Target-Level Sentiment Analysis on Various Genres

Viktor Hangya

Supervisor: Dr. Richárd Farkas

A thesis submitted for the degree of doctor of philosophy of the University of Szeged

University of Szeged
Doctoral School of Computer Science

June 2019

Preface

Textual data conveys a massive amount of information about every aspect of our lives which is easily accessible especially since the advent of social media. *Sentiment analysis* aims to detect the polarity of sentiments about various topics in textual content. Early approaches relied on manually assembled sets of sentiment bearing words by looking at the ratio of positive and negative words in documents. Machine learning made it possible to automatically learn the aspects that convey sentiments by means of annotated data. This dissertation introduces various techniques and a wide range of experiments on both English and Hungarian texts that push the limits of the state-of-the-art in two main areas.

In general, sentiment analysis focuses on the global sentiments in texts but we often express our opinions about certain entities. Various companies or public figures who employ sentiment analyzers are only interested in sentiments towards their actions, which introduces another dimension of analysis since the content referring to the target has to be located first. The dissertation proposes new techniques to the field of *target-level* sentiment analysis in order to locate and understand relevant content better.

For machine learning based sentiment analysis approaches, training data containing documents that are labeled with their sentiment polarity is of utmost importance. Creating such resources is a time consuming task, thus there are only a handful of languages and domains with sufficient amount of annotated data. *Domain adaptation* is an important field of machine learning tackling resource sparseness in various scenarios by exploiting out-of-domain or – in a more extreme case – out-of-language data. The dissertation shows novel techniques which improve sentiment analysis performance in resource-poor setups by relying on easily accessible out-of-domain and foreign language data.

Acknowledgments

First of all, I would like to thank my supervisor, Dr. Richárd Farkas, for his guidance and for supporting my work with his useful comments.

I am indebted to my senior colleagues who showed me interesting undiscovered fields and helped give birth to new ideas during our inspiring discussions. In alphabetical order: Dr. Gábor Berend, Dr. Fabienne Braune, Prof. Dr. Alexander M. Fraser, Prof. Dr. Hinrich Schütze, Dr. István Varga and Dr. Veronika Vincze.

I would also like to thank my colleagues and friends who helped me to realize the results presented here and to enjoy the period of my PhD studies at the University of Szeged. In alphabetical order: István Kádár, György Kalmár, Melinda Katona, Krisztián Koós, Martina Katalin Szabó and Zsolt Szántó.

I am indebted to the organisers of the evaluation campaigns that produced the datasets which enabled me to work on the extremely challenging and interesting problems discussed here. I would also like to thank the anonymous reviewers of my publications for their useful comments and suggestions.

I would like to thank Dr. Veronika Vincze for scrutinizing and correcting this thesis from a linguistic point of view.

I would like to thank my girlfriend Erika for her endless love, support and inspiration. Last, but not least, I wish to thank my parents and my brother for their constant love and support. I would like to dedicate this thesis to them as a way of expressing my gratitude and appreciation.

This work was supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013), the German Academic Exchange Service (DAAD) Research Grants – Short-Term Grants (2016) and the ÚNKP-16-3 New National Excellence Program of the Hungarian Ministry of Human Capacities. I am grateful for this support, which definitely acted as an accelerator for the submission of this thesis.

Contents

Preface **iii**

Acknowledgments **v**

1 Introduction **1**

1.1 Structure of the Dissertation 3

2 Background **5**

2.1 Sentiment Analysis 5

2.2 Machine Learning for Natural Language Processing 9

2.2.1 NLP Techniques 10

2.2.2 Machine Learning Techniques 12

3 Target-Level Sentiment Analysis **17**

3.1 Introduction 17

3.2 Used Datasets 18

3.3 Document-Level System 20

3.3.1 Preprocessing 20

3.3.2 Feature Set 21

3.3.3 Results 23

3.4 Target-Level Feature Engineering 23

3.4.1 Surfaceform-Based Feature Engineering 24

3.4.2 Syntax-Based Feature Engineering 25

3.4.3 Results 27

3.5 Discussion 29

3.5.1 Error analysis 29

3.6 Task-Specific Systems 30

3.6.1 RepLab-2013 31

3.6.2 ABSA-2014 33

3.7 Related Work . . . 35
3.8 Summary of the Thesis . . . 36

4 Fine-Grained Sentiment Analysis **39**
4.1 Introduction . . . 39
4.2 Latent Syntactic Structure-Based System . . . 40
4.2.1 Sentiment Tree Representation . . . 40
4.2.2 Latent Structure Decoder . . . 41
4.2.3 Training Algorithm . . . 42
4.2.4 Tree Based Feature Engineering . . . 42
4.3 Sentence-Level Experiments . . . 43
4.3.1 Datasets . . . 43
4.3.2 Experimental Setup . . . 44
4.3.3 Results . . . 45
4.4 Target-Level Experiments . . . 45
4.4.1 Dataset . . . 46
4.4.2 Latent Sentiment Tree-Based Feature Engineering . . . 46
4.4.3 Results . . . 47
4.5 Discussion . . . 47
4.6 Related Work . . . 49
4.7 Summary of the Thesis . . . 50

5 Sentiment Analysis on Various Genres **53**
5.1 Introduction . . . 53
5.2 Approach . . . 54
5.2.1 System . . . 54
5.2.2 Datasets . . . 55
5.3 Results . . . 56
5.4 Discussion . . . 57
5.4.1 Domain Differences . . . 57
5.4.2 Genre Differences . . . 58
5.4.3 Language Differences . . . 59
5.5 Related Work . . . 60
5.6 Summary of the Thesis . . . 61

6 Domain Specific Sentiment Lexicons **63**
6.1 Introduction . . . 63
6.2 Sentiment Lexicon Creation . . . 64

6.2.1 Lexicon Translation . . . 65
6.2.2 Bootstrapping Sentiment Lexicons . . . 65
6.2.3 Extending Seed Lexicons . . . 66
6.3 Evaluation Setup . . . 68
6.3.1 Datasets . . . 68
6.3.2 Sentiment Classifier . . . 69
6.4 Results . . . 69
6.5 Related Work . . . 71
6.6 Summary of the Thesis . . . 72

7 Cross-Lingual Domain Adaptation for Sentiment Analysis 75
7.1 Introduction . . . 75
7.2 Domain Adaptation Approaches . . . 76
7.2.1 Adaptation of Bilingual Word Embeddings . . . 76
7.2.2 Semi-Supervised Learning . . . 78
7.3 Cross-Lingual Sentiment Classification . . . 80
7.3.1 Datasets . . . 80
7.3.2 Systems . . . 81
7.4 Results . . . 81
7.4.1 Embedding Adaptation . . . 81
7.4.2 Semi-Supervised Method . . . 83
7.5 Medical Bilingual Lexicon Induction . . . 85
7.6 Previous Work . . . 88
7.7 Summary of the Thesis . . . 89

8 Summary 91
8.1 Summary in English . . . 91
8.1.1 Target-Level Sentiment Analysis . . . 91
8.1.2 Fine-Grained Sentiment Analysis . . . 92
8.1.3 Sentiment Analysis on Various Genres . . . 93
8.1.4 Domain Specific Sentiment Lexicons . . . 93
8.1.5 Cross-Lingual Domain Adaptation for Sentiment Analysis . . . 93
8.2 Magyar nyelvű összefoglaló . . . 95
8.2.1 Célorientált szentimentelemzés . . . 95
8.2.2 Aprólékos szentimentelemzés . . . 95
8.2.3 Különböző stílusú szövegek szentimentelemzése . . . 96
8.2.4 Doménspecifikus szentimentlexikonok . . . 96
8.2.5 Keresztnyelvi szentimentelemzés doménadaptációja . . . 97

List of Tables

1.1 The relation between the thesis topics and the corresponding publications. In the results presented in Chapters 3, 5 and 6 the author's contribution was prominent while Chapters 4 and 7 are the results of joint works with other researchers. We state the contributions of the author in each chapter. 4

3.1 Basic statistics about the used corpora. The columns show whether a corpus is annotated on entity- or aspect-level, its language and genre. We have also listed the average number of words in an instance and the number of distinct labels in the corpora. 19
3.2 Label distribution and the overall number of annotated instances in each corpora which were used in our experiments. 19
3.3 Results of three document-level systems for the target-level corpora. 24
3.4 Document-level improvements in terms of F_1 score compared with the unigram baseline for the target-level corpora. 24
3.5 Results of the target-level systems (distance weighted n-grams, syntax-based features and both of them). All three systems were compared with the document-level system (differences are in parentheses). 28
3.6 Target-level (hybrid) improvements in terms of F_1 score compared to the document-level. 28
3.7 Sentiment classification accuracy on the RepLab dataset while gradually enabling the developed techniques. 32
3.8 Results with different numbers of LDA topics on the RepLab dataset. 33
3.9 Official accuracy, reliability, sensitivity and F_1 scores on the test data for each participant. Our system is called SZTE_NLP. 33
3.10 Results of several high ranked participants out of 26. Our system named SZTE-NLP was ranked 6^{th} and 3^{rd} on the restaurant and laptop domains, respectively. 35

4.1 Basic statistics of datasets with sentence-level annotation. 43

4.2 Accuracy scores achieved on the three domains. RNTN is our reference system (Socher et al., 2013), the baseline is a model using word unigrams only and latent refers to the proposed system. 45
4.3 Accuracy scores of the target-level classifier whose feature set is enriched by sentiment-tree based features. We calculated the accuracy using 10-fold cross validation on the *absa-laptop* and *absa-restaurant* databases using the sentiment tree based features. . 47

5.1 Basic statistics about the used corpora. The columns show the corpora's language, genre, the average number of words in the documents and the number of labels in the corpus. 55
5.2 Label distribution and the overall number of annotated documents in each corpora. . 55
5.3 Results of three systems on the used corpora. The accuracy (acc) scores and macro-averaged F_1 scores were calculated using 10-fold cross-validation. Differences among systems can be seen in parentheses. The unigram baseline was compared with the most frequent class (MFC) system, while the document-level was compared with the unigram baseline. 57
5.4 F_1 score improvements of the document-level system compared with the unigram baseline. 57

6.1 An example sentence and the features extracted from it. The sentiment word in the sentence is *better*, which has 3.5 polarity value. 69
6.2 Extension of seed lexicon with wordnet (wn) and cluster based methods. The accuracies on OpinHu and ProdRev corpora were measured using 10-fold cross-validation. 70
6.3 Achieved accuracies using different lexicons on OpinHu and ProdRev. Data used for training baseline systems, initializing seed or pmi lexicons are indicated in the first column. 70

7.1 Accuracy of the BWE adaptation approach on the target-level sentiment classification task. The oracle systems used Spanish sentiment training data instead of English. . 82
7.2 Accuracy on CLSC of the adapted BWE approach with the semisup (target-ignorant with additional loss functions) system compared to the target-ignorant in brackets. . 83
7.3 Accuracy on CLSC of both target-aware and target-ignorant systems using English or/and Spanish labeled sentiment training data. Column *lang* shows the language of the used training data. Differences compared to semisup are indicated in brackets where available. 84
7.4 Results (F_1) for medical BLI with the cosine similarity and the classifier based systems. We present baseline and our proposed domain adaptation method using both general (BNC) and medical lexicons. 86

7.5 Results with the semi-supervised system for BLI. Differences compared to previous results are indicated in brackets. Baseline results are compared to rerun experiments of Heyman et al. (2017) using BWEs instead of MWEs. We use the medical labeled lexicon in all cases. 87

List of Figures

2.1 Example output of fine-grained sentiment analysis. The figure demonstrates how sentiment is propagated through the syntax of the sentence. 7

2.2 Sample constituency (left) and dependency (right) parse trees. 12

3.1 Equation for distance-based weighting (left), where n is the length of the input text, i and j are the positions of the actual word and the mention of the target. An example sentence where *Metallica* is the target can be seen on the right. 25

3.2 Dependency parse tree (Bohnet parser), where *computer* is the target. Adjectives that are in close proximity are indicated in bold and also the relations whose head word is in the sentiment lexicon. 26

3.3 Constituency parse tree (Stanford parser), with dotted and dashed subtrees for the *food* and *service* targets, respectively. 27

3.4 Ablation study showing accuracy scores on the restaurant (left) and laptop (right) test data. 34

4.1 Representation of sentiment trees in the Stanford Sentiment Treebank (Socher et al., 2013) (left) contains 5-level sentiment annotation {`0=very negative`, `4=very positive`} for each node of the binary syntactic tree. On the other hand, we assume that we have access only to sentence-level polarity annotation, i.e. only the label of the root is given (right). Here, the states of the inner nodes are described by latent discrete variables {`A,B,C`}. 41

4.2 The highlighted subtree with latent labels {`A, B, C`} is the subject of local feature extraction for node `A`. 42

4.3 Average accuracy improvements in percentage points of the latent system over the baseline system on the restaurant test dataset depending on sentence length. 48

7.1 Domain adaptation of BWEs. Red, gray and blue word clusters correspond to words with positive, neutral and negative sentiment, respectively. The polarity of word *cool* is not clear because it depends on the target domain. *OMG* is a domain specific word and was not seen in general domains. 77
7.2 Walking cycles of the semi-supervised approach. Red and blue nodes illustrate labeled samples with 2 possible labels while gray depicts unlabeled ones. Green and red arrows show correct and incorrect cycles, respectively while opacity shows step probabilities. 78

Chapter 1

Introduction

Due to the development of Web 2.0 a large amount of user generated data is produced daily. A significant portion of this data is in textual form which contains the opinions and sentiments of the authors. These contents are very useful to gather insights on the general opinion about various entities thus organizations started to exploit them. Several studies have been carried out in the area, e.g. in monitoring brands (Jansen et al., 2009), predicting the results of elections (Sang and Bos, 2012) and disaster management (Varga et al., 2013). Because of the large number of these textual contents, manual processing is not applicable, therefore automatic analysis is needed.

The general approach to sentiment analysis is to decide the overall sentiment polarity of a given sentence which could be positive, negative or neutral in most cases. When writing product reviews, people usually share their opinions about various properties of the given product instead of writing just one global feeling about it. Furthermore, people often compare multiple entities in one sentence, which means that the very same sentiment expression can be positive for one entity while it is negative for another one. Thus, *document-level* analysis is not sufficient in these cases since it is important to be aware of the expressed sentiments' targets. The aim of *target-level* sentiment analysis is – given a pair of text and target entity – to classify the polarity of the expressed sentiments related to that target.

One contribution of this dissertation is the introduction of novel techniques for the task of target-level sentiment analysis. We develop new features which increase the performance of supervised classifiers. In order to detect the correct sentiment polarity referring to a given target, it is important emphasize those parts of texts which are related to the target. For this, we exploit both shallow and deep syntactic structures of sentences to produce indicative features for the task.

Another aspect of the analysis is its granularity. Most of the sentiment classification approaches predict one polarity label for a sentence or document independent of performing document- or target-level analysis. The next step in terms of granularity is to apply the analysis on the level of phrases, i.e. to predict a sentiment score for each word, phrase, sub-sentence in a given sentence and

for the whole sentence as well. The benefit of such an analysis is twofold. First, on the sentence level, the system can profit from exploiting the sentiment scores of phrases, as it can analyze the discourse relation between these phrases, e.g. they can be contrastive or one can intensify the other. Second, fine-grained analysis can be used for other tasks, e.g. target-level sentiment analysis. The problem of fine-grained sentiment analysis systems is that a sentence level annotation is insufficient while the more detailed ones are expensive to create. We contribute to this field by introducing a latent syntactic structure based system which can predict fine-grained labels using sentence-level annotated data only. We show that the output of this system is useful for target-level sentiment analysis, e.g. using only the labels of sub-sentences which are related to the target in question.

