators annotated every EDU with the aspect and sentiment it expresses. Annotators needed to choose at least one aspect (multi aspects in one EDU is allowed) from a candidate aspect list (aspect label *rest* is used when EDUs don't refer to any aspect or refer to a very rare aspect) and one polarity score from (-1,0,1) standing for negative, neutral or positive, respectively. Consider the following examples:

Example 3.1. *TripAdvisor Dataset annotations*

(a) *Booked this hotel based on the reviews and the reasonable pricing .* **value** pos

(b) *Beautiful setting and excellent service .* **location, service** pos

[10] http://www.sfu.ca/ mtaboada/research/SLSeg.html

There are three parts for each item (EDU): the content of the EDU, its aspect(s) and its polarity. Multiple aspects are allowed, for instance in (b) there are two aspects *location* and *service* since both are mentioned in the EDU. There is no multiple polarities case for any EDU in this dataset.

Table 1 and Table 2 show the distribution of aspect and polarity in this dataset. The distribution of polarity is naturally uniform.

Aspect	**Frequency**	**Percentage**
service	246	15.96
value	55	3.57
location	121	7.85
rooms	316	20.51
sleep quality	56	3.63
cleanliness	59	3.83
amenities	180	11.68
food	81	5.26
recommendation	121	7.85
rest	306	19.86
Total	1541	100

Table 1. Aspect Distribution, TripAdvisor

Polarity	**Frequency**	**Percentage**
positive	575	37.46
negative	549	35.77
neutral	411	26.78
Total	1535	100

Table 2. Polarity Distribution, TripAdvisor

[Lazaridou et al. (2013)] used Cohen's kappa score to measure the inter-annotator agreement(IAA): 0.66 for aspect labelling, 0.70 for the sentiment annotation and 0.61 for the joint task of both annotations.

3.1.2 Multi-Domain-Sentiment Dataset

This dataset is rearranged from the dataset used in [Zirn et al. (2011)](referred as the Zirn Dataset or Multi-Domain-Sentiment dataset), by filtering out duplicated reviews and rearranging the EDU segmentation. The contents of the Zirn dataset are retrieved from Amazon[11], subdivided into three categories “Cell Phones & Services”, “Gourmet

[11] www.amazon.com

Food" and "Kitchen & Housewares". Three annotators labelled all *passages* of reviews as positive, negative and neutral, where a passage was defined as "a sequence of words sharing the same opinion." The boundaries of passages were chosen by the annotators independently also. For each annotator, the input is the reviews text files, and the output is a list of passages with their polarities. Fleiss kappa score was used to measure the inter-annotator agreement, 0.40 to 0.45 for negative reviews (*fair agreement*) and 0.60 to 0.84 for positive reviews (*strong agreement*).

For each word in the corpus, a polarity label was given as follows:

1. Find out three polarity labels that three annotators have chosen for the respective passages containing the word.
2. If the majority of the three labels is positive or negative, it is taken as the polarity label of this word;
3. Otherwise the general polarity of this entire review is given to the word.

Once the labels for every word are determined, the polarity label of each EDU is assigned as the majority label of words in this EDU. This procedure can be taken in such a way that a polarity label was chosen for each word according to its polarity in the context. Then the polarity of an EDU is defined as the majority label of polarity labels of all tokens in this EDU. Using this method, the problem of EDU boundary disagreement could be solved and the rearrangement of EDU segmentation is more flexible. The annotators can choose the texts inside which they think the sentiment is continuous so that the annotations are not limited by some other EDU segmentation tool.

Consider a simple example as follows:

Example 3.2. *This is a good knife, and that also.*

Assume three annotators choose the same word boundary for this sentence, and all label this as *positive*. The task is given the EDU *and that also.*, determine its polarity label.

First we need to assign a polarity label to each word in this EDU, starting with *and*. This *and* belongs to an *passage* that three annotators annotated as *positive*, so the polarities of this *and* are *positive, positive, positive* respectively to three annotations. The majority is *positive*, so the polarity of this *and* is positive, according to the annotation. Similar decisions can be made for the other words in this EDU: *that* is *positive*, *also* is *positive.* The majority polarity of this EDU is *positive*, so the gold annotation of this EDU is *positive.*

We use the segmentation results of the Feng-Hirst parser to obtain the EDUs and use the above method to generate the polarity labels for each EDU. For each EDU, a polarity label is attached. (There are no aspect annotations in this dataset, as in the TripAdvisor Dataset). An example of one item in this dataset is given as follows:

Example 3.3. *Multi-Domain-Sentiment Dataset Annotation*

This is NOT the result of customer abuse but a manufacturing defect. **neg**

The Multi-Domain-Sentiment dataset we extracted contains 97 reviews (5677 EDUs in total), 31 under *Cell Phone & Service* category, 31 under *Gourmet Food* category, 35 under *Kitchen & Housewares* category. The distribution of polarity counted by number of EDUs can be found in Table 3:

Category	positive	negative	neutral	**Sum**
Cell Phones & Service	833	1093	359	2285 (40.25%)
Gourmet Food	426	463	244	1133 (19.96%)
Kitchen & Housewares	816	1000	443	2259 (39.79%)
Sum	2075	2556	1046	5677

Table 3. Polarity Distribution, Multi-Domain-Sentiment

3.2 Lexical Resources

We would like to set up a simple lexicon-based baseline for this work, in order to compare the performance of our methods. We introduce three lexical resources to calculate a lexical score as the feature used in this lexicon-based method. There are several publicly available lexical resources for sentiment analysis. These resources provide list of words and their polarities, i.e., whether they are positive or negative or neutral. Some resources provide some additional information such asa polarity score that gives a numerical indication of how positive or negative a word is. In order to have a wider coverage and a better fitting score, we use three lexical resources. As introduced in Chapter 2, they are referred to AFINN, Hu-Liu and Opinion Finder Lexicon. We use each lexical resource to classify the polarity of an EDU separately, and use a voting schema (as introduced later in Chapter 4) among these three results to get the final lexical decision.

AFINN [Nielsen (2011)] is an affective lexicon collected by Finn rup Nielsen. It is developed in the *Responsible Business in the Blogosphere* project whose purpose is "to investigate how corporate reputations as responsible business are constructed online in virtual social networks." AFINN consists of 2477 English words, originally extracted from Twitter and later extended. Each word is rated by a valence value from -5 to +5. There are 878 tokens with positive scores and 1599 tokens with negative scores, respectively 35.45% and 64.55%.

Hu-Liu is a word list being collected over years starting from [Hu and Liu (2004)] by Minqing Hu and Bing Liu. This list contains two sub-lists, one for positive words and one for negative words. There are 6789 words in total, 2006 positive (29.55%) and 4783 negative (70.45%). The authors also mentioned they included some misspelled words since they appear frequently in social media content.