An important aspect of sentiment analysis is the difference among various text domains. Texts from social media come from various genres and languages, which makes the creation of one unified sentiment analysis system nearly impossible because text styles can be very different from each other. For example, considering texts coming only from social media platforms, blogs and news posts have the most standard language, they are the most correct in terms of spelling and usually there is no length limitation thus topics can be described in more detail. On the other hand, micro-blogs like Twitter have length limitations, which makes the texts very concise and it contains a lot of slang words, as well as abbreviations. In addition, because they are used for quicker communication, posts can contain many misspellings.

Due to the differences of various text domains, it is hard to develop a general sentiment classifier system that performs well in each case. We show main differences of various domains on the sentence level and develop different sets of features to show their applicability and effect on the performance of sentiment classifier systems in these domains. We perform a detailed analysis on the hard cases that a given technique can solve.

We contribute to this area by developing techniques for building domain specific sentiment lexicons, which are useful external language resources for improving the performance of sentiment analysis systems. They contain words with their sentiment polarities and can be used for adding external knowledge to the feature extraction phase. As mentioned, text domains can differ to a large extent which means that a given expression can convey different sentiment in different domains. For example, the word *loud* has a positive meaning when reviewing various types of speakers while it is negative in the kitchen appliances domain. Most of the publicly available sentiment lexicons are for general use thus they do not have domain specific knowledge. We show that using a general purpose lexicon can even hurt performance when using in a specific domain. We propose to automatically create and adapt sentiment lexicons in order to improve performance on a given domain. We use different word similarity signals based on labeled and unlabeled data to transfer information from general to a specific domain.

In addition, we propose a technique for domain adaption in cross-lingual setups. For many languages there is an insufficient amount of labeled data to build good quality sentiment classifiers.

Word embeddings got a lot of attention in the neural network era because semantic information in unlabeled corpora can be exploited and the performance of systems relying on them can be improved. By representing words from different languages in a shared vector space, bilingual word embeddings can bridge the gap between languages to some extent. By relying on these resources, cross-lingual transfer learning made it possible to build models using annotations from a resource rich language and apply it to a resource poor one. Although these methods work well in many scenarios, they often have low quality when the training and test data are from different domains. To overcome this issue, we perform domain adaptation of bilingual sentiment classification without the need of additional annotated data.

1.1 Structure of the Dissertation

This dissertation comprises two main topics. The first deals with target-level sentiment analysis while the second tackles the problem of handling genre, domain and language differences. In this section we outline the structure of the dissertation and present the publications of the author.

In Chapter 2, we give an introduction and motivation to sentiment analysis. We position our work in the related literature and provide the necessary definitions for the thesis.

Chapter 3 introduces the problems of target-level sentiment analysis. We propose novel techniques for the task based on both the surface form and the syntax of sentences, which helps to detect target related information. Results are presented on various English and Hungarian datasets including ones released for shared-tasks. It is shown that the introduced techniques reached state-of-the-art performance, moreover, the developed shared-task specific systems performed among the best official participants.

Most sentiment analysis systems work on document or sentence level. Chapter 4 deals with a more fine-grained analysis, i.e. detecting the sentiments of each constituent of a sentence. For this a semi-supervised technique was developed which handles the sentiment of constituents as latent variables. For our approach only sentence-level signal is needed for training as opposed to previous work, which makes it broadly applicable. We show that the system improves performance for both sentence- and target-level analysis by extracting sentiment indicators for the fine-grained output in the case of the latter one.

The second topic of this dissertation is introduced in Chapter 5. Genre, domain and language differences are a serious issue for sentiment analysis. In this chapter a detailed comparison of these differences is shown as the foundation of the last two chapters.

Chapter 6 focuses on the shortcomings of sentiment lexicons over various domains. The same expression can often have opposite sentiment polarity meanings in different domains. To overcome this problem, adaptation methods are introduced using labeled and unlabeled lexical resources. Results are shown on Hungarian datasets.

Finally, Chapter 7 addresses the problems of cross-lingual sentiment analysis related to domain differences. We introduce a semi-supervised method for domain adaptation which relies on no additional labeled data. We show that both semantic knowledge from general domain data and domain specific information are needed to achieve good performance in bilingual transfer learning.

Table 1.1 summarizes the relationship among the thesis chapters and the key referred publications.

			Chapters				
			3	4	5	6	7
CLEF	2013	(Hangya and Farkas, 2013a)	•				
CogInfoCom	2013	(Hangya and Farkas, 2013b)	•				
ICCIA	2017	(Hangya et al., 2017)		•			
SEMEVAL	2013	(Hangya et al., 2013)			•		
AIRE	2017	(Hangya and Farkas, 2017)	•		•		
TSD	2015	(Hangya, 2015)				•	
ACL	2018	(Hangya et al., 2018)					•

Table 1.1: The relation between the thesis topics and the corresponding publications. In the results presented in Chapters 3, 5 and 6 the author's contribution was prominent while Chapters 4 and 7 are the results of joint works with other researchers. We state the contributions of the author in each chapter.

Chapter 2

Background

In this chapter we provide the background to understand key concepts in the dissertation. First we give a general overview of sentiment analysis and position our work in the field. Next, we define basic common notations and definitions for Machine Learning in Natural Language Processing. We gradually introduce datasets used for our experiments in later chapters.

2.1 Sentiment Analysis

The task of sentiment analysis is to decide the sentiment polarity of texts which could be positive, negative or neutral in most cases. Recently, the popularity of social media has increased. People post messages on a variety of topics, like products and political issues thus a large amount of user generated data is created in textual form. Owing to its direct marketing applications, sentiment analysis has become an active area of research (Liu, 2012). O'Connor et al. (2010) showed that public opinion correlates with sentiments extracted from texts. Sentiment analysis using social media can capture large scale trends using the large amount of data generated by people. In (Jansen et al., 2009) consumer opinions from micro-blogs concerning various brands were investigated. It was shown that 19% of micro-blog messages contain the mention of a brand and 20% of these contain sentiments related to the brand. Just to name a few use cases, monitoring these sentiments allows companies to gain insights into the positive and negative aspects of their products. Furthermore, the analysis of micro-blogs permits political parties to manage their campaign in a better way. For instance, Sang and Bos (2012) used Twitter messages to predict the outcome of Dutch elections. It was shown that by counting tweets that mention political parties is not sufficient to obtain good predictions. By mining sentiments related to the parties, the results became nearly as good as traditionally obtained opinion polls but it is faster and also cheaper. In 2011 after the great Eastern Japan earthquake, people used Twitter to report problems and aid messages. The authors of (Varga et al., 2013) created a system for disaster management using these reports. Their method was

based on matching the problem reports and aid messages with each other. There are numerous publications about this topic and its applications. An honorable mention is (Liu, 2012), which gives a broad overview of the task. More formally, the problem of sentiment analysis is a supervised classification task in the field of natural language processing, which we introduce in Section 2.2 in more detail.

Level of Analysis

Sentiment analysis can be applied at different levels depending on the depth of information which we want to extract from the texts. The task of the most general case is to decide whether a given document or sentence contains sentiments on a global level and determine its polarity. Consider the following example:

Um I just learned that Sunday is NATIONAL CHOCOLATE DAY. I'm totally taking advantage of that. (2.1)

The above text expresses the idea that the author has a positive sentiment because of the forthcoming *chocolate day*. To detect this sentiment, all the sentences in the document have to be analyzed because it expresses the positive sentiment as a whole. This level is called **document-** or **sentence-level** sentiment analysis depending on the type of input.

Target-level analysis performs a more focused opinion extraction (Jiang et al., 2011). In this case, the output of the system is a combination of a polarity label and a referred target. The target can be an entity (person, product, service, etc.) or some aspect (battery life, quality of a service, etc.) of a given entity. In some cases this level is called *entity-* or *aspect-level* sentiment analysis depending on the type of the target. In the following examples both types of targets can be seen:

I do agree that money can't buy happiness. But somehow, it's more comfortable to sit and cry in a ***BMW*** *than on a bicycle!* (2.2)

The ***menu*** *is limited but almost all of the* ***dishes*** *are excellent.* (2.3)

In the first example, the target entity is *BMW*, but the negative sentiment is not related to it. Because there is no sentiment towards *BMW*, the polarity label for this example is neutral. In the second example, the target aspects are the *menu* and *dishes*, which are aspects of a restaurant. The sentence contains sentiments related to both aspects, one with negative polarity and one with positive. Target-level analysis can be divided into two major steps: target detection and sentiment classification. Since in many real-world scenarios targets of interest are known a priori, i.e. a company is only interested in its own products, we focus only on the second step in this dissertation assuming given document-target pairs as inputs. The difficulty comparing to document-level is that textual parts have to be detected which are related to the given target and only those have to be analyzed in order to classify the correct sentiment. We introduce novel feature-sets based on both shallow

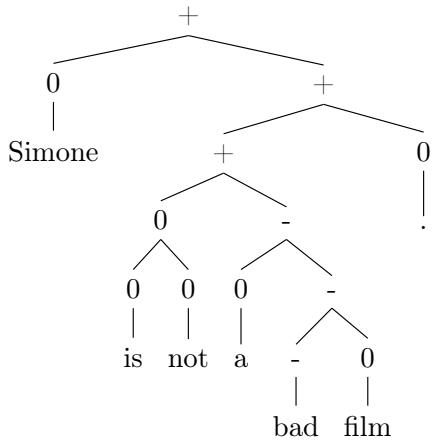

Figure 2.1: Example output of fine-grained sentiment analysis. The figure demonstrates how sentiment is propagated through the syntax of the sentence.

and deep sentence structures to emphasize related parts and to improve performance.

An orthogonal level, comparing to those mentioned above, is **fine-grained** sentiment analysis, where the ultimate goal is to improve performance in general by deciding the sentiment polarity on the phrase level. By exploiting the relations between phrases we can have a deeper understanding how sentiment is propagated across a given sentence, which leads to better performance. An example output of fine-grained analysis is shown in Figure 2.1, which illustrates how negation affects the negative phrase *a bad film* leading to positive overall sentence polarity. Such an analysis is beneficial not only on sentence level but on target level as well by allowing the analysis of only those phrases which are related to the target in question. A disadvantage of this level is that corpora with phrase-level sentiment annotation are needed to train such models, which is time-consuming thus expensive to create. We contribute to this field by proposing a technique incorporating latent syntactic structure, which can overcome this issue by relying only on sentence-level annotations.

Domains and Genres

Sentiment classification is highly sensitive to the domain from which the data is extracted (Liu, 2010). A classifier trained using documents from one domain often performs poorly on data from another domain. The reason is that words and even language constructions used in different domains for expressing sentiments can be different. However, out-of-domain resources have often to be used when in-domain data are not available.

One important source for getting insights about people's sentiments are forum posts and product reviews sites. Texts from these sources are mostly well structured and correct in terms of spelling, which is due to the fact that authors have more time to write coherent posts comparing to other

social media genres. Because of this fact, NLP tools, e.g. syntactic or semantic parsers, can perform well on such texts making it easy to exploit them in end-tasks like sentiment analysis. However, this genre covers many domains where different phrases could be used for expressing a certain sentiment. For example, while the expression *very loud sound* is a positive aspect in the electronic devices domain when talking about a speaker, it is negative in the kitchen appliances domain. In contrast with forum posts and product reviews, micro-blogs posts like tweets are created almost in real-time so their form is less standard and they contain many more spelling errors, slang words and other out of vocabulary words. This causes many problems when analyzing them, usually a more intensive preprocessing is needed but syntactic and semantic parsers trained on standard texts still do not perform well on them due to the lack of training data in such domains (Foster et al., 2011).

In addition, languages also have many differences in terms of how things are expressed, beyond their vocabulary differences, thus systems and feature-sets engineered to one language could not be applicable to another after retraining on the target language. In the dissertation we aim to identify sentiments in English texts, as well as Hungarian, the latter being a free word order morphologically rich language. It has several word forms due to inflections, which may mean more out-of-vocabulary and rare words. A word's syntactic role is defined by its morphology, unlike English, where word order is determinative.

In this dissertation we highlight domain and language differences and compare domain specific techniques for sentiment analysis. To overcome the issues of resource poor domains, data from resource rich domains are incorporated by domain adaptation techniques (Ben-David et al., 2010). We show two domain adaptation scenarios for sentiment analysis.

Shared Tasks and Approaches

In recent years shared tasks have been organized to promote research in sentiment analysis for social media. The goal of *SemEval – Sentiment Analysis in Twitter* tasks, organized from 2013 to 2017 (Wilson et al., 2013; Rosenthal et al., 2014, 2015; Nakov et al., 2016; Rosenthal et al., 2017), was to classify a given Twitter message into positive, negative or neutral classes. In the contribution of (Hangya et al., 2013) it was shown that Twitter specific normalization of texts like URL and emoticon unification is a significant step before classification. Most of the participating systems were based on supervised machine learning techniques. The features used included word-based, syntactic and Twitter specific ones such as abbreviations and emoticon handling. The systems heavily relied on word polarity lexicons like MPQA (Wilson et al., 2005), SentiWordNet (Baccianella et al., 2010) or lexicons designed specifically for this task (Zhu et al., 2014). We discuss these techniques in the following section.

The focus of *RepLab-2013 – An evaluation campaign for Online Reputation Management Systems* (Amigó et al., 2013) shared task was on target-level sentiment analysis, more specifically on entity-level. The participants' task was to perform online reputation monitoring of particular entities using

Twitter messages. The best performing systems tried to capture the contexts which are related to the given entity. In (Cossu et al., 2013) Continuous Context Models were used, which tend to capture and model the positional and lexical dependencies existing between a given word and its context. In the system which achieved the best results (Hangya and Farkas, 2013a), besides various features, distance weighting was used to model the context of a given entity. An SVM-based classifier was proposed in (Jiang et al., 2011) which incorporated target-dependent features relying on the syntactic parse tree of the sentences. Since tweets are usually short texts, the entire conversation, i.e. the tweet which has to be classified and all replies to it, were taken into consideration in the classification phase. Another shared task series which focused on target-level sentiment analysis is *SemEval – Aspect Based Sentiment Analysis* organized between 2014 and 2016 (Pontiki et al., 2014, 2015, 2016). The goal of this task was to identify the aspects of given target entities (Poria et al., 2014; Li et al., 2015) and the sentiment expressed for each aspect. Data provided by the organizers consist of restaurant and laptop reviews. The best performing systems (Wagner et al., 2014; Kiritchenko et al., 2014) were based on SVM classifiers. The features used were the following: n-grams; target-dependent features (using parse trees, window of n words surrounding the aspect term); polarity lexicon based features. It was shown that the most useful features were those derived from the lexicons. Other systems exploited constituency parse trees by selecting constituents which are related to the aspect in question (Hangya et al., 2014). The performance of the introduced techniques in this dissertation are evaluated on such shared tasks' data.

In addition to traditional feature engineering based techniques, the recent advancement in the field of neural networks made it possible to build neural systems for sentiment classification achieving great performance. The use of word embeddings enabled systems to exploit semantic similarity of words based on large unlabeled corpora (Mikolov et al., 2013a). Based on embeddings various neural systems were proposed, such as the convolutional neural network of Kim (2014) for sentence-level analysis or the recurrent neural network of Zhang et al. (2016) for target-level classification. We propose neural techniques as well relying on both systems in Chapter 7.