The Opinion Finder Lexicon provides more detailed information. According to the instructions, there are 6 aspects of descriptions for each word:[12]

a. *type - either strongsubj or weaksubj. A clue that is subjective in most contexts is considered strongly subjective (strongsubj), and those that may only have certain subjective usages are considered weakly subjective (weaksubj).*

b. *len - length of the clue in words. All clues in this file are single words.*

c. *word1 - token or stem of the clue*

d. *pos1 - part of speech of the clue, may be anypos (any part of speech)*

e. *stemmed1 - y (yes) or n (no). If stemmed1=y, this means that the clue should match all unstemmed variants of the word with the corresponding part of speech. For example, "abuse, pos1=verb, stemmed1=y", will match "abuses" (verb), "abused" (verb), "abusing" (verb), but not "abuse" (noun) or "abuses" (noun).*

f. *priorpolarity - positive, negative, both, neutral The prior polarity of the clue. Out of context, whether the clue seems to evoke something positive or something negative.*

In this lexicon there are 8221 *clues* among which there are 6878 distinct words (The difference is due to the fact that the same word with different part of speech tags will result in different clues). Although this lexicon provides rich descriptions for each word, we take only two of the subjects: word and priorpolarity. It is found in this lexicon that the priorpolarities of the same word with different POS tags are consistent, which makes it unnecessary to distinguish between different POS tags. We extract from this lexicon a list of (word, priorpolarity) pairs and use it for our lexicon-based features.

[12] Following description taken from official instructions by the developers, a *clue* here refers to one line in this lexicon.

3.3 Summary

In this chapter we discussed the datasets and additional lexical resources used in this thesis. The TripAdvisor dataset has a smaller size, but contains annotations of aspect information. The Multi-Domain-Sentiment dataset is larger, and includes three different domains. Three additional lexical resources are used to construct reliable lexicons for the lexicon-based approaches of our sentiment prediction task.

4 Sentiment Classification

In this chapter, we describe the features used in our sentiment classification task. The first set of features contains lexical scores for EDUs, from three lexical resources; the second set of features considers the influences of adjacent EDUs; the other two sets of features respectively model parsing results of the Feng-Hirst parser and the Lin parser.

The basic assumption of using discourse features is as follows: when we want to predict the sentiment of a certain EDU (we call it current EDU), we can use the sentiments of other EDUs which have some discourse relation(s) with the current one. The type of discourse relation might tell whether there is going to be a polarity shift from the linked EDU and the direction of the polarity shift. For example, a contrast relation is likely to trigger a polarity shift, and the shift is probably from negative to positive or vice versa. The goal of modelling discourse features is to extract information from our parsing results and normalize this information in order to use it for training the machine learning algorithm.

This chapter is organized as follows, section 4.1 explains a simple lexicon-based method which we take as a baseline; section 4.2 introduces how we use the polarities of adjacent EDUs to predict the polarity of the current EDU; section 4.3 and 4.4, respectively, present the procedure for modelling of the parsing results of the Feng-Hirst parser and the Lin parser.

4.1 Lexicon-based Features

As there are many ways to take advantage of lexical resources in the sentiment analysis task, as introduced in Chapter 2, we would like to take a simple one as the baseline of this work. We implement two methods for using three of our lexical resources: the first one takes an EDU as a bag-of-words; the second one considers syntactic structure.

4.1.1 Simple Lexical Score

The bag-of-words method works as follows. For each EDU, we look up the polarity score of each word in three lexical resources (any unfound word will be assigned 0), and sum up the scores of each word of the EDU. Since we have three dictionaries, we have three summed scores for each EDU. We take each score as the decision (1 for positive, -1 for negative, 0 for neutral) of each dictionary, and vote among these three decisions. For example, a voting result of 3 means all three dictionaries give a positive score for certain EDU. This voting result is taken as the simple polarity score. Its value is an integer from range [-3,3] corresponding to most negative to most positive. This method is applied to both of the datasets and used as the baseline for the experiment.

The bag-of-words method additionally considers negations, which is common in sentiment analysis. Once a negation connective[13] is found in an EDU, the polarity score of the rest will be multiplied by **-1**, which means the polarity of the rest of the EDU is inverted. Consider the following example,

Example 4.1. *negation*

This(0) is(0) not(0) a(0) nice(3) place(0) to(0) stay(0).

The numbers in brackets are the polarity scores found in AFINN for each word. The final AFINN polarity score of this EDU is calculated as:

$$(0 + 0 + 0) + (-1) * (0 + 3 + 0 + 0 + 0) = -3$$

4.1.2 Relevant Aspect Classification

We also implement another method for the TripAdvisor dataset, taking into consideration more syntactic information. In this dataset we have the annotations of aspect information and we want to distinguish contents within one EDU by whether it is aspect relevant. If some part of an EDU is not relevant to the topic/aspect being talked about in this EDU, we will ignore the polarity value of this part of text. To understand the motivation of this method, see the following example,

Example 4.2. *Aspect relevant*
On that terribly rainy night we were glad to meet that helpful staff. **service** pos

In this EDU, the token *terribly* bears negative sentiment while it is not relevant to the current aspect (service). So its polarity score shouldn't be counted into the polarity result of this EDU.

A procedure of aspect classification is needed since we want to know which part of an EDU is aspect relevant. We implemented a simple method since our main focus is using discourse structure for the sentiment prediction task. The goal of this step is to gather representative noun phrases or nouns of each aspect. With this step we can distinguish between relevant and irrelevant content with regard to the current aspect. Aspect information is obtained from annotations of the corpus. We extract all the nouns and noun phrases in one document using a chunker TreeTagger [Schmid (1994)].[14]

[13] Negation connectives used in this thesis include: "not", "no", "don't", "doesn't", "never", "hardly", "none", "nothing", "nowhere", "neither", "nor", "nobody", "scarcely", "barely", "can't", "won't", "wouldn't", "shouldn't", "couldn't".

[14] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

For each aspect we cluster the most "representative" lemmas such that these lemmas appear more than once in the document. Each aspect corresponds to a bag of representative lemmas. For each EDU, we first identify words relevant to the concerned aspect, using the aspect clusters mentioned above. We then try to find out connections between these aspect relevant words and words that bear a non-zero polarity score. We use the Stanford Dependency Parser [De Marneffe et al. (2006)] to find out these connections. If there exists a dependency between a polarity-bearing word and an aspect-relevant word (or a pronoun), or there exists another word that has a dependency with a polarity-bearing word and an aspect-relevant word (or a pronoun), then this polarity-bearing word is considered as aspect relevant. Words that are not aspect relevant are not counted when calculating the polarity score of a EDU. Consider the example mentioned above,

Example 4.3. *Aspect relevant lexical score*
On(0) that(0) terribly(-3) rainy(-1) night(0) we(0) were(0) glad(3) to(0) meet(0) that(0) helpful(2) staff(0). **service** pos

Scores in the brackets are found in AFINN. The simple polarity score of this EDU would be:

$$sum(0, 0, -3, -1, 0, 0, 0, 3, 0, 0, 0, 2, 0) = 1$$

Assume the word "night" is not found in the aspect relevant cluster of the current aspect ("service") and "staff" is. In this aspect relevant method, we wouldn't consider polarity-bearing words that have close connections with "night". We use the parsing results of this EDU to judge which words are close connected to "night" and "staff". The following dependencies are dependencies with "night" and "staff" involving:

advmod(rainy-4, terribly-3), amod(night-5, rainy-4), nsubj(glad-8, we-6), amod(staff-13, helpful-12)

"terribly" and "rainy" are closely connected to "night" which is an aspect-irrelevant word. "glad" and "helpful" are connected to aspect-relevant words. A decision is made that "terribly" and "rainy" are aspect irrelevant, while "glad" and "helpful" are aspect relevant. The aspect-relevant polarity score is updated as:

$$sum(0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0) = 5$$

The polarity scores of "terribly" and "rainy" are filtered out. The aspect-relevant score could better reflect the real polarity trend for this EDU, since the attitude the write wanted to express in this EDU is *positive.*

With this extra step, we are able to filter out the noise from aspect irrelevant words and get a more accurate polarity score. The result of this method is referred to as aspect-relevant polarity score, to be distinguished from the results of the previous method, namely simple polarity score.