2.2 Machine Learning for Natural Language Processing

Natural language processing is the joint area of computer science and linguistics with the goal to help the interaction between humanity and computers (Jurafsky and Martin, 2009). The field includes various tasks including text and speech understanding and generation. In the research field, text and speech processing are two major directions which are usually well separated. In this work we focus on the former only. In the early stages of the field, rule-based systems were used due to the lack of computational power. The improvement of computers allowed the field to use statistical approaches such as various artificial intelligence and machine learning algorithms. These approaches are able to infer task specific decision rules during training based on statistics relying on textual corpora,

i.e. from the training data. In the following we introduce machine learning as well as text processing techniques that are used throughout the dissertation.

2.2.1 NLP Techniques

Each NLP application can be imagined as a pipeline where each step is responsible for performing a given task on the input. In general, these pipelines involve the preprocessing of texts, transforming it to an algorithmically manageable representation and finally applying various algorithms which enrich the input with additional information, such as syntactic parses or sentiment labels. The possibilities for the last step are nearly endless, thus we only mention syntactic parsing in this chapter due to its use in our approaches.

Preprocessing

The input of NLP pipelines are raw strings which need to be cleaned and converted to textual units. This step involves the following sub-steps which can vary depending on the downstream applications (Korde and Mahender, 2012):

- Tokenization: Most NLP applications cannot deal with long raw strings, thus they need to be split to smaller units. In case the input contains complete articles, they have to be split into paragraphs and sentences. This can usually be done deterministically by splitting paragraphs at empty lines and sentences by punctuation. Since issues could occur with these simple approaches, e.g. splitting text at abbreviations such as *Mr.* makes an error, more advanced tools were developed based on rules or statistics (Manning et al., 2014). Sentences are further split into tokens, such as words or sub-word units. Similarly as before, it can be done either by splitting words by whitespace characters but more advanced methods can be used as well which are aware of various language phenomena (Manning et al., 2014).

- Stopword removal: Many words, such as *the, a, and*, etc. occur frequently in texts but do not carry useful information for some tasks. For example, the words mentioned above do not carry any sentiment meaning. On the other hand, stopwords often cause noise. These words are often removed using a list of pre-defined words or patterns which are dependent on the downstream task.

- Normalization: Words can be written in various forms, i.e. with various inflections, upper or lower cased, etc. In many cases these forms do not contain useful information for a given task. On the other hand, it increases data sparseness, i.e. some tokens could be infrequent in the training data, which could lead to the decrease of performance. To overcome these issues, various task dependent steps can be applied, e.g. lowercasing words or unifying tokens ($www.google.com \rightarrow [URL]$). To remove inflections of words two main approaches exists:

stemming and lemmatization. Both of them aim at finding the root of a given word. The former is often based on rules and can produce word stems that are invalid words (Porter, 1980), while the latter involves using lookup-tables or morphological analysis and results more accurate word roots, i.e. lemmas (Manning et al., 2014).

Text Representation

After having the raw input texts cleaned and tokenized into text pieces, they have to be converted into a representation which can be handled easily by NLP applications. Various such representations are possible (Jurafsky and Martin, 2000, Chapter 4). Here we present two which we use in later chapters.

Bag of Words A common approach in text classification is to represent documents as a bag of their words (tokens). More precisely, we use a vector with dimensions of the size of our vocabulary, where each index indicates how many times the corresponding word is present in the document. This way the word order is disregarded and only their frequency is kept. In addition to words, the presence of higher level features can also be indicated in the model, thus often referred to as bag-of-features model. For more detail see Section 2.2.2.

Vector Semantics In the case of the previous approach, words are represented with their indices in the vocabulary. The disadvantage of this is that indices do not convey any information about the meaning of words. In real life, words are not orthogonal but some are more similar than other, e.g. *dog* is more similar to *cat* than *apple* or both *good* and *bad* can be used to describe quality but with an opposite meaning. To represent the meaning of words and to measure their similarity, vector semantics was proposed (Joos, 1950), where the idea is to represent a word as a point in some multidimensional semantic space. Vectors for representing words are often called embeddings. Many approaches were developed each based on the idea that words having similar meanings occur in similar contexts. Among others, *word2vec* (Mikolov et al., 2013a) is a popular approach to build word embeddings using neural networks where the idea is to predict a given word based on its contexts (*skipgram*) or vice-versa (*cbow*). Latent Dirichlet Allocation, which assigns topic distributions to words and documents (Blei et al., 2003), or the Brown clustering algorithm, which is a hierarchical word clustering algorithm based on co-occurrence statistics (Brown et al., 1992), can also be used to acquire vector representations. Finally, in order to represent a whole document using word embeddings, a simple approach is to perform an element-wise averaging of vectors corresponding to the words in the given document. The resulting vector represents the meaning of the whole document. Another approach mostly used with neural networks is to concatenate word embeddings and using the resulting matrix as the input.

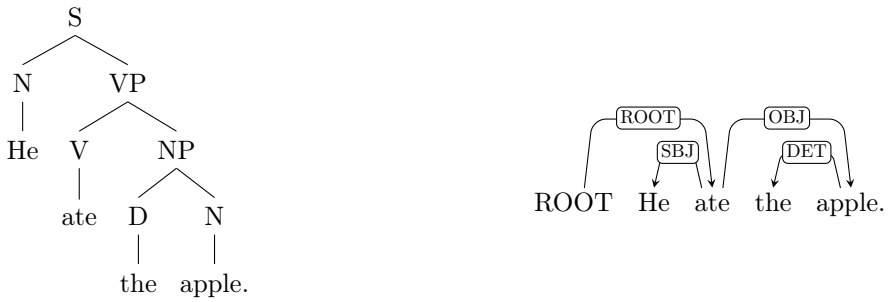

Figure 2.2: Sample constituency (left) and dependency (right) parse trees.

Syntactic Parsing

After converting input to manageable units, various information can be extracted from them. These information could be the final answer which we are looking for, such as sentiment classes as in this dissertation, or they could further be used by other applications. Since some of our proposed methods rely on the syntax of sentences we describe syntactic parsing as a possible application.

Syntax is meant to show how words group together and relate to each other as heads and dependents. Syntactic parsing is the task of assigning a syntactic structure to a given sentence (Manning and Schütze, 1999, Chapter 12). This structure is represented by an ordered tree called *parse tree*, derivation tree or syntax tree. Two main approaches exist for the generation of parse trees: constituency and dependency parsing: sample parses can be seen in Figure 2.2. The aim of the former, also known as phrase structure parsing, is to break sentences into a hierarchy of phrases. Each non-terminal node (inner node) of the parse tree represent a phrase which is also called constituent, while terminal nodes (leaves) are the words in the sentence. Furthermore, non-terminal nodes are labeled with the category of the represented phrase or the POS tag of the represented word in the case of pre-terminal nodes. In contrast, dependency parsing connects words directly according to their relationship. The child node of a directed edge is dependent on the parent while the edge label indicates the relationship category. The *root* node and edge show the main verb in the sentence. In order to train such parsers, a treebank is required which contains sentences along with their manually annotated syntactic trees, such as the Penn Treebank for English (Taylor et al., 2003).

2.2.2 Machine Learning Techniques

In the following we present the machine learning techniques, including classification approaches and others, related to the work done in this dissertation.

Classification Algorithms

The task of sentiment analysis is a supervised machine learning problem where the input text units have to be assigned to one of the pre-defined sentiment polarity classes. Formally, the task of supervised classification is to take an input d and a fixed set of output classes $C = \{c_1, c_2, ..., c_n\}$ and return a predicted class $\hat{y} \in C$. To train a supervised classifier we use a training set of N documents that have each been labeled with a class: $T = \{(d_1, y_1), ...(d_N, y_N)\}$. Using the trained model we can assign a class $\hat{y} \in C$ to an unseen document d. Various algorithms were proposed which differ in multiple aspects, e.g. the representation of documents or the objective function.

A well proven approach for document classification is to represent text samples as bag of words and features and apply one of the many classifier algorithms. The performance of a given approach is highly dependent on the used features. The basic features which are nearly always used for NLP tasks are word n-grams, i.e. words, word pairs, word triples etc. occurring in the given text. In addition, higher level feature templates, which vary from task to task, are often developed to boost the performance of systems. These feature templates can take various forms, such as binary values indicating the match of a given regular expression pattern, number of occurrences of entities based on external knowledge bases or syntactic parse tree based word relations. Our work on which the dissertation is based involves a considerable amount of such feature template engineering which we introduce in later chapters.

Various supervised classification algorithms can be applied to the representation of documents, such as the Naïve Bayes classifier, support vector machines (SVMs) or the maximum entropy classifier (Bishop, 2006). We mostly use the latter in our work based on our previous experiences. The maximum entropy classifier, also known as multinomial logistic regression, learns a linear predictor function that calculates a score using a set of weights and the representation of a given observation using their dot product. During training separate sets of weights are learned for each possible label while maximizing the log-likelihood of the data. The prediction is made by selecting the label associated with the highest score based on its weights and the sample representation.

Neural Networks Neural network based approaches have received a lot of attention recently (Goldberg, 2016). The power of neural networks lies in its non-linearity, which makes them able to learn high-level features without the need of manual feature engineering. Another big conceptual jump when moving from linear models to neural network based models is to stop representing each feature as an index which is independent from others and representing them instead as dense vectors. These embeddings can either be learned during the training process using labeled samples or they can be pre-trained using techniques mentioned earlier on large amounts of unlabeled text (Collobert et al., 2011). Furthermore, various techniques were developed, often to mimic the way how humans approach different tasks, such as convolutional, recurrent or recursive neural networks, each excelling at different tasks. Although neural networks have the ability of learning high level

feature representations, a large amount of data is required to do so. For that reason, linear classifiers with well designed feature templates often perform better when a large amount of annotated data is lacking, which is often the case for sentiment analysis.

Evaluation

In order to measure the quality of a system's output and to compare different systems, we need to evaluate them on a given labeled test set. The two most common metrics used for sentiment classification is *accuracy* and the precision and recall based F_1 score (Jurafsky and Martin, 2009, Chapter 4).

Accuracy is a measure calculating the percentage of the correctly predicted samples in the test set. More formally it is calculated by the following formula:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\delta(\hat{y}_i, y_i) \tag{2.1}$$

where n is the number of samples in the test set, $\hat{y}_i$ and y_i are the predicted and true labels of sample i and $\delta(\cdot,\cdot)$ is 1 if the two arguments are equal and 0 otherwise. Although accuracy is often used for classification tasks, it has a disadvantage when using in cases when the label distribution is skewed in the test set. More precisely, a system predicting the same label independent of the samples may achieve high accuracy in an unbalanced scenario.

The label-wise precision and recall measures aim to overcome the issue of unbalanced label distribution. Precision measures the percentage of samples that were correctly predicted as c out of all samples predicted as c (precision is considered 1.0 if none of the samples are predicted as c):

$$Precision_c = \frac{1}{\sum_{i=1}^{n}\delta(c, \hat{y}_i)}\sum_{i=1}^{n}\delta(c, y_i)\cdot\delta(c, \hat{y}_i) \tag{2.2}$$

In contrast, recall measures the percentage of samples that were correctly predicted as c out of all true c samples (recall is considered 1.0 if none of the samples have c as true label):

$$Recall_c = \frac{1}{\sum_{i=1}^{n}\delta(c, y_i)}\sum_{i=1}^{n}\delta(c, y_i)\cdot\delta(c, \hat{y}_i) \tag{2.3}$$

In order to aggregate the two measures into one single value their harmonic mean called the F_1 measure is used most commonly:

$$F_{1,c} = 2\cdot\frac{Precision_c \cdot Recall_c}{Precision_c + Recall_c} \tag{2.4}$$

In addition, to aggregate the label-wise measures to global precision, recall and F_1 scores, their average is calculated, called macro averaging. We show results in percentage points throughout the dissertation.

Structured Prediction

The vast majority of machine learning algorithms are built to solve prediction problems whose outputs are simple values, e.g. classification or regression. In contrast, structured prediction deals with tasks which require a complex output (Smith, 2011). Typical structures could be sequences like in the case of POS-tagging, named entity recognition or machine translation, or parse trees like in the case of the already mentioned syntactic parsing task. Solving structured prediction involves making individual sequential decisions while each decision depends on all previous ones.

A well performing approach for structured prediction, which we also employ in our work, is the structured perceptron algorithm (Collins, 2002). The method combines the perceptron classifier (Bishop, 2006, Chapter 4) with the Viterbi inference algorithm (Jurafsky and Martin, 2009, Appendix A). The algorithm maps the possible structures of samples to features using a feature function which are scored using the parameters of the perceptron classifier just as in the case of a classification task. On the other hand, the number of possible structures of a given sample is huge thus an exhaustive search of structures is not possible. The task of finding the most probable structures of a sample is called decoding. The Viterbi algorithm (Forney, 1973), which is a dynamic programming method, uses the model parameters to decode the most probable structure (or the n most probable ones with the beam search algorithm) of a given sample, which are then scored using the feature function. The highest scoring structure is selected as prediction which is also used for the perceptron update rule upon misclassification.

Domain Adaptation

Most machine learning algorithms assume that models are trained and tested using data drawn from some fixed feature space and distribution, i.e. domain (Ben-David et al., 2010). On the other hand, in many real life scenarios there are not enough data or no data at all to train a model in a given domain. For such cases out-of-domain data are used for training and the learned model is applied in-domain. Due to the different distribution of data coming from these domains, the performance of the model drops. Three main scenarios are considered: *unsupervised domain adaptation* when labeled and unlabeled samples are available on the source domain but only unlabeled ones on the target; *semi-supervised domain adaptation* when a very small portion of the target samples are also labeled; and *supervised domain adaptation* when both the source and target side samples are labeled but the target domain data is too small to train an adequate model. Various methods were proposed for domain adaptation. In the case of *bootstrapping approaches* a model trained on the source data is used to predict the label of target samples, which are then used in the next iteration to retrain the model (Wu et al., 2009). In the case of a small amount of annotated target samples, *weighting* the importance of source data based on the similarity of their distribution to the target domain is a well performing approach (Jiang and Zhai, 2007). Another exciting approach is to separate features of samples that have similar distribution over the two domains from those that do not and use them

in the case of both source and target training samples or only for one domain (Daume, 2007). We perform domain adaptation experiments in Chapters 6 and 7.

Cross-Lingual Transfer Learning Domain adaptation is closely related to the field of transfer learning (Pan et al., 2010). Transfer learning is a general term that refers to a class of machine learning problems that involve not only transferring knowledge between different domains but often between different tasks as well. In the literature, a standard definition of transfer learning cannot be found yet and it is often used interchangeable with domain adaptation (Li, 2012). In the case of cross-lingual transfer learning the task is to transfer knowledge from one language to the other. For many resource-poor languages, annotated resources could be lacking for a given task, independent of the domain, thus the possibility of exploiting resources in resource-rich languages is of great interest. The task could also be considered as an extreme case of domain adaptation where the source and target domains differ largely both in their feature representation and distribution. A simple approach to solve this problem is to employ machine translation systems in order to translate either the source language training data to the target language or to translate the test samples (Wan, 2009). On the other hand, machine translation could introduce noise to the data, which decreases performance. In addition, acquiring parallel data to train a translation system for a resource poor language could be difficult as well. Bilingual word embeddings (Ruder, 2017), i.e. embedding words from the source and target languages to a joint semantic vectors space, are cheap resources which can boost the performance of cross-lingual transfer learning approaches by bridging the feature representation gap between languages. With the representation of source and target language texts in a shared feature space, it is possible to train a model on the source data and simply apply it to the target samples (Upadhyay et al., 2016). Since such embeddings can be built by using monolingual data for both languages and a small dictionary (Mikolov et al., 2013b), they are easily applicable in low-resource setups. We perform cross-lingual sentiment experiments using bilingual word embeddings in Chapter 7.