4.2 Adjacency-based Features

In order to predict the sentiment of an EDU (current EDU) that may not have an evident lexical score, we might want to consider the polarities of EDUs adjacent to the current EDU. This method is based on the assumption that people tend to write coherent text. This coherence exists between text spans (e.g. EDUs) that are close to each other. So it is very likely that adjacent EDUs share the same polarity as the current EDU. Consider the following example from TripAdvisor, in which we want to predict the sentiment for EDU(4):

Example 4.4. *Adjacent EDUs*

(1) *It is outdated ,* **rooms** neg

(2) *the television was old and did not work properly ,* **rooms** neg

(3) *the phone did not work properly ,* **rooms** neg

(4) *and there was this exercise bike next to the bed .* **rooms** *?*[15]

In order to predict the sentiment of EDU(4), we can take into consideration EDU(1) to EDU(3) and predict the fourth one as *negative* also.

To generalize this adjacency-based method, we can set up a window of how many adjacent EDUs should be considered. The polarities of these EDUs will be modelled as a feature for the current EDU with respect to their relative location. For example, in order to predict the sentiment of EDU(4) with window size 2, we add a feature named *previous-1* with the value *negative*, and a feature named *previous-2* with the value *negative. next-1* and *next-2* could be modelled similarly. The polarities used here could be the predicted polarities or the gold standard polarities.

4.3 RST-based Discourse Features

For each review, the Feng-Hirst parser gives a RST parse tree as result. The leaves of the tree are EDUs. Two EDUs are connected by a relation, and this relation is taken as a relation node for further tree construction. The term *relation node* is defined as the sub-tree structure with this relation as the root.

Our task here is how to model the parse tree such that each EDU gets a set of features which can indicate the current EDU's relations with others. These relations include the type of discourse relation that links the current EDU to other EDUs, the role the current

[15] In the examples, ? indicates an unknown sentiment we want to predict.

EDU plays in a certain relation (say, nucleus or satellite), and whether there is an aspect shift from another linked EDU (used for the TripAdvisor dataset only). Since the tree structure modelling is complicated, we start with a representation of direct relations that connect two EDUs (leaves in the tree), and then explain how to extend the representation for relations that connect nodes.

4.3.1 Relation Representations

Direct relation representation .

Consider the following example from our parsing results of the Feng-Hirst parser for the TripAdvisor dataset:

Example 4.5. *Contrast relation*

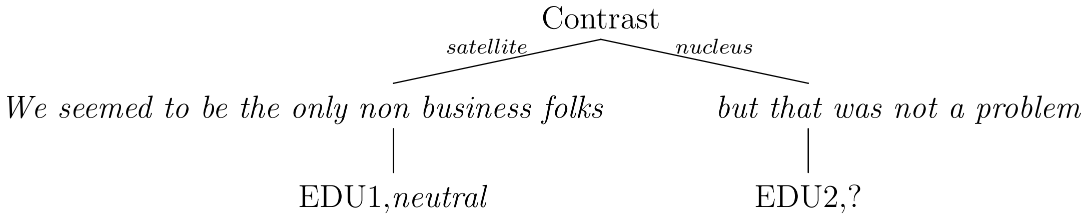
Contrast [S][N]

[We seemed to be the only non business folks] $_{\text{EDU1}}$

[but that was not a problem] $_{\text{EDU2}}$

EDU1 (satellite) and EDU2 (nucleus) are connected by a contrast relation. When we want to predict the sentiment of EDU2, we would like some help from EDUs we've seen in the text, in this case, EDU1. Given that EDU1 is neutral, the information we can use for predicting the sentiment of EDU2 is that it is the Nucleus of a contrast relation in which the other EDU is neutral, formally *nucleus-contrast-neu*.

To make it easier to understand, we can think the role (nucleus or satellite) as the name of edge in tree structure. The available information we have for predicting sentiment of EDU2, represented as a tree structure, is shown in the following diagram:


Contrast
satellite
nucleus
We seemed to be the only non business folks
but that was not a problem
EDU1,neutral
EDU2,?

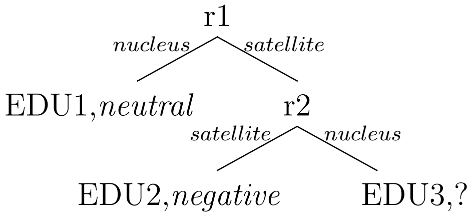
For EDU2, *nucleus-contrast* is the path it has to go through in order to connect to EDU1, and polarity of EDU1 is neutral. Then we add *nucleus-contrast-neu* as a feature for EDU2 and name *nucleus-contrast-neu* the path of EDU2 in this contrast relation. We don't include in the path the edge that connects EDU1 (not like nucleus-contrast-**satellite**) because the role the current EDU plays is the central concern and the other role can be inferred.

Longer path representation .

In the previous section we discussed how to model a relation that links two EDUs directly. Modelling higher levels of the RST tree is done in a similar way. There are two more issues for describing indirect relations: the first one is how to extend the path representation if it is longer; the second is how to determine the polarity of a relation node.

Consider the following example:

Example 4.6. *Longer path*



There are two usable paths for EDU3, the first one is the direct path of relation r2:

(1) *nucleus-r2-neg*

the second one is a longer one, up to relation r1:

(2) *nucleus-r2-satellite-r1-neu*

For a given EDU, there is a discourse relation that takes this EDU as its child directly and there is another node/leaf as the other child of this relation. If this relation node is not the root node of the parse tree, there will be another relation node that takes this relation node as a direct child and provides a sibling node for this relation node. So from the given EDU, every time we seek upwards, we would find a relation that takes the current node as source node and another node as target node until we reach the root node. For every relation we go through, we can add a path feature for the current EDU with the format "p-s" where p stands for the path the current EDU has to go through to reach the relation and s stands for the sentiment of the other child node of this relation. Consider Example 4.6, we start from EDU3 and look upwards. A relation r2 is found, the path to r2 is *nucleus-r2* and the polarity of the other node/leaf of r2 is *negative*, so we get the first path Path(1) with the "p-s" format; then we continue looking up, relation r1 is found, the path from EDU3 to r1 is *nucleus-r2-satellite-r1* and the polarity of the other node of r1 is *neutral*, so Path(2) is obtained.

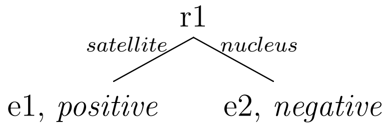
Once we extend the representation to this level, every EDU is involved into one relation at each level of the tree, which is not realistic. Some EDUs play their roles locally and locally only. In RST, *satellites* are those EDUs that are “less important”, and shouldn’t influence others so much. So in our path representation, we filter our some paths that get more than two satellites involved. Consider EDU2 in Example 4.6., the path for EDU2 at length two is *satellite-r2-satellite-r1-neutral*. There are two satellites in this path, hence we consider this path nonsignificant and filter it out. The intuition of this filter is that satellites are the complement part in one relation and they are weakly involved in upper level relations. This procedure is done during the generation of discourse representations.

Polarity of relation node We also need to define the polarity of a relation node. The polarity of a RST-style discourse relation node depends on polarities of the leaves under this relation. Its value is assigned as follows,

1. If there is only one nucleus in this relation, the polarity of this relation is the same as the polarity of its nucleus child node (relation clause or leaf EDU).
2. If there is more than one nucleus in this relation, the polarity of this relation equals to the majority polarity of these nuclei.
3. If there is more than one nucleus in this relation and there is no majority, the polarity of this relation is set as ‘neutral’.

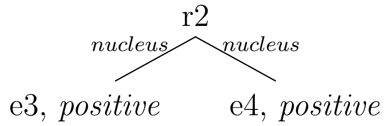
Consider the following examples:

Example 4.7. *Relation node polarity Rule 1*



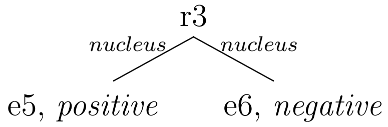
A relation node r1 consists of two EDUs e1 and e2, and e2 is the only nucleus. The polarity of the nucleus is assigned to the clause. So the polarity of r1 is negative.

Example 4.8. *Relation node polarity Rule 2*



A relation node r2 consists of two EDUs e3 and e4, and both of them are nuclei. The polarity of their agreement is assigned to the node. So the polarity of r2 is positive.

Example 4.9. *Relation node polarity Rule 3*

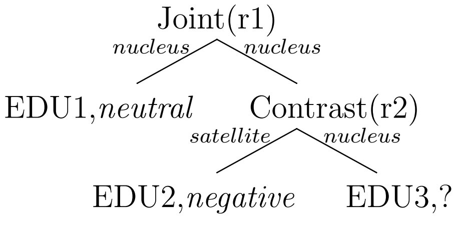


A relation node r3 consists of two EDUs e5 and e6, and both of them are nuclei. The polarity of r3 is set as neutral since there is no majority among the polarities of the child nodes.

Path length It has been discussed that an RST tree is a tree where leaves are EDUs and nodes are discourse relations that connect its children. We use *path length* to describe how far it is for an EDU to reach a relation node. For a tuple (r, e, n) where r is a discourse relation that connects the current EDU e and the other node/leaf n, the path length of this node r with respect to EDU e is defined as the number of relation nodes in the path from e to r. Take paths in Examples 4.6, Path(1) *nucleus-r2-neg* has a path length 1, Path (2) *nucleus-r2-satellite-r1-neu* has a path length 2.

From the structural information of a parse tree, we can extract more information to distinguish discourse relations of different heights. Consider a pair (r1, e, n1) and (r2, e, n2) with e as the current EDU whose sentiment we want to predict, if r1 has a longer path length than r2, then relation r1 might be weaker, which makes r2-n2 more important than r1-n1 when predicting the sentiment of e. Consider the following example,

Example 4.10. *Path length influence*
[My partner and I stayed for two weekend nights.] EDU1 *[While it was not a hideous hotel experience,]* EDU2 *[there was no joy going back to our room.]* EDU3



In order to predict the polarity of EDU3, we can use the polarities of EDUs that have discourse relations with it, in this case, EDU1 and EDU2. EDU3 and EDU2 are connected directly by a contrast relation r2 (path length 1); EDU3 and EDU1 are connected indirectly through another node and the corresponding relation is a joint relation r1 (path length 2). A normal interpretation is that EDU3 is more closely related to EDU2 than to EDU1, which means the contrast relation r2 should have more influence than the joint relation r1 when predicting the polarity of EDU3.

The more general assumption of this scenario is that the further a relation is from an EDU, the less important the role it could play on sentiment prediction. In order to test this assumption, we implement two models based on the parsing results of the Feng-Hirst parser: one takes into consideration all available discourse relations that are reachable from the current EDU (the one whose sentiment we want to predict); the other one is limited to relations within a certain path length.

4.3.2 Different RST-based Models

Full path model In this model all relations are considered and treated equally. Consider Example 4.10, where EDU3 is involved in two discourse relations: Contrast relation r2 connects two EDUs, while Joint relation r1 connects a EDU and a relation node. There are two paths that could be taken as useful information for predicting the polarity of EDU3, respectively,

(a) *nucleus-Contrast-nucleus-Joint-neutral*

(b) *nucleus-Contrast-negative*

Path (b) describes the Contrast relation between EDU2 and EDU3 and the fact that EDU3 is the nucleus in this relation and EDU2 is negative. Direct relations such as this one are clear, while indirect relations like Path(a) are more complicated. Facts we can extract from this joint relation include: The Joint relation connects one EDU and one relation node; the polarity of this EDU is neutral; both this EDU and the node are nuclei. When we use Path(a) to predict the polarity of EDU3, what's actually being used is the influence from the Joint relation r1. In other words, Path(a) is applicable because EDU3 is a descendant of a node which is a child of this Joint relation. So in Path(a), the Contrast relation is more a connecting node than a relation that provides more expressive information. Motivated by this, we can rewrite Path(a) as a short version Path(a.s),

(a.s) *nucleus-Joint-neutral*

Its meaning is *current EDU is the nucleus of this joint relation, and the other argument of this relation is neutral.* We can benefit from the short rewriting role such that it's easier for the classifier to find the regularity of how each relation works since the representation is uniform. By rewriting the path in a shorter version, we can also avoid the problem of feature sparsity. In fact, it is shown from our datasets that if the longer version of the path feature is used, the feature space is too sparse to train an applicable model.

Path length controlled model This model restricts the relations to a certain path length. The assumption is that EDUs that are closer to each other tend to express related topics and to be more coherent. This can be applied to the discourse parse tree: For a certain EDU, if using a relation with a longer path to model discourse structure, the further the connecting text span is, the less coherent it is expected to be. In other words, if we choose a higher level discourse relation to model the discourse structure of an EDU, it might not be helpful for the sentiment prediction task. Moreover, it might introduce noise if it is treated equally as other lower level relations.

In order to test this, we implement a path length controlled model which limits the path length from the current EDU to a relation no longer than two. It means only two types of relations are considered, the first type of relations connect the current EDU to some other node/leaf; the second type of relations connect the parent node of the current EDU to some other node/leaf. We set the maximum length to two, but it is extendable; another length restriction can be replaced and tested in the model easily.

In this model, we also distinguish the path length of a relation in the feature space. For example, for a contrast relation with path length one where the current EDU is the nucleus and the other node is negative, instead of using *nucleus-contrast-neg* as the feature, we use *length1-nucleus-contrast-neg.* Since our path length limitation is 2, the increase in the size of features is still manageable.

4.4 PDTB-based discourse features

For each review, the Lin parser identifies all the explicit and implicit discourse relations including the connectives (if any) and the corresponding arguments (Arg1 and Arg2). Modelling the relations as features for PDTB-style parsing results is not as complicated as modelling RST trees, but it is not obvious how to determine the boundaries of EDUs. In this section, we first explain how the parsing results of the Lin parser are mapped to EDUs, and then introduce the procedure of feature modelling.

4.4.1 EDU Segmentation

The annotations in our datasets are EDU-based such that every EDU has a polarity label, while the parsing results of the Lin parser (any other PDTB-style parsers should be similar) segment texts into *Connectives, Arg1s* and *Arg2s.* There are two differences between this representation and RST-style EDU representation:

(1) Both RST and PDTB representations from the Lin parser's parsing results take two major text spans. But the parsing result has a special *connective* which doesn't

belong to Arg1 or Arg2. We would like to map arguments to EDUs, but there will be some cases that PDTB arguments and annotated EDUs cannot match perfectly since EDUs contain the connective tokens but arguments don't.