Chapter 3

Target-Level Sentiment Analysis

3.1 Introduction

As described in the previous chapter, detecting the polarity of sentiments in texts is important for various reasons. In many cases our interest is only to get insights about a certain topic or target entity while ignoring sentiments expressed about other entities. Texts usually contain multiple sentiments referring to different targets, e.g. reviewing the aspects of an entity, such as the food quality of a restaurant, or comparing multiple entities against each other. In these cases, we are not interested in the global sentiment of the given text, which is the overall feeling of the author, thus document-level systems ignoring the target of interest are not appropriate for this task. In this chapter we describe the task of target-level sentiment analysis and show methods that make a system target aware. We show the results of our proposed methods on multiple datasets.

When dealing with target-level sentiment analysis, one can face two subtasks: *target detection* and *classification of sentiments* that are related to the targets. The task of the first one is to identify the targets in sentences that can be referred to and can hold sentiments while the second deals with the actual detection of target related parts of a sentence and classifying their sentiments. In many cases the first step can be skipped because entities to which target-level analysis is to be applied can consist of a predefined set of targets, e.g. a company could be interested in sentiments regarding their own products. For this reason this dissertation only deals with the second subtask while shortly introducing the first one below. A general approach to detect possible targets is to look for frequent nouns or compound nouns in a corpus or use the output of named entity taggers. Despite the simplicity of frequency based methods, they tend to perform well. Another approach is to apply a syntactic parser on sentences and look for various patterns, e.g. an adjective modifying a noun (Schouten and Frasincar, 2016). After having the targets identified, systems for the second subtask can be applied in order to detect related sentiments. We also mention that there are approaches to deal with the two subtasks jointly (Lazaridou et al., 2013).

From now on by target-level sentiment analysis we refer to the second subtask, i.e. given a pair of sentence and target, the task is to correctly classify the polarity of the sentiment that is related to the given target in the given sentence. In the literature target-level sentiment analysis is often referred to as *entity-* or *aspect-level* sentiment analysis, which are slightly different scenarios regarding the possible targets. In the case of the first one, targets during the analysis are bigger entities, e.g. individuals, products or services. Sentences analyzed in this scenario usually contain one expressed opinion, which can convey opposite sentiment polarity for the entities in the same sentence or could be totally unrelated to them as well. In contrast, targets in the aspect-level are properties of an entity, e.g. battery life of a laptop or service quality of a restaurant. Sentences usually contain multiple aspects of an entity, each with a sentiment expression referring to it.

In this chapter we first introduce our document-level sentiment analysis system which is the basis of our target-level system. We show the results of this target-ignorant system on multiple target-level datasets in English and Hungarian. As the main contributions of this chapter, which were published in (Hangya and Farkas, 2013a,b, 2017), we introduce novel techniques for making a system target-aware. We show two target-level feature sets that can be used additionally to document-level ones: *surfaceform-* and *syntax-based*. We deal with both entity- and aspect-level datasets. We develop similar solutions for both scenarios and we analyze their advantages and disadvantages in the discussion.

3.2 Used Datasets

First we introduce the used datasets for the evaluation of our systems. To show the generality of the proposed methods we use 4 annotated corpora. Out of them 3 contains English and 1 Hungarian texts, while 2-2 are annotated at entity- and aspect-level. Details of the datasets can be seen in Tables 3.1 and 3.2 while their description follows below.

RepLab A Twitter database was created for the *RepLab-2013* shared task (Amigó et al., 2013) for entity reputation monitoring. The collection comprises English and Spanish tweets on 61 entities taken from four domains: automotive, banking, universities and music. The names of these entities were used as query to crawl tweets using the Twitter public API. For our study we used the English training portion of the provided database, which consisted of $27,520$ tweets[1]. We show shared task specific experiments on both English and Spanish tweets in Section 3.6.1. For each of the tweets, an entity was given which was the target of the sentiments in the text. The polarity labels could be positive, negative or neutral.

[1]The number of tweets can slightly differ from the original number of annotated tweets due to their availability through the API.

	target	language	genre	avg. doc. length	#labels
RepLab	entity	EN	tweet	14.62	3
OpinHu	entity	HU	news	26.36	3
ABSA	aspect	EN	review	18.27	4
JDPA	aspect	EN	review	25.81	2

Table 3.1: Basic statistics about the used corpora. The columns show whether a corpus is annotated on entity- or aspect-level, its language and genre. We have also listed the average number of words in an instance and the number of distinct labels in the corpora.

	Positive	Negative	Neutral	Conflict	Overall
RepLab	16,362	3,630	7,528	–	27,520
OpinHu	882	1,629	7,495	–	10,006
ABSA	3,169	1,682	1,099	136	6,086
JDPA	3,000	2,932	–	–	5,932

Table 3.2: Label distribution and the overall number of annotated instances in each corpora which were used in our experiments.

OpinHu OpinHuBank is a Hungarian corpus created for sentiment analysis purposes (Miháltz, 2013), which consists of sentences taken from various Hungarian news, blogs and forum sites. Documents coming from these sources were processed with automatic text processing tools and segmented into sentences. Each sentence is at least 7 token long, has proper punctuation at its end and has at least one person entity, detected by a named entity recognizer tool, which is the target in that sentence. The sentences were annotated with three polarity levels, where neutral was also used if both positive and negative sentiments are referring to the given target beside the case when no sentiment is related to it. The corpus contains $10,006$ instances.

ABSA The organizers of the *SemEval-2014 Task-4 – Aspect Based Sentiment Analysis* created a corpus containing $3,045$ and $3,041$ laptop and restaurant review sentences, taken from various customer review sources (Pontiki et al., 2014). For each review, aspects of an entity are annotated, such as the battery life of a laptop. For each aspect notation in a sentence the polarity label is given depending on the sentiments related to the given aspect in that review. In this dataset, 4 polarity labels were used, which are positive, negative, neutral and conflict, where the latter was used when both positive and negative sentiments were referring to the same aspect.

JDPA We also experimented with *The J.D. Power and Associates Sentiment Corpus* (Kessler et al., 2010), which consists of blog posts containing opinions about automobiles and digital cameras. In this corpus, sentiment expressions in sentences are annotated along with their polarities and targets. We used these expression level annotations to create a corpus for the target-level task. We took texts that contained at least one sentiment expression and created a training instance for each

sentiment expression in the corpus using the sentence which contains the expression as the textual content, the expression's annotated target as the target and its label (positive or negative) as the sentiment polarity. Since it is possible for one sentence to contain multiple sentiment expressions, we create as many instances based on the sentence as many expressions it has. We used only the posts about automobiles and this way we got $5,932$ instances.

3.3 Document-Level System

First we show our baseline system for sentiment analysis. It ignores the target in question, thus functions on the document-level extracting global sentiments from texts. We use this system as a starting point and make it target-aware in Section 3.4. The system is composed of two main steps: *preprocessing* normalizes the input texts in order to overcome data sparsity while the goal of *feature engineering* is to extract sentiment indicative features for the machine learning algorithm. We describe both modules below. After having texts represented as features, we apply a *maximum entropy* classifier. We do not compare various classification algorithms because our focus is on the developed feature sets rather than on the classifier algorithms. Our preliminary experiments showed that the maximum entropy classifier has high results performing similar to *support vector machines*. For implementation we used the MALLET toolkit, which is a Java-based package for natural language processing (McCallum, 2002).

We note that the used target-level datasets can be confusing for a document-level system because the instances included do not always reflect the correct global sentiment of a sentence. For example, it is possible that a globally positive sentence is annotated as neutral because the sentiments have nothing to do with the target in question or the same sentence is present multiple times with different sentiment labels due to multiple targets in it. We still used these datasets since our goal is not to show the performance of techniques presented in this section but to introduce a basic system which we extend to target-level in the later sections.

3.3.1 Preprocessing

Before extracting features from the texts, we applied the preprocessing steps listed below. Because corpora containing tweets are noisier than texts coming from more standard and less fast paced sources, cleaning texts is essential for this genre. We introduce some Twitter specific steps but the rest are general enough to be applicable to all used corpora. We apply the following steps during preprocessing:

- Words can be used in multiple forms, i.e. upper or lower cased, in an inflected form, etc. For sentiment analysis most different forms do not contain additional information about the sentiment meaning of a word but they drastically increase the size of the vocabulary, which

leads to data sparsity. In order to eliminate the multiple forms of a single word, we converted them into lowercase form, except those which are fully upper cased because people tend to write in uppercase to convey sentiments. In the case of the Twitter corpora we also stemmed the words with the Porter stemming algorithm (Porter, 1980).

- Emoticons are used frequently to express sentiments. On the other hand, due to their wide variety for both positive and negative sentiments, data sparsity is an issue for them as well. To overcome this problem, we grouped them into positive and negative emoticon classes. We treated *:), :-),:), :D, =), ;), ;), (:* and *:(, :-(, : (,):,) :* as positive and negative and replaced them with the *[POSITIVE_EMOTICON]* and *[NEGATIVE_EMOTICON]* placeholders, respectively.

- In the case of words that contained character repetitions, more precisely those that contained the same character at least three times in a row, we reduced the length of this sequence to three. For instance, in the case of the word *yeeeeahhhhhhh* we got the form *yeeeahhh*. This way, we unified these character repetitions, but we did not lose this extra information since character repetition could indicate sentimental value.

- Although the numbers can hold polarity information in some cases, we have to understand the meaning of the number in the given context to exploit it. For example, consider the following two sentences: *I have 1 dollar left* and *I have 1 task left*. Without deeper semantic analysis it is hard to decide the sentiment value conveyed by the number. Keeping the exact value of numbers introduces data-sparsity problems hence we convert them to the *[NUMBER]* form.

- We replaced the @ Twitter-specific tag and each URL with the *[USER]* and *[URL]* notations, respectively. Next, in the case of a hash tag we deleted the hash mark from it; for example we converted *#funny* to *funny*. This way, we did not distinguish Twitter specific tags from other words.

- We removed the following unnecessary characters: `'"#$%&()*+,./:;<=>\^{}~`.

3.3.2 Feature Set

To represent texts we use **n-gram** (unigram and bigram) features in the bag-of-words model. To further increase the performance of our model, we also employ special features which characterize the polarity of the documents. In the following we introduce these additional techniques.

A good indicator of sentiments is the sentiment polarity of each word in a text. To determine the **sentiment of a word** we employed sentiment lexicons. In the case of English texts, SentiWordNet was used for this purpose (Baccianella et al., 2010). In this resource, synsets, i.e. sets of word forms sharing some common meaning, are assigned positivity, negativity and objectivity scores lying in the $[0, 1]$ interval. These scores can be interpreted as the probability of seeing some representatives of the

synsets with a positive, negative and neutral meaning, respectively. However, it is not unequivocal to determine automatically which particular synset a given word belongs to in its context. Consider the word *great* for instance, which might have multiple entirely different sentiment connotations in different contexts, e.g. in expressions such as *"great food"* and *"great crisis"*. We determine the most likely synset a particular word form belong to based on its context. We select the synset, the members of which are the most appropriate for the lexical substitution of the target word in the target context. The appropriateness of a word being a substitute for another word was measured relying on Google's N-Gram Corpus, using the indexing framework described in (Ceylan and Mihalcea, 2011). We looked up the frequencies of the n-grams which we derived from the context by replacing the target words with its synonyms from various synsets. For example, to decide the synset of the word *great* in the context *food is great*, we count the frequency of the phrases *food is good* and *food is big* in a huge set of in-domain documents (Ceylan and Mihalcea, 2011). Then we choose the meaning (synset) that has the highest probability value, which is *good* in this example.

In the case of the Hungarian corpus we used a simple sentiment lexicon, which contains a set of words with their polarity values also in the $[0, 1]$ interval. The lexicon was constructed in-house by a linguist expert and it contains 3322 words.

After we had assigned a polarity value to each word in a text, we created two new features for the machine learning algorithm, which were the numbers of positive and negative words in the given document. We treated a word as positive or negative if the related positive or negative value was greater than 0.2. We determined this value by manually looking at some entries in the lexicons.

We also tried to group **acronyms** according to their polarity. For this purpose, we made use of an acronym lexicon[2]. For each acronym we used the polarity of each word in the acronym's description and we determined the polarity of the acronym by calculating the rate of positive and negative words in the description. This way, we created two new features that are the numbers of positive and negative acronyms in a given message. This feature is mostly important in the case of tweets.

Our hypothesis is that people like to use **character repetitions** in their words to express their happiness or sadness as already stated before. Besides normalizing these tokens, we created a new feature as well which represents the number of these kinds of words in a tweet. Although the number of these words is not indicative on the positiveness or negativeness of a sentence directly, but it is helpful for deciding whether it is neutral or not.

Beyond character repetitions, people like to write words or a part of the text in **uppercase** in order to call the reader's attention to it. Because of this we created yet another feature which is the number of uppercase words in the given text.

Since **negations** are quite frequent in user reviews and have the tendency to flip polarities, we took special care of negation expressions. We collected a set of negation expressions like *not* and

[2] `www.internetslang.com`

don't and a set of delimiters like *and* and *or*. We assume that the scope of a negation starts when we detect a negation word in the sentence and it lasts until the next delimiter. If an n-gram was in a negation scope, we added a *NOT* prefix to that feature. On the other hand, we did not invert the sentiment polarity of a word coming from the lexicons if it is negated. Our earlier experiments showed that this technique did not improve the results. The reason is that if a word with positive or negative polarity is negated, its polarity does not necessarily get inverted; for instance *not excellent* does not necessarily means that it is awful, rather, the positiveness is weakened. Vilares et al. (2013) created a syntax-based negation detector which can detect negations more precisely in case we have accurate dependency trees. We used this simpler method in order to create a robust system on various text genres.

3.3.3 Results

We report accuracy and F_1 score results on the datasets running 10-fold cross-validation in Tables 3.3 and 3.4. Three baseline systems were used: the most frequent class in the training set is predicted for each test case by *MFC*; only unigram features are used without preprocessing in the case of *unigram baseline*; while *document-level* is the system introduced above. It can be seen that none of the datasets are trivial to solve, thus machine learning based solutions could considerably increase performance comparing to MFC. We use the results of document-level as baseline when introducing target-level techniques next. The system performs well on all datasets. Interestingly, it suffers a small drop on the conflict label in F_1 comparing to the unigram baseline on the ABSA dataset. The reason for this lies in the low number of conflict samples in the corpus. By manual analysis of the results, we conclude that the preprocessing step increased performance to a great extent on the Twitter domain because of the unification of various emoticons, URLs and other noise that are more frequent on this domain. Furthermore, preprocessing was also beneficial for the non-Twitter corpora but it was less significant. We also found that features based on the sentiment lexicon are good indicators of sentiment polarities in a sentence. Often, false predictions are caused by the incorrect weight of a feature learned by the classification algorithm, which is due to data sparsity. By relying on knowledge bases like sentiment lexicons, we incorporate external information to the decision process, which helps the data sparsity problem. For further analysis of document-level features we refer to Section 3.6 and Chapter 5.