(2) As discussed in Chapter 2, the argument of a PDTB discourse relation doesn't have to be continuous. Some other text could embed into one argument. But RST EDUs are continuous, as is our annotated data.

To solve Issue(1), in the procedure of mapping the Lin parser's parsing results to annotated EDUs, we accept loose mappings. If a text span of the parsing result and the content of an annotated EDU is similar enough such that the difference is no more than one token, the mapping is considered tenable. Then we can analyse the relations between EDUs in this PDTB-style discourse structures. Consider the following example:

Example 4.11. *PDTB EDU mapping*

(a) {*Exp_1_Arg1* **There 's a great big tv** *Exp_1_Arg1*} {*Exp_1_conn_Conjunction* **and** *Exp_1_conn*} {*Exp_1_Arg2* **the bathroom is quite nice** *Exp_1_Arg2*}

(b) *EDU1: There 's a great big tv* **amenities** pos
EDU2: and the bathroom is quite nice **rooms** pos

(a) is the representation of the parsing results from the Lin parser, and (b) is the annotated data. In (a) *Exp_1* indexes this is the first explicit discourse relation in this text and *_conn* indicates a connective. We number the two EDUs as EDU1 and EDU2. EDU1 could be mapped to the Arg1 in the relation, while there is difference of one token ("and") between EDU2 and Arg2. This difference is ignored and we consider EDU2 and Arg2 to be a match. Similar to an RST relation then, a PDTB-style Conjunction relation holds between EDU1 and EDU2.

Issue(2) happens quite often in the parsing results, and the solutions varies depending on the concrete cases. Consider the following example:

Example 4.12. *PDTB argument that crosses EDU boundaries*

{*Exp_3_Arg1* **We were on the 10th floor which I was pleased about** {*Exp_3_conn_Cause* **because** *Exp_3_conn*} {*Exp_3_Arg2* **we had a view of the city plus** *Exp_3_Arg2*} **we did not hear a lot of trafiic .** *Exp_3_Arg1*}

Exp_3 indexes that this is the third explicit discourse relation in this text and *_conn* identifies the connective. In this example Arg2 is embedded within Arg1. We can divide the text into three EDUs if there is no other constraint, but we need to map this to the annotated data, which is as follows:

Example 4.13. *Annotated data which cannot map to PDTB results*

1. *We were on the 10th floor which I was pleased about* **rooms** pos
2. *because we had a view of the city plus we did not hear a lot of trafiic .* **rooms** pos

The contents of EDU2 appear both in the Arg1 and Arg2 of the Cause relation of the parsing result in the previous example, which results in a case that this EDU cannot be directly mapped to the parsing results. We are not able to assert whether EDU2 belongs to the first or second argument of this relation, and cannot assign to it the *relation-polarity_of_the_other_argument* pattern either. In this case, we drop this relation because incorrect mappings introduce noise.

Once all annotated EDUs and arguments are mapped, we can take the discourse relations between arguments and build the PDTB-style discourse model for our sentiment prediction task.

4.4.2 Relation Representations

One EDU (argument) in the PDTB parsing result involves at most two discourse relations, and sometimes none, since we filter out the implicit relations. The PDTB-style discourse relations model local and linear structures, which means all relations connect only EDUs rather than relation nodes. Modelling PDTB discourse relations is quite similar to modelling direct RST relations. Instead of using nucleus/satellite to distinguish different arguments, PDTB relations use Arg1 and Arg2. Since Arg1 and Arg2 have their meanings (See Chapter 2), we decide to keep the distinction.

Consider the following example from the parsing results of the TripAdvisor dataset:

Example 4.14. *PDTB feature modelling*

(1) {*Exp_2_Arg1* The location is great *Exp_2_Arg1*} {*Exp_2_conn_Conjunction* and *Exp_2_conn*} {*Exp_2_Arg2* it 's a beautiful , grand old hotel. *Exp_2_Arg2*}

(2) *The location is great* **location** pos
and it 's a beautiful , grand old hotel. **amenities** ?

(1) is the parsing result of the Lin parser and (2) is from the annotated data. Suppose we want to predict the sentiment of the second EDU, what can be extracted from the parsing results includes the fact that two EDUs hold between a Conjunction relation, the second EDU is Arg2, and Arg1 is positive. We represent this as *Arg2-Conjunction-pos.*

This representation is similar to the representation of RST direct relations (those relations that has a path length 1).

The PDTB features are sparse. There are two main reasons:

1. All the implicit relations are filtered out.
2. Unlike RST, it's not necessary for every EDU to be involved in at least one discourse relations. In fact in the parsing results of the Lin parser, there are many EDUs standing alone, not connecting to any other EDU.

The second reason is also the major difference between RST and PDTB. RST cares more about a global discourse tree whose leaves are EDUs, while PDTB pays more attention to discovering discourse relations whose two arguments could be text spans of any length. In other words, RST concerns discourse structures at all levels and PDTB concerns more local structures. We will evaluate both methods and examine which representation is better in our sentiment prediction task.

4.5 Aspect Shift

We also hypothesized in Chapter 1 that discourse relations may signal a shift of aspect. In the TripAdvisor dataset, there are annotations of aspects. So when modelling the path representation, we could also add some information about aspect shift. Consider Example 4.14., the original representation is *Arg2-Conjunction-pos*, and we find there is an aspect shift from these two EDUs. So we rewrite this representation as *Arg2-Conjunction-pos-Yes* where "Yes" indicates that the aspect information of these two EDUs are shifted.

As discussed in Chapter 1, the aspect shift information may help us predict the change of sentiment from EDUs to EDUs better. But we only have the aspect annotations in one dataset (the TripAdvisor dataset), and the size of the dataset is small. If the aspect shift information is introduced into the representations, the features will be more sparse and hence it will be harder for the classifier to be trained.

4.6 Summary

In this chapter we discussed four sets of features used in this thesis. The first set is lexicon based and assigns a polarity lexical score to each EDU; the second set is adjacency based which considers the sentiments of surrounding EDUs to predict the current one; the third and fourth sets are discourse based: the RST discourse parse tree is linearized to model the relations between EDUs; the PDTB relations are modelled in a similar way.

5 Experiments

In this chapter we explain the implementation of features introduced in Chapter 4 on our two datasets. First we introduce the settings of the experiments, including the tool and algorithm used in the experiments. Then we analyse the parsing results, to be aware of the distribution of discourse relations. Finally we introduce the evaluations of different models, respectively two baselines, RST-based models and PDTB-based models.

5.1 Experimental Settings

We use *Weka* [Hall et al. (2009)] to perform our classification task, with a *Logistic Regression* classifier.

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka contains a collection of algorithms for data analysis and model prediction. After modelling the features introduced in Chapter 4, we transform the data into the format that Weka takes and perform the classification task. We use 10 fold cross-validation and measure the results by labelling accuracy.