3.4 Target-Level Feature Engineering

In this section we introduce techniques which were developed to make the document-level system target aware. The key factor in tackling target-level analysis is to detect parts of the input sentences that are referring to the target in question. For this, we developed methods to emphasize parts of the sentences during classification and also features which indicate sentiments referring to a target. We

	MFC		Unigram baseline			
	acc	F_1	acc		F_1	
RepLab	59.45	24.85	71.93	(+12.48)	64.93	(+40.08)
OpinHu	74.91	28.55	78.82	(+3.91)	59.66	(+31.11)
ABSA	52.07	17.12	64.08	(+12.01)	45.72	(+28.60)
JDPA	50.57	33.58	73.70	(+23.13)	73.66	(+40.08)

	Document-level			
	acc		F_1	
RepLab	73.16	(+1.23)	65.97	(+1.04)
OpinHu	79.20	(+0.38)	60.64	(+0.98)
ABSA	66.48	(+2.40)	45.70	(–0.02)
JDPA	75.23	(+1.53)	75.19	(+1.53)

Table 3.3: Results of three document-level systems for the target-level corpora.

	macro F_1	positive F_1	negative F_1	neutral F_1	conflict F_1
RepLab	1.04	1.10	0.83	1.18	–
OpinHu	0.98	1.29	1.53	0.11	–
ABSA	-0.02	1.04	3.88	2.64	–7.63
JDPA	1.54	1.30	1.78	–	–

Table 3.4: Document-level improvements in terms of F_1 score compared with the unigram baseline for the target-level corpora.

group our proposed methods into two groups. The first set of techniques are based on the surface form of the sentences with the aim of fast and easy applicability on texts coming from various domains and languages. On the other hand, the second batch of techniques aims for a deeper understanding of sentences exploiting their syntax. We evaluate these techniques on the target-level corpora separately and analyze the results in the following section. In addition, we also combine them and show that they complement each other thus pushing performance even further.

3.4.1 Surfaceform-Based Feature Engineering

A good indicator whether a part of the text is related to the target in question is the **distance** between its tokens and the mention of the target in the text. The closer a token is to the target, the more the possibility that the given token is related to the target. For example, consider the following tweet, where the first sentence does not refer to *BMW* at all:

> I do agree that money can't buy happiness. But somehow, it's more comfortable to sit and cry in a BMW than on a bicycle!

For this reason, we weighted each ngram feature in the message by its distance from the mention of the given target using equation 3.1.

$$w(i,j) = \frac{1}{e^{\frac{1}{n}|i-j|}} \qquad (3.1)$$

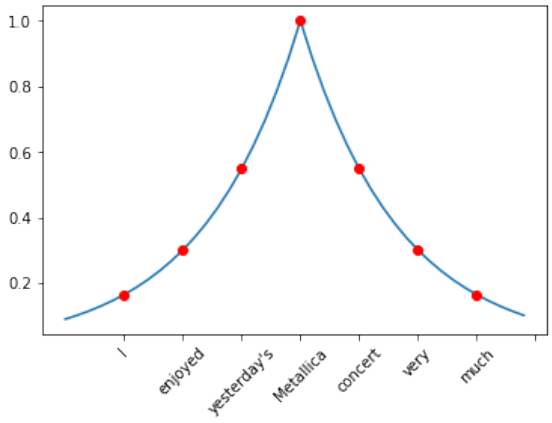

Figure 3.1: Equation for distance-based weighting (left), where n is the length of the input text, i and j are the positions of the actual word and the mention of the target. An example sentence where *Metallica* is the target can be seen on the right.

People tend to write more positively or negatively about certain entities, aspects or topics. We introduce features which make possible for the classification algorithm to learn these phenomena. One can think of these features as **a priori knowledge** about the tendency of sentiments related to different targets and topics. We use the name of the target as a feature during training and prediction to indicate this a priori knowledge about different targets. Similarly, we use topic information for a priori sentiment knowledge of the topic of the sentence. The goal of topic modeling is to discover abstract topics that occur in a collection of documents. We used Latent Dirichlet Allocation (LDA) (Blei et al., 2003) with 50 topics for this purpose, which was trained on the training portion of the given dataset. From the results of LDA, we get the topic distribution for each document, from which we used the three most probable topic IDs as additional features.

3.4.2 Syntax-Based Feature Engineering

Distance-weighting is a simple method for approximating the relevance of a text fragment from the target mention point of view. To understand sentences more deeply, we create features based on their syntax by employing dependency and constituency parsers.

We showed in Section 3.3 that the **sentiment polarities of words** are important features for the classification. We extend this idea further and create syntax specific indicators based on the sentences' dependency parse trees. We defined a feature template for tokens whose syntactic head is present in our positive or negative lexicon. For example, consider the sentence and its parse in Figure 3.2. The word *easy* is represented as a positive word in our sentiment lexicon thus from the dependency of *making* $\rightarrow$ *easy* we create the bigram of *[POSITIVE]_making*. In addition, since our lexicons are not complete, i.e. they do not contain all words that have sentimental value, we

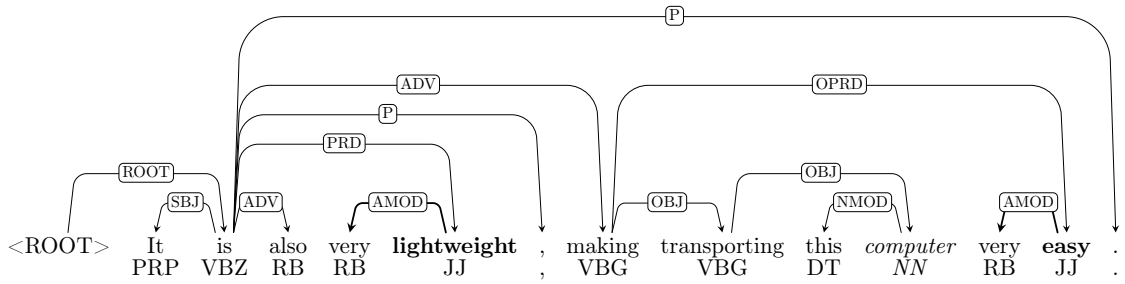

Figure 3.2: Dependency parse tree (Bohnet parser), where *computer* is the target. Adjectives that are in close proximity are indicated in bold and also the relations whose head word is in the sentiment lexicon.

also consider **adjectives**. We create bigram features if the token of the target is modified by an adjective. We consider the target modified by an adjective if a non-directed path between them in the dependency tree exists with a length less than 6. In Figure 3.2 there is a 3 long path between the adjective *easy* and the target *computer* thus we create the bigram feature *easy_ computer*.

We also attempted to identify longer phrases which refer to the target. In a sentence, we can express our opinions about more than one target, so it is important to ditinguish phrases containing opinions about other targets. We developed a simple rule-based method for selecting the appropriate **subtree** from the constituent parse of the sentence in question (see Figure 3.3). In this method, we set the leaf node which contains the given target as the root of this subtree in the first step. In subsequent steps, the subtree containing the target in its yield gets expanded until all the following conditions are met:

- The yield of the subtree consists of at least five tokens.
- The yield of the subtree does not contain any other target besides the five-token window frame relative to the target in question.
- The current root node of the subtree is either the non-terminal symbol `PP` or `S` in English and `CP` or `NP` in Hungarian.

Relying on these identified subtrees, we introduced novel features. We created additional n-gram features from the yield of the subtree to emphasize these phrases. In addition, in the case of English texts, we determined the **polarity of this subtree** with a method proposed by Socher et al. (2013) and used it as a feature. Note that the training of the system of (Socher et al., 2013) is resource heavy and we propose an alternative method for fine-grained sentiment analysis in Section 4.

For English texts we used the Bohnet dependency parser (Bohnet, 2010) and the Stanford constituency parser (Klein and Manning, 2003). For Hungarian, we used the Magyarlanc (Zsibrita

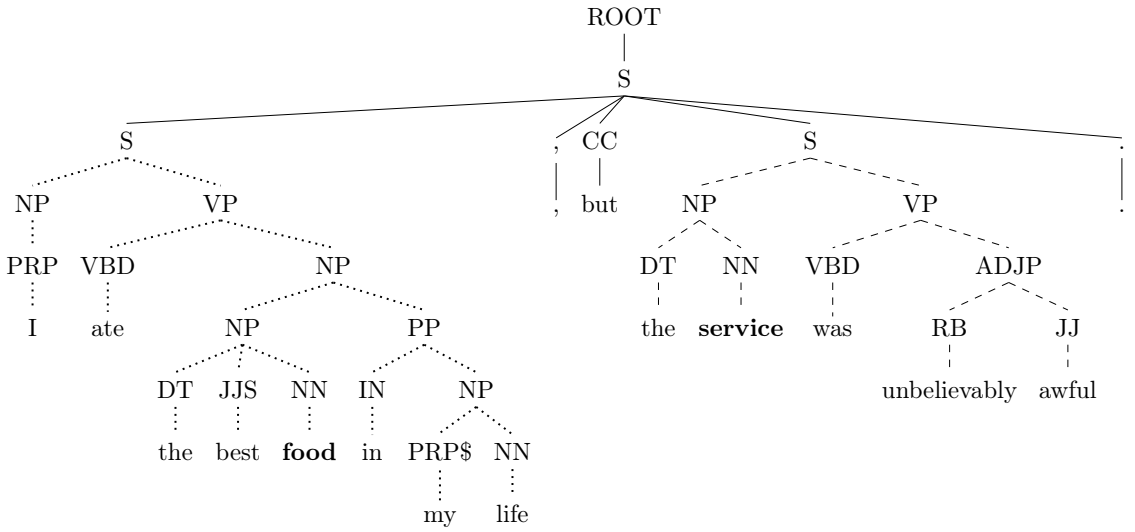

Figure 3.3: Constituency parse tree (Stanford parser), with dotted and dashed subtrees for the *food* and *service* targets, respectively.

et al., 2013) and the parser of Szántó and Farkas (2014) for dependency and constituency parsing, respectively.

3.4.3 Results

The results of the target level techniques are shown in Tables 3.5 and 3.6. We used the developed techniques besides those in the document level settings. In Table 3.5, the results of the surfaceform-based and the syntax-based features can be seen separately as well as jointly in the so-called *hybrid* system where we used both sets of techniques. All three systems were compared with the document-level system in Table 3.3.

It can be seen that in most of the cases we managed to improve the results with both techniques. The syntax-based features were more useful due to their capability to identify the clauses which are related to the given target more precisely. The exception is the RepLab database, where weighting was more useful than the parser. The reason for this is that syntactic parsers tend to perform worse on the informal Twitter messages. We achieved the best results by combining the two techniques in the hybrid system. The only exception is the RepLab Twitter corpus, where we got the best results by using only the surfaceform-based features. Similarly to Vilares et al. (2015), who experimented with syntactic features on Spanish tweets, we found that syntactic parsing just confused our system. Kong et al. (2014) created a Twitter specific dependency parser which can handle the characteristics of tweets. We ran the same experiment with all of our features (hybrid) on the RepLab corpus and only replaced the Bohnet dependency parser with the Twitter specific one. We got 0.18% increase

	Distance-weighting				Syntax-based			
	acc		F_1		acc		F_1	
RepLab	74.12	(+0.96)	67.68	(+1.71)	74.06	(+0.90)	67.34	(+1.37)
OpinHu	79.94	(+0.74)	61.55	(+0.91)	80.27	(+1.07)	61.91	(+1.27)
ABSA	67.23	(+0.75)	46.51	(+0.81)	68.23	(+1.75)	47.51	(+1.81)
JDPA	75.10	(–0.13)	75.05	(–0.14)	77.32	(+2.09)	77.27	(+2.08)

	Hybrid			
	acc		F_1	
RepLab	74.09	(+0.93)	67.36	(+1.39)
OpinHu	80.74	(+1.54)	62.71	(+2.07)
ABSA	68.40	(+1.92)	47.50	(+1.80)
JDPA	77.38	(+2.15)	77.36	(+2.17)

Table 3.5: Results of the target-level systems (distance weighted n-grams, syntax-based features and both of them). All three systems were compared with the document-level system (differences are in parentheses).

	macro F_1	positive F_1	negative F_1	neutral F_1	conflict F_1
RepLab	1.39	0.46	0.89	2.83	–
OpinHu	2.07	0.27	4.97	0.96	–
ABSA	1.80	1.22	3.18	5.07	–2.26
JDPA	2.16	1.72	2.61	–	–

Table 3.6: Target-level (hybrid) improvements in terms of F_1 score compared to the document-level.

in accuracy, which indicates that the Twitter specific parser performs better on tweets than those which were trained on more standard texts. On the other hand, the difference is small, i.e. the simple distance weighting method performs similarly to a syntactic parser on tweets in contrast with the well-formed reviews. We have to note that the reason for the relatively small improvement might be the fact that our polarity classification system is not relying heavily on dependency trees, i.e. further improvements might be achievable if these specialties would be exploited. In Table 3.6, F_1 score differences among the hybrid and the document-level systems can be seen. The F_1 score of conflict label for the ABSA corpus was worse due to the low number of conflict documents, however, all the other values were better.

All our improvements over the baseline document-level system are statistically significant at 0.05 significance level with McNemar's test. In this section, we showed that the proposed techniques make a simple baseline system target-aware, which further improves performance. Next, we discuss results and errors in more detail.

3.5 Discussion

In the previous section we reported quantitative results of our system. Now we discuss the results and also analyse some representative examples in more detail.

Document-level features are meant to capture the global sentiment of the sentences, thus they cannot handle opinion about a particular target only. In our target-level system, we first tried to capture and emphasize clauses which are relevant to the target in question. In the following example, the targets of the sentence are *controls*, *gauges* and *numeric counter*.

All the ***controls*** *looked to be in a good place: easy to read* ***gauges****, although the* ***numeric counter*** *on the speedo was small.* (3.1)

The document-level system manages to decide the global polarity of the sentence correctly, which is positive in the example, but it assigns this polarity to all the targets because the same features are extracted in each case. On the other hand, the target-level system can differentiate by extracting different sets features for each target. In the example, the yield of the selected syntactic subtree for the target *gauges* is *to be in a good place: easy to read gauges*. Emphasizing this clause makes it clear for the classifier that the sentiments related to the target are positive. Similarly for the target *numeric counter*, the sentiment was correctly classified as negative based on the target's yield *the numeric counter on the speedo was small*. A more complex task is to decide the polarity related to the target control because the yield of the selected subtree is the whole sentence that contains both positive and negative sentiments. In this case, our distance-weighted bag-of-words features emphasize the positive *good* and *easy* words against the negative word *small*. Using these techniques we improved the accuracy by 1.63% in average.

Table 3.6 shows that the average improvement in F-score comparing to the document-level system is +1.83 for positive and negative labels, which is lower than that for the neutral label. From this, we may conclude that target-level features are useful for deciding whether a sentiment in the text is related to the target in question because we were able to classify more precisely those cases that were neutral, but were classified as positive or negative by the document-level system.

3.5.1 Error analysis

We manually investigated incorrectly classified sentences in order to reveal typical and critical sources of failures. We randomly sampled 100 incorrectly classified sentences by the hybrid system from all target-level corpora. Based on manual analysis of these documents, we recognized two error categories related to the target.

One target specific feature used by our system is the name of the target. The goal of this feature is to make the classifier able to learn the **a priori sentiment about a given target**, i.e. whether people tend to speak positively or negatively about it. We showed that this feature increased the accuracy of the system. In contrast, in cases when the sentiment in the text is not referring

to the target, i.e. the document is neutral, the a priori sentiment caused misclassifications. This phenomenon covers 19% of the manually examined errors. The following example was incorrectly classified as positive instead of neutral due to the usual positive sentiments about BMW:

*Can I ride with you in your **BMW**?* (3.2)

The second error category is caused by **incorrect subsentence detection** (13%). We introduced the distance-weighting method and the syntax-based features to emphasize phrases in a sentence which are strongly related to the target. We showed that this technique was crucial in making the system target-aware. On the other hand, these techniques can also cause errors by selecting wrong phrases, which can happen in the case of incorrect syntactic parsing or in the case of long distance between the target mention and the sentiment expression. The former can only occur with the syntax-based features and can be eliminated with a better parser. In contrast, the latter occurs with both of the introduced techniques and to resolve the issue, deeper understanding of the sentence is needed. Consider the following example for the second case:

*I had **dinner** with my best friend last week but it was inedible.* (3.3)

which was classified as positive since the word *best* is near to the target while the related sentiment is expressed at the end of the sentence. One way to overcome the error in this sentence is to employ coreference resolution (Jurafsky and Martin, 2009) but further investigation is needed, which could be done as future work in this field.