The Logistic Regression classifier Weka implemented was based on [le Cessie and van Houwelingen (1992)]. The algorithm builds and uses a multinomial logistic regression model (in our case, three category classification). A *Logistic* function, which is also referred to as *sigmoid* function, is employed in Logistic Regression classifier. It takes a vector of variables **x** as input and outputs the probabilities of **x** to each class. For a binary classification problem (say, two classes y = 0 and y = 1), the probability of given data point **x** belonging to each class is defined as follows where **w** is the parameter vector:

$$p(y=1|\mathbf{x};\mathbf{w}) = p_1(\mathbf{x}) = \frac{1}{1+e^{-w*x}}$$

$$p(y=0|\mathbf{x};\mathbf{w}) = 1 - p_1(\mathbf{x})$$

This classifier could also work for multi-class classification problem. For a k class classification problem, the probability of **x** belonging to class K is:

$$p(y=K|\mathbf{x}) = \frac{1}{1+\sum_{l=1}^{K-1} exp(\mathbf{w_l} * \mathbf{x})}$$

5.2 Statistics of Parsing Results

In this section we analyse the parsing results of different parsers on our datasets. We would like to see the distribution of different discourse relations among texts from different domains. We take the two different datasets as two domains: the TripAdvisor dataset for hotel reviews; the Multi-Domain-Sentiment dataset for product reviews.

Relation	**TripAdvisor**		**Multi-Domain**	
Elaboration	881	60.67%	4032	62.58%
Joint	204	14.05%	562	8.72%
Contrast	119	8.20%	249	3.86%
Explanation	54	3.72%	242	3.76%
Evaluation	44	3.03%	167	2.59%
Background	37	2.55%	218	3.38%
Enablement	22	1.52%	91	1.41%
Condition	21	1.45%	80	1.24%
Attribution	19	1.31%	428	6.64%
Temporal	18	1.24%	68	1.06%
Cause	16	1.10%	44	0.68%
same-unit	8	0.55%	194	3.01%
Summary	7	0.48%	44	0.68%
Comparison	2	0.14%	11	0.17%
Topic-Change	0	0.00%	1	0.02%
textual-organization	0	0.00%	2	0.03%
Topic-Comment	0	0.00%	1	0.02%
Manner-Means	0	0.00%	9	0.14%
Sum	1452		6443	

Table 4. RST relation distribution

Table 4 shows the distributions of RST relations among two datasets, measured by frequency and portion, sorted by counts in the TripAdvisor dataset. Both datasets contain over 60 percent Elaboration relations. There are also quite many Joint, Contrast, Explanation, Evaluation, Attribution relations. The frequency rankings of relations between the two domains are not so different, except that Attribution relation appears more often in product reviews.

Table 5 shows similar statistics for the PDTB parsing results. No PDTB relations take more than 50 percent and the distribution is more uniform. Conjunction and Contrast make up the majority of relations.

By comparing Table 4 and Table 5, we can see that the number of relations discovered in the Lin parser is much smaller than for the Feng-Hirst parser (around one third). It means the modelling of PDTB is more sparse than the one of RST. The number of PDTB relations is also less than half of the number of EDUs (587 v.s. 1541 for the TripAdvisor data and 1824 v.s. 5677 for the Multi-Domain-Sentiment dataset), which means there are

Explicit Relation	**TripAdvisor**		**Multi-Domain**	
Conjunction	218	37.14%	510	27.96%
Contrast	119	20.27%	336	18.42%
Cause	63	10.73%	255	13.98%
Condition	57	9.71%	217	11.90%
Synchrony	53	9.03%	217	11.90%
Asynchronous	37	6.30%	191	10.47%
Concession	23	3.92%	36	1.97%
Alternative	10	1.70%	42	2.30%
Restatement	3	0.51%	12	0.66%
List	2	0.34%	1	0.05%
Instantiation	1	0.17%	5	0.27%
Exception	1	0.17%	1	0.05%
Pragmatic-condition	0	0.00%	1	0.05%
Sum	**587**		**1824**	

Table 5. PDTB relation distribution

some EDUs are not involved in any relations (otherwise the number of relations would be at least half of the number of EDUs).

5.3 Evaluation

5.3.1 Lexicon-based Baseline

We set up two baselines in our experiments. The first one implements the lexicon-based method (referred as *L*) and the second one takes the lexicon-based method as its basis, and adds adjacency-based features.

The lexicon-based method assigns each EDU a score from -3 to 3, from most negative to most positive. The classification procedure is simple: if the score is a positive number, then classify this EDU as positive; if the score is negative, then classify this EDU as negative; if the score equals to 0, then classify this EDU as neutral. For the TripAdvisor dataset, there are two ways of getting this lexical score: one is the simple score, the other one is the aspect relevant score. They are referred as *L-lex* and *L-asp* respectively. For Multi-Domain-Sentiment dataset, there is no annotations of aspects, so there is only *L-lex* score. The results measured by accuracy are shown in Table 6. The aspect-relevant method doesn't outperform the simple score. This might because there are not enough polarity-bearing words in a short text span as EDU; filtering out some by aspects might lead to more EDUs without any sentiment scores. The lexicon-based baseline for the

Multi-Domain-Sentiment dataset performs worse than the TripAdvisor dataset, this might be because we use the segmentations of the Feng-Hirst parser for the Multi-Domain-Sentiment dataset and the segmentations produced by this parser are often too short to include some polarity-bearing words. These short EDUs will be classified as *neutral* while in this dataset there are not many true neutral EDUs.

Model	TripAdvisor	Multi-Domain
L-lex	61.0	52.0
L-asp	50.3	-

Table 6. Lexicon baselines, classification accuracy

5.3.2 Adjacency-based Baseline

The adjacency-based method considers the sentiments of previous and next EDU or EDUs. We take two window sizes. The Prev2Next2 (stands for adjacent window width 2) model considers two adjacent EDUs both before and after the current EDU (four in total). The Prev1Next1 model, similarly, means adjacent window width 1 with one EDU in each direction from the current EDU. Another method of considering only the previous EDU is tried and named Prev. We set this Prev model since in real prediction system, only known contents should be used to predict the unknown EDUs. The value of each feature is the sentiment of the corresponding EDU from gold standard, *'Start'* if it's the first EDU of a review, or *End* if it's the last EDU. If the corresponding EDUs are not the starting or ending EDUs, the polarities from the gold standard are used. Table 7 shows a mock-up example of how the feature representation works.

EDU	Prev1	Next1
EDU1, pos	Start	pos
EDU2, pos	pos	neg
EDU3, neg	pos	neg
EDU4, pos	neg	End

Table 7. Adjacency representation

The TripAdvisor dataset also has aspect annotations. We add some more information about whether there is an aspect shift from the current EDU to previous/next EDUs. For example, if the polarity of the previous EDU is "pos" and there is an change of aspects, the feature will be valued as *pos-Yes* instead of *pos*. The aspect shift feature is named as *Asp* and it applies to the TripAdvisor dataset only.

Table 8 shows the accuracy results after adding adjacency-based features, where the relatively better results are bonded. There is a significant improvement for both datasets. The results show that if we take into consideration adjacent EDUs, the prediction is more accurate. Adjacency can be taken as a simple idea of measuring how two EDUs are related. Although adjacency doesn't indicate rich linguistic information, it suggests to some degree the continuity of polarity expressions among EDUs. In results of models on the TripAdvisor dataset, adding *Asp* improves the performance, but not significantly. It might be more helpful if the size of the TripAdvisor dataset were larger.

	TripAdvisor				**Multi-Domain**			
	Pos	**Neg**	**Neu**	**All**	**Pos**	**Neg**	**Neu**	**All**
Prev2Next2 + Asp	81.5	77.5	48.6	**71.3**	-	-	-	-
Prev2Next2	79.5	77.8	40.6	68.47	80.9	**90.8**	64.8	**82.39**
Prev1Next1 + Asp	**82**	74.9	**50.4**	71	-	-	-	-
Prev1Next1	78.6	**78.5**	42.3	68.86	**81.9**	90.5	61.6	82.05
Prev + Asp	**82**	76.6	36.3	67.87	-	-	-	-
Prev1	81.9	72.5	39.9	67.3	80.8	82.6	**67.7**	80

Table 8. Adjacency baselines

5.3.3 RST-based

There are two main different models for RST based features: full path model (referred as *RST-FP*) and length-controlled model (referred as *RST-LC*). We use the short version of the full paths since the original ones are too sparse to introduce any new information. The length limitation we set in this implementation is 2. In these models, features extracted from discourse structures are added to the baseline features. The results measured by accuracy are shown in Table 9.