3.6 Task-Specific Systems

In this chapter we focus on shared tasks that initially drove our attention to research in this area and we describe the systems which we developed for these shared tasks. Since these systems were aimed to compete on the then ongoing tasks, they contain slight differences compared to the system in the previous section while the latter is more general and widely applicable. We indicate these differences in the sections describing these systems. In Section 3.6.1 we detail our surfaceform-based submission (Hangya and Farkas, 2013a) to the *RepLab-2013 – An evaluation campaign for Online Reputation Management Systems* (Amigó et al., 2013) shared task, which outperformed all participants. We show our syntax-based submission (Hangya et al., 2014) to the *SemEval-2014 – Aspect Based Sentiment Analysis* (ABSA-2014) (Pontiki et al., 2014) in Section 3.6.2, which was also competitive comparing to other participants. We show a more detailed quantitative analysis of our different feature sets. We also refer to some of our other works in this area for completeness. In (Hangya et al., 2015) we describe our target-level sentiment analysis system for Hungarian and in (Szabó et al., 2016; Szabó et al., 2016) we took part in a project involving the annotation of an

aspect-level Hungarian sentiment corpus.

3.6.1 RepLab-2013

The goal of the *RepLab-2013 – An evaluation campaign for Online Reputation Management Systems* challenge (Amigó et al., 2013) was to monitor the reputation of several entities, like companies, organizations, celebrities, etc. exploiting Twitter messages. Target entities are given for each tweet. The organizers defined four tasks, namely filtering, sentiment classification, topic detection and assigning priority to topics, from which we took part in the first two ones. In the case of the filtering task the goal was to determine which tweets are related to a given entity and which are not, for instance, distinguishing between tweets that contain the word "Stanford" referring to the Stanford University or to Stanford as a place. This step could be considered as preprocessing for the later steps. In this section we only focus on the latter while we describe our filtering approach in (Hangya and Farkas, 2013a).

Data The data provided by the organizers of the challenge consist of English and Spanish tweets. In Section 3.2 we introduced how the data were assembled and the size of the English portion which we used in the experiments in Section 3.4 with 10-fold cross-validation. For the shared task experiments we used the official dataset split consisting of $45,679$ training and $96,848$ undisclosed test tweets with $4:1$ ratio of English and Spanish instances.

System As the sentiment classifier we used the system introduced in Section 3.4.1, i.e. the document-level system with preprocessing and general feature engineering with the additional surfaceform-based feature sets to make it target-aware. The majority of our techniques are language independent thus we applied the same system to the Spanish texts as well without any language specific feature engineering. During our initial experiments we compared training classifiers separately on the English and Spanish subsets against concatenating instances and training a joint classifier. We got better results with the latter thus we report these results. The reason for better performance with the joint model is that the Spanish subset contains a small number of English tweets as well, probably due to incorrect language detection, which lets the joint model benefit from the English training data in the case of the Spanish test set.

Evaluation Two measures were used as official evaluation metrics: accuracy and the reliability and sensitivity (R&S) based F_1 score (Amigó et al., 2012). R&S assumes that any organization task consists of a bag of relationships between documents. In brief, R&S computes the precision and recall of relationships produced by the system with respect to the gold standard. In case of sentiment analysis the relationship values could be $r \in \{<, =, >\}$ by considering the negative, neutral and positive sentiment labels as -1.0, 0.0 and 1.0 values respectively. In other words, reliability,

	acc
baseline	65.0
+ preprocessing	66.1
+ bigrams	67.5
+ extra features	67.6
+ weighting	67.7
+ entities	68.1

Table 3.7: Sentiment classification accuracy on the RepLab dataset while gradually enabling the developed techniques.

similarly to precision, calculates the percentage of document pairs that have the correct relationship value r, based on their predicted sentiment label and the gold annotation, out of all pairs predicted to have relation r. Sensitivity is similar to recall and it is calculated analogously to reliability. Reliability and sensitivity are combined to the F_1 score, by calculating their harmonic mean. The reason for using this measure instead of the traditional precision and recall based F_1 score is that it gives the possibility for exploiting the relation between sentiment labels. In addition, it was shown that this measure satisfies more formal constraints than previously existing evaluation metrics. For more details about R&S we refer to (Amigó et al., 2012) and (Amigó et al., 2013).

Results

In Table 3.7, we show the effects of our developed techniques as we gradually enable them on the training set using 10-fold cross-validation. Later, we show the official shared task results on the test set and we analyze domain differences at the end of this section. Our baseline system uses only unigram features without any preprocessing steps or extra features. *Preprocessing* increased the accuracy considerably, which shows the noisiness of the Twitter genre. In the next steps we enabled *bigrams* and the *extra features*, namely the sentiment of words and acronyms, the presence of character repetitions and upper case words and negation detection, which further increased performance. The target-level features *distance-weighting* and the a priori knowledge about the *entities* give further boost to the system in cases when the global sentiment of the input is different from the target related sentiment. We note that the numbers are lower in these experiments compared to those in Section 3.4 due to the inclusion of Spanish tweets in the used data.

In Table 3.8 we show the additional effects of the topic based a priori information using different numbers of LDA topics. By not using enough number of topics (20), performance was decreased compared to the best number in Table 3.7 due to the incorrect clustering of tweets. On the other hand, by using more topics we achieved further improvements. In our final system we used 50 topics due to runtime considerations.

We show results on the the official test data in Table 3.9. Interestingly, some teams achieved

topic number	acc
20	68.0
50	68.2
100	68.3

Table 3.8: Results with different numbers of LDA topics on the RepLab dataset.

team	accuracy	reliability	sensitivity	F_1
SZTE_NLP	69	48	34	38
LIA	65	37	27	29
POPSTAR	64	43	34	37
UAMCLYR	62	33	29	30
UNED ORM	62	32	29	30
BASELINE	58	32	29	30
IE	58	29	22	25
NLP IR UNDED	58	33	31	32
DIUE	55	33	22	25
VOLVAM	54	31	39	34
DAEDALUS	44	31	40	34
GAVKTH	37	37	21	27

Table 3.9: Official accuracy, reliability, sensitivity and F_1 scores on the test data for each participant. Our system is called SZTE_NLP.

high accuracy but lower F-measure and vice versa, for example team VOLVAM (Mosquera et al., 2013). The reason for this is that we do not need to predict the correct labels of tweets to achieve high F_1 score, just the correct relation between them. Our submission *SZTE_NLP* was ranked top compared to other participants.

3.6.2 ABSA-2014

The aim of *SemEval-2014 – Aspect Based Sentiment Analysis* (Pontiki et al., 2014) was to identify aspects of given entities and classify sentiments referring to them. In contrast to RepLab, where the targets of the target-level analysis task are entities, in this task various aspects (battery life, food quality, etc.) of entities (smartphone, restaurant, etc.) are given as targets. Four subtasks were defined, which are aspect term extraction, sentiment and category prediction of aspect terms and category polarity classification. In the following, we describe our submission (Hangya et al., 2014) to the aspect sentiment classification subtask where aspects are given for each input text.

Data We have introduced the ABSA dataset in Section 3.2. As a quick recap, it contains laptop and restaurant reviews in English. We used the $3,041$ and $3,045$ restaurants and laptop instances as training, respectively and the additional official $800 - 800$ test instances for evaluating.

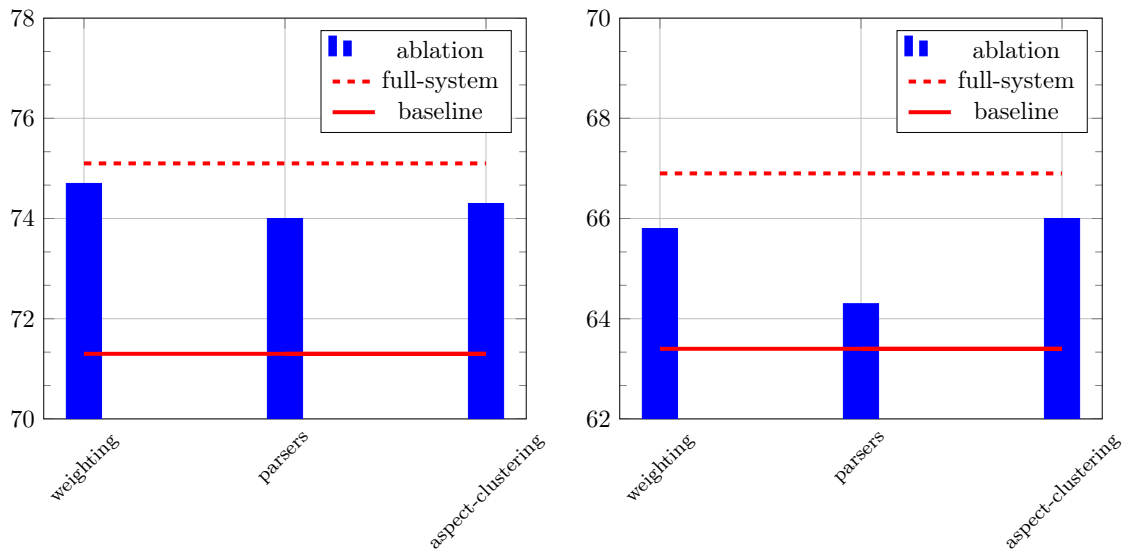

Figure 3.4: Ablation study showing accuracy scores on the restaurant (left) and laptop (right) test data.

System As the system for the target-level sentiment classification we used our *hybrid* approach, i.e. the document-level system with various feature sets and both surfaceform-based and syntax-based feature sets (Section 3.4). The only difference is that instead of using LDA for topic detection, we clustered known aspect terms based on the context they are used in. We represent each aspect by a vector which indicates the words co-occurring with them. Singular value decomposition was then used to project these aspect vectors into a lower dimensional semantic space on which k-means clustering (with $k = 10$) was carried out to acquire the clusters. For each classification instance, we regarded the cluster ID of the particular aspect term as a nominal feature.

Results

In this section, we perform an ablation study[3] to show the effect of different features sets separately on both restaurant and laptop test sets. We measure performance using accuracy, which was the official evaluation metric of the shared task. The results of the ablation study can be seen in Figure 3.4. On the x-axis the accuracy drop can be seen comparing to our *baseline* (n-gram features only) and *full-system*, while turning off various sets of features. First, the distance-*weighting* of features are turned off, then dependency and constituency parsing is omitted and finally the feature based on aspect clustering is absent. It can be seen that omitting any of the target-level features caused a significant drop in performance. For both domains syntactic parsing caused the highest drop, which shows that in the case of more standard text, on which parsers can be applied with

[3]The ablation study was performed after the official shared task evaluation.

Team	restaurant	laptop
DCU	80.95	70.48
NRC-Canada	80.15	70.48
SZTE-NLP	75.22	66.97
UBham	74.60	66.66
USF	73.19	64.52
ECNU	70.72	61.16
BASELINE	64.28	51.37

Table 3.10: Results of several high ranked participants out of 26. Our system named SZTE-NLP was ranked 6^{th} and 3^{rd} on the restaurant and laptop domains, respectively.

good accuracy, syntax-based features can really help in detecting text parts that refer to the target. Feature sets are complementary and by turning all of them on, we achieved the best results.

In Table 3.10 the results of several participating teams (out of the overall 26) can be seen on the restaurant and laptop test data. The official baseline system performs a per aspect term k-nn classification based on the Dice coefficient of sentences. Our submission achieved the 6^{th} and 3^{rd} best results on the restaurant and laptop domains, respectively.

3.7 Related Work

Target-level analysis is a fundamental sub-task of sentiment classification (Pang and Lee, 2008; Liu, 2010). It can be divided into two major subtasks which are aimed at by most methods: target-detection and sentiment classification. A simple yet effective approach is based on mining frequent nouns for the former. The mining of various aspects of entities were proposed in (Hu and Liu, 2004b,a) based on frequent nouns and compound nouns. Since not all frequent nouns are aspects, they were pruned by removing single-word aspects which appear only as part of a multi-word aspect. Hai et al. (2011) made the process more accurate by mining frequent association rules of nouns and sentiment words based on word co-occurrences. The disadvantage of this process is that words which co-occur in the same sentence are not always related. To overcome this issue, Zhao et al. (2010) mine frequent syntactical patterns between targets and sentiment words based on annotated data to detect targets more precisely.

Many shared tasks were proposed to boost research for the second subtask, which also motivated our work (Amigó et al., 2013; Pontiki et al., 2014, 2015, 2016). Most of the participating systems were based on supervised classification using specially engineered features. The most useful sets of features for sentiment classification in general are those based on sentiment lexicons due to the fact that they enable classifiers to incorporate external knowledge in the procedure (Kiritchenko et al., 2014). Other higher level features like irony or negation detection are also used by many systems (Wiegand et al., 2010). The recent advancements of neural networks made them competitive for

sentiment analysis as well. A top performing system (Deriu et al., 2016) of SemEval 2016 was based on a convolutional neural classifier (Kim, 2014) using pre-trained word embeddings (Mikolov et al., 2013a) and distant supervision. To make a system target-aware, approaches try to capture the contexts which are related to the given target. Weighting words instead of using binary or frequency based text representations was applied by multiple approaches to emphasize important parts of the input (Filgueiras and Amir, 2013; Paltoglou and Thelwall, 2010). Cossu et al. (2013) used Continuous Context Models, which tend to capture and model the positional and lexical dependencies existing between a given word and its context. Similarly, in our system, distance weighting was proposed to model the context of a given target (Hangya and Farkas, 2013a). Jiang et al. (2011) proposed an SVM-based classifier for classification of tweets which incorporates target-dependent features and related tweets (retweets and replies) as well to maximize the amount of information. On more standard texts, syntactic parsers were applied by many approaches to mine additional features. Similarly to distance weighting, the distance of sentiment words and the target mention based on dependency parse tree were used as additional feature in (Wagner et al., 2014). Our systems (Hangya et al., 2014) exploited both constituency and dependency parse trees to extract target-related features. A target-aware neural system was proposed by Zhang et al. (2016) which learns context specific features using the given targets.

3.8 Summary of the Thesis

In this chapter the problem of target-level sentiment analysis was introduced. The contributions are the different sets of novel target-specific features which make a simple document-level classifier target aware. The contributions were grouped into two sets: surfaceform-based and syntax-based. The former aimed for a shallow understanding of sentences relying on the lexical distance of words. Furthermore, these techniques also exploited common sense sentimental trends about possible targets and topics. The latter set of techniques exploited the syntactic structure of the sentences aiming at their deeper understanding.

First, techniques were compared on 4 datasets coming from different genres and languages (Hangya and Farkas, 2017). It was shown that the detection of target related parts of sentences (especially with syntactic parsers) is superior to shallow understanding in terms of sentiment classification performance. On the other hand, parsers do not perform well on noisy genres like Twitter thus surfaceform features excel in this case. Furthermore, the best results could be achieved by combining all techniques.

It was also shown that the proposed methods perform well compared to other techniques. In the *RepLab–2013* shared task surfaceform-based techniques reached state-of-the-art performance by ranking 1^{st} among all participants (Hangya and Farkas, 2013a,b). Furthermore, it was also presented that most of the developed techniques are language independent since they perform well

on Hungarian and Spanish texts as well. Our system which introduced the syntax-based features and was submitted to the *ABSA–2014* shared task was also competitive with other state-of-the-art systems (Hangya et al., 2014). On datasets containing laptop and restaurant reviews the system was ranked 3^{rd} and 6^{th}, respectively during the official evaluation.

Chapter 4

Fine-Grained Sentiment Analysis

4.1 Introduction

The general approach to sentiment analysis is to predict one sentiment label, either on sentence- or target-level, for each of the input sentences. In this chapter we dig deeper and analyze sentences in a more fine-grained way. Instead of predicting one label per text instance, we predict a series of sentiment labels for the substructures of the input. The goal of such an analysis is twofold: by analyzing the substructures we can exploit linguistic phenomena such as conjunction, negation, intensification, etc., which results in deeper understanding of texts thus further improving performance; and we can use such a fine-grained analysis for extrinsic tasks, like target-level sentiment analysis, for feature engineering in our case.

Recent studies have been experimenting with the effects of syntactic structure of sentences on enhancing sentiment analyzers not only for target-level but sentence-level as well. Most of these proposals use hand-crafted rules based on the syntactic parse of the sentence (Vilares et al., 2013). These rules are engineered to address certain restricted sets of challenges of sentiment analysis, like negation and intensification. To overcome the need of such manual feature engineering and to cover a wider range of language phenomena, the *Recursive Neural Tensor Network* (RNTN) and the *Stanford Sentiment Treebank* were introduced (Socher et al., 2013). In the treebank, a sentiment label was manually assigned to each constituent of the sentence's phrase structure parse. This treebank can be utilized as a training dataset for fine-grained sentiment classifier methods and it makes it possible to exploit the syntactic structure of sentences without restricting the models to a closed set of language phenomena, neither demands the direct modeling of those phenomena. It enables the application of supervised machine learning techniques to model how morphosyntactic and lexical structures alter the polarity of constituents.

On the other hand, the fully supervised approach has the disadvantage of requiring a manually annotated sentiment treebank. This treebank is domain-dependent, i.e. sentiment analyzers trained

on it perform well only in-domain while the annotation of new treebanks for other domains is expensive. To overcome this issue, we propose a *latent syntactic structure-based approach* which requires only sentence-level polarity labels for training. We eliminate the need of an annotated sentiment treebank by handling the polarity labels of internal nodes in parse trees as latent variables. Our hypothesis is that given a large amount of sentence-level annotated data, the parameters of latent variables can be inferred using expectation-maximization. Since sentence-level polarity labels can be easily obtained in a huge amount for various domains, take for instance pro/con or bottomline summaries of product review sites, our system is easily applicable in various scenarios.

In this chapter we introduce our latent syntactic structure-based method for fine-grained sentiment analysis. We run two experimental setups for the investigation of the proposed approach. The first batch of our experiments show that the sentence-internal latent structure improves the prediction of sentence-level polarity by exploiting various language phenomena. The second batch of experiments show that the sentence-internal latent structures themselves are also meaningful when we extract features from them for the target-level task. We run our experiments on movie, IT products and restaurant reviews to show that our approach is easily applicable to various domains. We show that a sentiment analyzer that exploits syntactic parsing and has access only to sentence-level polarity annotation for in-domain sentences can outperform state-of-the-art models that were trained on out-domain parse trees with sentiment annotation for each node. The results of this chapter were published in (Hangya et al., 2017).

4.2 Latent Syntactic Structure-Based System

The goal of our fine-grained sentiment analysis approach is not only to predict a sentence-level sentiment label for the input sentences, but to detect the sentiment of each individual sub-sentential unit. We represent a given sentence using a tree structure and consider each node of the tree as a unit which has to be assigned a sentiment label during decoding. In the following subsections we discuss the tree structure for representing input sentences, the decoding, training and feature engineering methods separately.

4.2.1 Sentiment Tree Representation

We follow the approach of (Socher et al., 2013) and use constituency parsing to acquire an initial tree representation of input sentences. The use of such parser is motivated by the fact that it detects grammatical units of the inputs and builds a tree structure from them based on their grammatical relations. Since one of the goals of our approach is to learn and exploit these relations between sub-sentential units, it serves as a good starting point. Having a fixed tree structure, we can assign a sentiment label for each node of the tree, which gives the output of our system.

Since our goal was to develop a broadly applicable system, we assume that it has access to

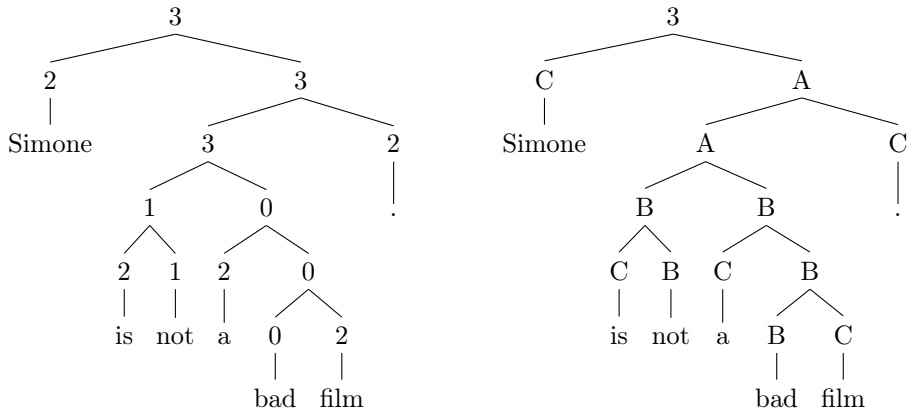

Figure 4.1: Representation of sentiment trees in the Stanford Sentiment Treebank (Socher et al., 2013) (left) contains 5-level sentiment annotation {`0=very negative`, `4=very positive`} for each node of the binary syntactic tree. On the other hand, we assume that we have access only to sentence-level polarity annotation, i.e. only the label of the root is given (right). Here, the states of the inner nodes are described by latent discrete variables {`A,B,C`}.

only sentence-level binary sentiment (`3=positive` and `0=negative`) annotations during training, which serve as the label of the syntactic parse tree's root node. As the labels of inner nodes of the tree are unknown, we treat them as latent discrete random variables. In our experiments, we use latent variables with three possible states `A,B,C` to be able to capture positive, negative and neutral sentiments, although the latent states are not directly connected with the sentiment polarities. In addition, the number of possible states can be easily changed but we postpone the investigation on the effect of different state space sizes for future research. We compare our proposed latent sentiment tree against the fully annotated Stanford Sentiment tree representation in Figure 4.1.

4.2.2 Latent Structure Decoder

Given the tree structure of an input sentence, the labels of inner nodes have to be decoded. We use the structured perceptron algorithm for this task (Collins, 2002). The decoder iterates through the tree in a bottom-up fashion while decoding labels and extracting local features for each node. We describe the used features in Section 4.2.4. Preliminarily we experimented with non-local features along with a beam-search decoder but their improvement was not considerable while running times increased exponentially. Based on the features each derivation is scored and the decoder selects the top scored derivation as the *prediction* with latent variables {`A,B,C`} at the internal nodes and polarity labels {`positive, negative`} for the root node. It is an exact search, i.e. the derivation space is complete, we do not filter the possible derivations. The branching factor of the derivation space is 9 as we work with binary parse trees and 3 possible states.

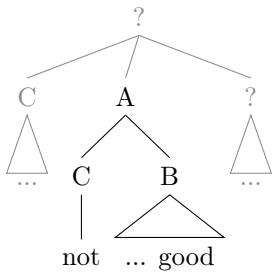

Figure 4.2: The highlighted subtree with latent labels {`A, B, C`} is the subject of local feature extraction for node `A`.

4.2.3 Training Algorithm

At training time, only the root label is available for the preprocessed sentences and the in-sentence polarity labels at the nodes of the parse tree is handled to be latent variables. We follow an Expectation-Maximization (EM) approach for training the structured perceptron. In the E-step, for each input examples in the batch we select a *gold derivation* which is the top scoring derivation matching the given gold standard label at its root node. In the M-step, the update rule of the averaged perceptron is employed (Collins, 2002), i.e. we update the model weights based on the predicted derivation and the gold derivation if their root labels differ. We set the algorithm parameters as follows, relying on preliminary experiments: learning rate 0.1, batch size to 30 and number of epochs 15. We used an in-house implementation of the EM algorithm with the Adagrad optimizer (Duchi et al., 2011).

4.2.4 Tree Based Feature Engineering

We implemented three feature templates to extract information from derivation candidates. We use only local features, i.e. which can be extracted from the 1-level subtree of the derivation (see Figure 4.2). The bottom-up decoder extracts features for each node of the derivation and adds them to its feature vector. We note that our main objective was to investigate whether our latent representation can improve in-sentence sentiment analysis. There is plenty of space for feature engineering, i.e. introducing other local or non-local features (e.g. the cube pruning approach provides an efficient procedure for incorporating non-local features which exploit the whole subtree (Huang, 2008)). The used features are the following:

- **Word features** extract the co-occurrence of a latent polarity label and the unigrams in the yield of the syntactic parse tree's node. From node `A` on Figure 4.2 we extract the following features: A-NOT, A-GOOD, etc.
- **Label features** describe the latent structure as they are the rules in the context-free grammar

		#sentences	#tokens	avg. token/sent. (%)	positive sent. ratio (%)
IT Products	train	254,700	4,821,072	18.93	49.03
	test	109,888	2,072,559	18.86	47.42
Restaurants	train	101,964	1,609,734	15.79	75.16
	test	43,700	693,605	15.87	74.63
Movie	train	599,121	4,022,494	6.71	50.02
	test	1,998	33,460	16.75	53.55

Table 4.1: Basic statistics of datasets with sentence-level annotation.

terminology. From our example we get: A-C-B.

- **Compositional features** are similar to the label features but we exchange one of the daughters' state label with the head of the particular constituent of the syntactic parse tree. This feature template is designed to capture the lexical dependencies of polarity changing words. In the case of the current example we get: A-C-GOOD, A-NOT-B. We experimented with two head finding strategies but the difference between taking the right-most word and the semantic head finding rules of Collins (2003) was negligible thus we used the former method because of its simplicity.

4.3 Sentence-Level Experiments

The goal of the sentence-level experiments is to classify the polarity of the sentiments which a given sentence globally conveys. In our first batch of experiments, we investigated whether the sentence internal latent structure helps the prediction of the sentence-level polarity. In the following we introduce the used corpora for both tasks and detail our setups and results afterward.

4.3.1 Datasets

For our sentence-level experiments we used 3 new corpora since the data used in the previous chapter were annotated on target-level. We used texts from the IT products, restaurants and movies domains annotated with `positive` or `negative` labels on the sentence level. Table 4.1 summarizes the basic statistics of the datasets. In our experiments we used train and test sets with 100,000 (10,000) and 1,000 randomly sampled text examples respectively in case of each domain.

IT Products We downloaded reviews from the Newegg[1] site. Each review on this site must contain the short pro and con summaries of the review in free textual form. We have downloaded the pros and cons of those products which were in the IT category and used them as positive and negative examples, respectively. The downloaded texts were noisy because many of them did not

[1] `www.newegg.com`

contain the appropriate sentiment (e.g. *PRO: I didn't find any.*). To overcome this problem we used only those texts the token length of which is between 6 and 40 and contain only one sentence.

Restaurants In the case of the restaurant review domain, we applied a similar procedure. We used the dataset provided by Yelp[2], which contains reviews about businesses (we only used those which are related to restaurants). Each review is annotated with stars from 1 to 5 by the reviewer. We selected only the ones which were annotated with 1 or 5 as negative and positive examples, respectively. Based on our manual analysis texts, starred between 2 and 4 are noisy in terms of their conveying sentiments thus we ignored them. In order to filter out further noise, we applied the same method as before.

Movie For the movie review domain we downloaded reviews similarly to (Socher et al., 2013) and (Dong et al., 2015) from Rotten Tomatoes[3]. We filtered this dataset as well and used only the reviews with score 1 and 5 as negative and positive examples.

4.3.2 Experimental Setup

Preprocessing Sentences are tokenized by the Stanford CoreNLP toolkit (Manning et al., 2014). The syntactic structure of sentences are fixed, i.e. we syntactically parse each sentence in the preprocessing step. We employed the BerkeleyParser (Petrov et al., 2006), a state-of-the-art phrase structure parser, with the English $6th$ iteration model. We used right-branch binarised and unlabeled (both POS tags and internal node labels are deleted) syntactic parse trees as input.

We predict the whole sentiment tree for the test sentences and we considered the label on the resulting trees' root node as the predicted sentence-level polarity. For comparison, we ran the RNTN system introduced in (Socher et al., 2013), which, similarly to our system, yields sentiment trees based on constituency parsing. The difference of this system and ours is that it was pretrained[4] on fully annotated trees from the Stanford Sentiment Treebank[5]. The system can predict sentiment labels along with their probability values on a five level scale (`very negative, negative, neutral, positive, very positive`). Using the label probability values we mapped its prediction to `positive` or `negative` labels according to the highest probability value ignoring the neutral label[6].

[2] `www.yelp.com/dataset_challenge`

[3] `www.rottentomatoes.com`

[4] We re-trained the system on the training part of the Stanford Sentiment Treebank instead of using the public model.

[5] The Stanford Sentiment Treebank was composed of movie reviews from RottenTomatoes.

[6] We experimented with various mapping strategies from 5 polarity levels to 2 levels but the difference between the achieved accuracies were negligible.

		IT products	Restaurants	Movies
	RNTN	62.1	79.1	76.9
10K	baseline	77.4	91.8	67.8
	latent	76.5	91.9	67.9
100K	baseline	82.9	93.6	75.7
	latent	83.4	93.8	76.6

Table 4.2: Accuracy scores achieved on the three domains. RNTN is our reference system (Socher et al., 2013), the baseline is a model using word unigrams only and latent refers to the proposed system.

4.3.3 Results

Our results can be seen in Table 4.2, which contains the accuracy of the systems of each domain. Our baseline system used only unigram features and a maximum entropy classifier, so it could not exploit the inner structure of the sentences. We used a small (10K sentences) and a big (100K sentences) training set to evaluate the unigram baseline and the proposed latent representation-based models.

Two conclusions can be made based on Table 4.2. Firstly, in the case of all the three domains with 100K training examples the latent system outperformed the baseline. This shows us that by exploiting the latent structure of the sentences, the performance of the system could be increased. With the feature templates introduced, our system managed to learn structures and thus, it can classify more sentences correctly than the simple bag-of-words models can. It also shows that 10K training sentences are not sufficient for the latent method to exploit the inner structure of sentences effectively, thus it achieved even worse results than the baseline on the IT products dataset.

On the other hand, it can be seen that the baseline and our system outperformed the reference system in the case of the IT product and restaurant domains but not in the movie domain. The reason why the RNTN system performed well on the movies domain but not on the other two is that it was trained on movie reviews. This confirms the fact that it is important to train a model on a domain which is similar to the one on which it will be used. If a fully annotated treebank is available in the given domain, the supervised model is more efficient but competitive results can be achieved by employing a 10 times bigger training dataset and the proposed latent representation. A more detailed discussion of the results can be read in Section 4.5.

4.4 Target-Level Experiments

In the second batch of experiments, we investigated the utility of the latent sentiment annotation for target-level polarity classification. The task of target-level analysis is to classify sentiments which refer to a given target as in Chapter 3. We utilized the sentiment trees for target-level task by inducing the sentiment trees then extracting features from them for a target-level polarity classifier.