	TripAdvisor				**Multi-Domain**			
	Pos	**Neg**	**Neu**	**All**	**Pos**	**Neg**	**Neu**	**All**
RST-FP + Prev2Next2	**80.6**	75.1	46.9	69.65	87.7	89.9	73.3	86.03
RST-FP + L	79.8	68.1	38	64.45	84.3	88.7	67.5	83.16
RST-LC + Prev2Next2	79.9	**75.6**	**48.9**	**70.11**	**88.3**	**90.3**	**73.5**	**86.49**
RST-LC + L	76.9	68.7	36.8	63.27	86.8	88.3	69.3	84.27
Prev2Next2 (+ Asp)	**81.5**	**77.5**	48.6	**71.3**	80.9	**90.8**	64.8	82.39

Table 9. RST-based features

The general performance of the models on the TripAdvisor dataset is poor, and adding *RST-LC* to *Prev2Next2* even makes the accuracy drop (71.6 to 68). The increase from

Prev2Next2 to *RST-FP+Prev2Next2* is not significant.[16] We attribute the non-successful results of the models on this dataset to the small size of the data. There are only 1514 EDUs in this dataset, while the size of the Multi-Domain-Sentiment dataset is more than triple this.

The results on the Multi-Domain-Sentiment dataset are more promising. The improvements of *RST-FP + Prev2Next2* (86.03) and *RST-LC + Prev2Next2* (86.49) over *Prev2Next2* (82.39) are highly significant. Moreover, *RST-LC + L* (84.27) is significantly better than *Prev2Next2*. Be reminded that the difference between *RST-FP* and *RST-LC* is that *RST-FP* models all possible discourse relations in a discourse tree while RST-LC restricts that the relations should not be in a too high of a position in the tree. The fact that *RST-LC + Prev2Next2* outperforms *RST-FP + Prev2Next2* suggests that discourse relations beyond a certain level in the discourse tree don't contribute to our sentiment prediction task very much.

Given the positive results from the Multi-Domain-Sentiment dataset, we conduct several more experiments on it. First we try using only features of discourse relations for both models *RST-FP* and *RST-LC* without lexical scores or adjacent EDUs' information attached. Then we try adding a smaller amount of adjacent EDUs, say one previous EDU and one next EDU (Prev1Next1). Finally, we construct another model based on *RST-FP* such that in order to predict the sentiment of the current EDU, we use only relations connecting EDUs that are previous to the current one. This is motivated by the natural progress of reading that people infer the meanings of elusive parts according to what has been read in this article instead of something has not been read so far (although they will be helpful to understand the elusive part). This looking-backward-only model is named as *RST-FPB*. It also assumes that when predicting the sentiment of unknown EDU, only previous known EDUs should be considered. The results measured by accuracy are shown in Table 10.

Using discourse features here slightly beats the *Prev2Next2* baseline, but the increase is not significant. Using fewer adjacent EDUs doesn't cause significant changes of the results for both *RST-FP* and *RST-LC* models. This might suggest that with sufficient discourse information, the more distant adjacent EDUs are less important. For the backward-looking models, *RST-FPB + Prev2Next2* is not as good as *RST-FP + Prev2Next2*, but still significantly better than the pure adjacency-based models. We address the reason to the lack of forwards information. It might suggest the influence of sentiment shift works in both directions: for two EDUs connected by a discourse relation, we could not only use the sentiment of the former EDU (the EDU that appears in the text before the other one) to predict the latter one, but also use the sentiment of the latter EDU to guess what has been talked about. Besides, since *RST-FPB + Prev2Next2* beat the adjacent baseline,

[16] We use the McNemar's test [McNemar (1947)], a non-parametric statistical test

Model	Multi-Domain
RST-FP	82.8
RST-LC	83.8
RST-FP + Prev1Next1	86.1
RST-LC + Prev1Next1	**86.3**
RST-FPB + Prev2Next2	**85.2**
RST-FPB + Prev1Next1	84.8
RST-PFB	72.8

Table 10. RST-based features, the Multi-Domain-Sentiment dataset

our model of using discourse information should still be applicable in real applications.

5.3.4 PDTB-based

In our PDTB model, we consider only explicit discourse relations since the Lin parser might not provide reliable enough results of classifying implicit relations. As discussed earlier in this chapter, the coverage of PDTB style discourse relations is low so that some EDUs don't participate in any relations. We run the experiments on two models: PDTB discourse features plus the lexical scores (*PDTB + L*), and *PDTB + Prev2Next2*. The results measured by accuracy are shown in Table 11.

	TripAdvisor				**Multi-Domain**			
	Pos	**Neg**	**Neu**	**All**	**Pos**	**Neg**	**Neu**	**All**
PDTB + Prev2Next2	80.4	76.9	41.6	68.79	**85.3**	90.3	**69.4**	**84.61**
PDTB + L	76.1	75.5	21.2	61.17	60.1	81.7	30	64.26
Prev2Next2 +Asp	**81.5**	**77.5**	**48.6**	**71.3**	80.9	**90.8**	64.8	82.39

Table 11. PDTB-based features

In the TripAdvisor dataset, neither model gives us positive results: *PDTB + L* beats the best lexical baseline to a tiny degree, and *PDTB + Prev2Next2* doesn't outperform *Prev2Next2* model at all. We explain this due to the same reason of the RST models' failure: the dataset is not large enough to recognize the prediction patterns among discourse relations.

In Multi-Domain-Sentiment dataset, *PDTB + L* beat the lexical baseline, but not the adjacent models. It is understandable that the amount of PDTB discourse features is much smaller than the amount of adjacent features. In this dataset there are 6443 EDUs, but only 1824 PDTB discourse relations. After mapping the EDUs and arguments of PDTB discourse relations, we found 3022 EDUs that are not involved in any relation. The amount of all PDTB discourse features is 3705, compared to the amount of all adjacent features (from model *Prev2Next2*): 25772 (4*6443, since we can extract 2 previous EDUs and 2 next EDUs as features for each EDU). The difference of the feature size is obvious, and we think the relatively low coverage is the reason that *PDTB + L* cannot beat *Prev2Next2*. Meanwhile, *PDTB + Prev2Next2* model considers both sets of features and significantly improves the accuracy over *Prev2Next2* (from 82.38 to 84.61).

5.3.5 Comparison: RST vs. PDTB

In this section we compare the performances of two sets of discourse features based on RST and PDTB respectively. Table 12 shows the best and worst performance of each approach, measured by accuracy.

	TripAdvisor	**Multi-Domain**
RST Best	72.6	86.49
PDTB Best	68.8	84.61
RST Worst	63.9	72.8
PDTB Worst	61.2	64.26

Table 12. RST-based features

Note that all the worst performance cases are from models using only discourse features. We can conclude that both RST and PDTB discourse features are not capable to work alone, while after feeding additional features such as lexical scores and adjacent EDUs' information the performance usually improves.

The major difference between RST style and PDTB style is that RST discovers EDUs and constructs a tree structure consisting of these EDUs, while the PDTB focus on discovering the discourse relations whose arguments could contain more than one EDU. Table 12 shows that RST-based approach always outperforms PDTB-based approach in general, which suggests that RST as a theory might be more suitable than PDTB in the task of sentiment analysis on EDU level. The reason is likely to be the higher coverage of texts such that one EDU could get involved in at least one discourse relation.

5.4 Summary

In this chapter we evaluated the features introduced in Chapter 4, focusing on the evaluation results of our discourse models. We first introduced the settings of the experiments and presented some statistics about distribution of discourse relations and EDUs in our parsing results. Our experiments can be divided into two parts: one for two baselines respectively using lexical scores and the polarities of adjacent EDUs; the other one dealing with discourse related features. We've shown that by adding discourse features to baselines the performance obtains a significant improvement. We also showed that our RST style discourse features are generally more useful than our PDTB style features in our task of sentiment prediction on the EDU level.

6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we proposed and investigated representations of discourse relations and use these representations as features for the task of elementary discourse unit (EDU) level sentiment analysis.

In Chapter 1, we first introduced the task of sentiment analysis. We think that sentiment analysis on short text level, precisely the EDU level, is helpful for a better understanding of natural language text, since there might be multiple opinions within one sentence and taking them uniform would miss some information. We hypothesized that discourse structure within one document is an important and effective feature for our EDU-level sentiment prediction task. When we want to predict the sentiment of a particular EDU, we could use the sentiments of other EDUs that share some discourse relations with the current one. We employed two discourse parsers based on two discourse theories. The Feng-Hirst parser implemented Rhetorical Structure Theory (RST) with the purpose of building a discourse tree for all EDUs in one text; the Lin parser is a PDTB style discourse parser that concerns more local discourse relations. Related work was discussed in Chapter 2, including different discourse theories, two discourse parsers, the general and discourse-related approaches to sentiment analysis. Chapter 3 introduced two of our annotated datasets. The TripAdvisor dataset has annotations of polarity and aspect, and the Multi-Domain-Sentiment dataset has polarity annotation only. In Chapter 4, we explained our representations of discourse relations. We extracted the relations from the discourse parsing results, and represented them in a way that each EDU can be taken as a consequence of a discourse relation with the sentiment of the other argument known. The representations measure the influence of sentiments of other EDUs on the current EDU through discourse relations. We proposed two representation models for RST-based relations: one considered every possible path in the RST parse tree; the other one considered paths whose length are less than 3. One representation model was proposed for PDTB-based relations, since PDTB relations don't constitute a parse tree but rather work locally. We use the parsing results and gold standard annotations to construct the representations. Two sets of baseline features were introduced in this Chapter, including lexicon-based features and features concerning adjacent EDUs with the current one. In Chapter 5 we evaluated our methods on two of our datasets. There is an improvement over baselines on classification accuracy if discourse features were added. Our RST-style representations outperformed our PDTB-style representations.

The performance of our method on the TripAdvisor dataset is not as good as on the Multi-Domain-Sentiment dataset. We attribute this as the size of the TripAdvisor dataset (1535 EDUs) is too small to train a reliable model. In order to make use of our discourse representations, a certain amount of data is required.

From the experimental results we conclude that discourse structure is an useful feature for the task of EDU level sentiment analysis. People might be able to develop a sentiment analysis system that doesn't rely on additional human-annotated lexical resources. One system could start with some text spans whose sentiments are clear, and use the relations between these text spans and other texts to predict unknown or un-understood texts in same document.

For almost every models we implemented (lexicon-based, adjacency-based, RST-based and PDTB-based), the prediction accuracy of *positive* and *negative* instances is better than the accuracy of *neutral* instances. Recognizing neutral text is always hard, even for human beings, since neutral text usually doesn't contain any polarity-bearing words. Our results show that adding context information (say, discourse features and adjacency-based features) helps improve the performance of predicting neutral EDUs. This matches our hypothesis that considering context will help understand the unknown parts of a document.

The fact that our RST based method beats our PDTB method suggests that discourse representation with a higher coverage of EDUs is more suitable for sentiment analysis task, since the PDTB-based representations pay more attention to local discourse relations while RST-based representations capture a global picture for the whole document. By comparing the performances of two models of RST-based representations (one considered every possible path; the other one considered path whose length is less than 3), we can see that adding too many discourse relations doesn't help either. This might be because the higher level discourse relations in a RST tree have relatively weak influences on the changes of sentiments.

6.2 Future work

When we try to predict the sentiment of one EDU, we use sentiments of other connected EDUs or relations. We've taken these sentiments from the gold standard. In the next stage of our work, we would like to replace these sentiments with the predicted values. The working procedure should also be modified. For one document, we should start with some EDUs whose sentiment are more straightforward and explicit, and then predict the EDUs that connect to them. Using the predicted values, we are closer to a realistic system that could be generally applied.

The implicit PDTB discourse relations were removed from the parsing results of the Lin parser in this thesis, since it might to be noisy. We would like to find out a way to deduct the noise and apply these relations since implicit relations are also important to signal the relations between texts.

RST and PDTB discourse features are measured separately. It might also be good if these two sets of features could be merged. We can combine the features to see whether the performance would improve.

References

Carlson, L., Marcu, D., and Okurowski, M. E. (2003). *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer.

Das, S. and Chen, M. (2001). Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*, volume 35, page 43.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Feng, V. W. and Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., and de Jong, F. (2011). Polarity analysis of texts using discourse structure. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1061–1070. ACM.

Hernault, H., Prendinger, H., Ishizuka, M., et al. (2010). Hilda: a discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).

Horrigan, J. A. (2008). Online shopping. *Pew Internet & American Life Project Report*, 36.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Lazaridou, A., Titov, I., and Sporleder, C. (2013). A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *To Appear in Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria. Association for Computational Linguistics.

le Cessie, S. and van Houwelingen, J. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2010). A pdtb-styled end-to-end discourse parser. Technical report, Cambridge Univ Press.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Nielsen, F. Å. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Polanyi, L. and Zaenen, A. (2006). Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications*, pages 1–10. Springer.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *LREC*. Citeseer.

Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., and Webber, B. L. (2007). The penn discourse treebank 2.0 annotation manual.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49. Manchester, UK.

Somasundaran, S., Namata, G., Wiebe, J., and Getoor, L. (2009). Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 170–179. Association for Computational Linguistics.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.

Taboada, M., Brooke, J., and Stede, M. (2009). Genre-based paragraph classification for sentiment analysis. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 62–70. Association for Computational Linguistics.

Taboada, M., Voll, K., and Brooke, J. (2008). Extracting sentiment as a function of discourse structure and topicality. *Simon Fraser Univeristy School of Computing Science Technical Report*.

Tong, R. M. (2001). An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, volume 1, page 6.

Webber, B., Egg, M., and Kordoni, V. (2012). Discourse structure and language technology. *Natural Language Engineering*, 18(4):437–490.

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35. Association for Computational Linguistics.

Wu, F. W. Y. and Qiu, L. (2012). Exploiting discourse relations for sentiment analysis.

Zhou, L., Li, B., Gao, W., Wei, Z., and Wong, K.-F. (2011). Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 162–171. Association for Computational Linguistics.

Zirn, C., Niepert, M., Stuckenschmidt, H., and Strube, M. (2011). Fine-grained sentiment analysis with structural features. In *IJCNLP*, pages 336–344.