4.4.1 Dataset

For the evaluation of target-level classifiers we used the *SemEval-2014 Task 4 – Aspect Based Sentiment Analysis* (**ABSA**) shared task (Pontiki et al., 2014) dataset, which we introduced in Section 3.2. As a quick recap, the dataset consists of laptop and restaurant reviews annotated on target level. In the database, 4 sentiment labels were used, which were `positive`, `negative`, `neutral` and `conflict`. We did not use the `conflict` class in these experiments because of its small number of occurrences in the corpus. The resulting database consists of 2,300 laptop and 3,602 restaurant reviews, which will be referred as *absa-laptop* and *absa-restaurant*. We only used the training sets of the official datasets and ran 10-fold cross-validation to obtain our results. The reason for this decision is that in our previous experiments we noticed that the standard deviation of the accuracy among each fold and the test set is high (2.9% and 2.3% for the laptop and restaurant datasets, respectively) thus by cross-validating we obtained more robust results.

4.4.2 Latent Sentiment Tree-Based Feature Engineering

To solve the target-level problem we used a bag-of-features model with Naïve Bayes classifier. The features describing a sentence consist of word unigrams along with features derived from the predicted latent sentiment tree. We selected a subtree of the whole latent sentiment tree, with a method similar to what we introduced in Section 3.4, in order to emphasize the part of the text which is related to the target in question. This subtree is the smallest subtree which 1) contains the target mention and 2) has at least as many leaves as the quarter of the number of words in the sentence. The exact features used by the classifier are the following:

- word unigrams
- label of the sentiment subtree's root
- the label sequence on the path from the root to the target mention in the subtree
- the number of each polarity label in the above path respectively
- the collapsed label sequence on the path from the root to the target, more precisely we collapsed the consecutive equal labels, e.g., 0_A_A_C_B_B_B → 0_A+_C_B+
- the same as the last 4 features but by using the entire tree

Note that the results of this system are not directly comparable to those in Sections 3.4.3 or 3.6.2 since most of the features used there are omitted in order to highlight the effects of latent sentiment tree based features.

	absa-laptops	absa-restaurants
baseline	64.30	67.42
baseline + RNTN features	64.81	66.50
baseline + latent-tree features	67.47	69.95

Table 4.3: Accuracy scores of the target-level classifier whose feature set is enriched by sentiment-tree based features. We calculated the accuracy using 10-fold cross validation on the *absa-laptop* and *absa-restaurant* databases using the sentiment tree based features.

4.4.3 Results

The accuracy of the target-level system can be seen in Table 4.3 for both the *absa-laptops* and *absa-restaurants* databases. The *baseline* for this experiment is a simple bag-of-words model, i.e. unigrams without the sentiment tree features. The other rows in the table differ in the model used for predicting the sentiment tree for the sentences. Similar to the sentence-level task, we used the pre-trained fully supervised RNTN system for comparison. In the case of the last row, our models were trained on 100,000 sentence-level annotated IT products and restaurants datasets for the absa-laptops and absa-restaurants, respectively.

From the results it can be seen that the performance of the target-oriented system could be considerably improved by using additional features derived from the sentiment tree. The RNTN system was trained on out-domain data, thus it was useful only for the laptop dataset but not for the restaurant reviews. Since our model was trained on in-domain data, it managed to capture latent semantics of the given domain more accurately and by using the sentiment tree-based features, we managed to increase the accuracy on both target-level corpora.

4.5 Discussion

We analyzed the output of our models used in our experiments in order to reveal the reasons for accuracy differences.

The reason why our latent model can outperform the supervised RNTN system (Socher et al., 2013) lies in the **domain differences** which were used to train the systems. The RNTN system was trained on movie reviews and it performed better on the movies test corpus but worse on the other two comparing to our system which was trained on the same domain as the test data. The domain difference can be captured at the lexical level. For instance, the word *cheap* has an opposite polarity content in the IT and movie domains as it is usually positive in case we want to buy a device but negative in the case of a movie because it implies the poor quality of a film. Similarly, *fast* and *quiet* acts the same. There are some strongly IT related terms like *WiFi* or *Gigabit*, which are positive in this domain but neutral in the movies domain thus the RNTN system interprets it incorrectly on IT reviews. The restaurant domain acts similarly, there are domain specific words as

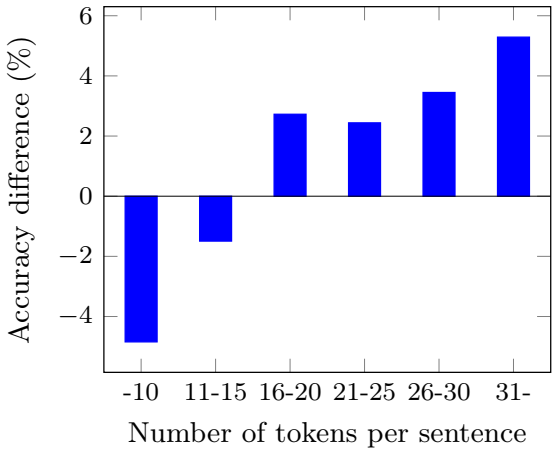

Figure 4.3: Average accuracy improvements in percentage points of the latent system over the baseline system on the restaurant test dataset depending on sentence length.

well which bear a different polarity content in the case of cuisines and otherwise.

We investigated the differences between the outputs of the baseline unigram classifier and our latent structure-based model. The only considerable explanation we found is that our in-sentence structure-based method could outperform the baseline with a **greater advance at longer sentences**. Figure 4.3 depicts the difference between the accuracies achieved by the two systems on the restaurant database depending on sentence length. Because short sentences do not contain deep syntactic structures, the sentiment tree based system cannot exploit them thus gets no advantage. As the sentences get longer and more contentful, our system gets a performance boost by better understanding the more complex meaning of these inputs.

In the case of the **target-level evaluation**, the performance increase was achieved by both the full sentiment tree and the selected subtree. In cases when only one target was present in a sentence, the correct label on the root of the sentiment tree helped the classification. As our latent model can only predict `positive` or `negative` on the root (due to the fact that it was trained using binary training data), this could not help in the case of the `neutral` label. On the other hand, when multiple targets were in a sentence, the label of the selected subtree helped the classification. We sorted the feature of the Naïve Bayes model by the absolute value of the learned feature weights. The top two features from the sentiment subtree-based features were the label `C`, which indicated the `neutral` class, and the number of each polarity label on the path from the root to the target. On the other hand, the full and the collapsed path-based label sequence features were less effective because of their sparsity. With the additional sentiment tree based features, we managed to improve the classification of the `positive` and `negative` labels on both absa datasets and in the case of the laptop domain, we increased the accuracy of the neutral class as well. The latter result is very

exciting since our latent model was trained only on `positive` and `negative` class labels.

4.6 Related Work

Fine-grained sentiment analysis methods were introduced in several studies before, using different approaches to segment sentences to sub-sentential units and classify them. Popescu and Etzioni (2005) extracted sentiment phrases from input sentences with the aim to detect product features before classifying them using sentiment lexicons. The disadvantage of this method is that it only uses the information in the extracted phrases ignoring the rest of the sentence and the relation of phrases which could lead to data sparsity. The impact of discourse relations were exploited in (Asher et al., 2009), where sentences were represented as opinion expressions. The final opinion was decided by aggregating expressions based on hand-written rules over manually annotated discourse relations. Zirn et al. (2011) segmented sentences using Rhetorical Structure Theory (Mann and Thompson, 1988), which is based on a hierarchical tree of segments, and exploited their relation for fine-grained analysis. Markov logic is used to integrate sentiment scores from different lexicons with information about relations between neighboring segments. They showed that significant improvements can be achieved by using structural features.

Several studies have been published on exploiting syntactic parsers for sentiment analysis. For instance, in (Vilares et al., 2013) a dependency parser was employed in order to detect intensifications and negations. They used hand crafted rules over dependency parses and lists of intensifiers and negation words respectively. In the Stanford Sentiment Treebank (Socher et al., 2013) a polarity label was manually assigned to each constituent of the sentence's phrase structure parse and they introduced a Recursive Neural Tensor Network-based procedure to capture the compositional effects of the sentences. Although this approach provides a more free representation for in-sentence analysis, it has the disadvantage of requiring a manually annotated treebank. To the best of our knowledge, the Stanford Sentiment Treebank is the only available sentiment-annotated treebank. This treebank is domain-dependent and the annotation of new treebanks for other domains is expensive. Our proposal is related to (Socher et al., 2013) as we also start from syntactically parsed sentences but we handle the polarity labels of internal nodes as latent variables. This way the inputs for training our system are texts annotated only on the sentence level. Our approach is similar to that of Tackström and McDonald (2011), who used document-level annotations and represented sentence-level sentiment labels as latent variables. A fine-grained approach which is based on in-sentence structures was also introduced in (Dong et al., 2015). They also propose a system which can learn in-sentence sentiment structures using exclusively sentence-level annotation. On the other hand, their system contains several hand-crafted assumptions and rules, e.g. they handle negations and intensifiers by dedicated rules, while our latent representation introduces the opportunity of exploiting the syntactic structure of sentences without restricting the models to a closed set of language phenomena, neither demands

the direct modeling of those phenomena. Another difference is that we use a syntactic parser to provide the in-sentence structure while they use a CYK sentiment parser. Although their approach provides an opportunity of learning also the structure itself, the running time is cubic hence it is not feasible to train on several hundred thousand sentences.

Beside sentiment analysis, our approach is also related to semantic parsing. Angeli et al. (2012) used latent temporal types in a latent CFG to learn temporal expressions. This semantic parsing problem is very similar to ours, since both methods require a sentence-level label in the training phase and use latent variables in the non-terminals, except the root. They assigned temporal types to non-terminal nodes, in contrast, we have polarity labels in these nodes.

We used structured perceptron for decoding the sentiment tree of an input sentence, which is successfully applied for structured prediction problems with latent variables in other areas of natural language processing as well. In (Björkelund and Kuhn, 2014) a system similar to ours was introduced for coreference resolution with latent tree structures of mention clusters. They used the passive-aggressive algorithm for this training task and updated against the highest scored tree with correct clustering of mentions.

4.7 Summary of the Thesis

In this chapter we took a step forward and analyzed texts in more detail. We focused on fine-grained sentiment analysis of texts and proposed an approach overcoming the limitations of previous work. We introduced a system which uses a latent state representation on the syntactic structure of the sentences to predict their sentiment trees. The disadvantage of previous systems are that they either rely on fine-grained annotations, which are available to only English movie reviews domain and are time and cost consuming to produce for other scenarios, or restricted by hand crafted rules which are only able to handle a pre-defined set of linguistic phenomena. Our proposed system uses only sentence-level polarity annotations for training by handling the sentiment label of sub-sentential units as a latent variable. We use an expectation-maximization based approach to infer the parameters of the latent sentiment tree decoder enabling it to learn compositional features of the language, such as negation and intensification, using only the data without requiring hand-crafted rules.

We ran two sets of experiments to prove that our system is beneficial on both sentence- and target-levels. The experimental results on the former support the fact that polarity classification is a highly domain-dependent task as the analyzers trained on out-domain sentences failed. We showed the the state-of-the-art RNTN system performs better only if in-domain fine-grained annotation is available. Our system which has access only to sentence-level annotated data outperforms the state-of-the-art model in all other domains. In practice, millions of sentence-level annotations are available for a particular domain thus our approach is applicable for training a sentiment analyzer for a new domain and it can exploit the syntactic structure of sentences.

Besides the evaluation of our approach on sentence-level, we utilized the internal structure of the sentiment trees in the target-level classification task as well. The features extracted from the sentiment trees had a considerable added value for target-level sentiment classification and we also showed that the latent sentiment trees predicted by models trained in-domain are more useful than the sentiment trees predicted by RNTN which was trained on an out-domain sentiment treebank.

Contribution

The work introduced in this chapter was a joint development with Zsolt Szántó. We would like to thank him for his invaluable cooperation in both development and execution as well. The base idea of fine-grained analysis relying only on sentence-level annotation is joint and indivisible. The work of Zsolt Szántó includes the first part of the chapter until Section 4.2.3, namely the sentiment tree representation, the latent structured decoder and the training algorithm while the merits of the author of this dissertation are the developments in the second part of this chapter starting from Section 4.2.4, i.e. tree-based feature engineering, the development of sentiment classifiers and carrying out the sentiment analysis experiments.

Chapter 5

Sentiment Analysis on Various Genres

5.1 Introduction

This introductory chapter paves the way for the second main topic of the dissertation, which focuses on the differences of texts from various sources. As we already saw in previous chapters, one problem of pre-trained sentiment analyzers is that they are produced for a specific domain or genre, which causes significant performance drop when applying the system to different texts. Here our aim is to point to the issues related to text differences. In Chapters 6 and 7 we deal with more specific problems related to this field.

An important source for getting insights about people's feelings are forum posts and product reviews sites. Texts from these sources are mostly well structured and correct in terms of spelling, which is due to that authors have more time to write coherent posts as opposed to other medias. Because of this fact, NLP tools such as syntactic or semantic parsers perform well on such texts making it easy to exploit them in end-tasks like sentiment analysis. Forum posts and product reviews are in contrast with the genre of micro-blogs posts like tweets, which are created almost in real-time so their form is less standard and they contain many more spelling errors, slang words and other out-of-vocabulary words. This causes many problems when analyzing them, usually a more intensive preprocessing is needed and the output of syntactic and semantic parsers trained on standard texts is hardly reliable on them (Foster et al., 2011).

In addition to genres, domains have their own style of word use, i.e. different phrases could be used for expressing a certain sentiment and the same phrase might convey different sentiment in different domains. For example, while the expression *very loud sound* is a positive aspect in the electronic devices domain when talking about a speaker, it is negative in the kitchen appliances

domain. The performance of a system that heavily relies on word and phrase features by learning their sentiment values based on texts coming from a given domain can significantly drop on other domains where the sentimental meaning of phrases are different. In many cases we only have access to out-of-domain training data but no in-domain data, which makes hard to build sentiment classifiers in some scenarios.

Furthermore, languages also have many differences in terms of how things are expressed, thus systems engineered to one language could achieve low performance after training them using texts from another language. Besides English, we will work with Hungarian in this chapter, which is a free word order morphologically rich language. It has several word forms due to inflections, which may mean more out-of-vocabulary and rare words. Also, a word's syntactic role is defined by its morphology, unlike English, where word order is determinative. For example, a system engineered for English text classification focusing on function words like prepositions or articles to determine the role of words in the sentence would be inappropriate for Hungarian since the role of words are determined by the suffixes of words in this language.

Due to the above, it is difficult to engineer and train one general purpose sentiment analysis system which performs well in all scenarios. The chief objective of this chapter is to investigate sentiment analysis on social media contents over various text sources and languages. We comparatively investigate texts from these sources and point to their hard aspects and suggest solutions. To achieve this, we use the document-level sentiment analysis system introduced in Chapter 3.3 and we comparatively evaluate it on publicly available English and Hungarian datasets, which contain text documents taken from Twitter and product review sites. The results of this chapter were published in (Hangya et al., 2013; Hangya and Farkas, 2017). After the investigations of this chapter we will turn to more specific domain difference related problems and their solutions in the following chapters.

5.2 Approach

In this section we detail the evaluation setup which we perform. As mentioned above, our goal is to show the issues with texts coming from various sources when using the same classifier.

5.2.1 System

As the system for sentiment analysis we use the same document-level approach introduced in Section 3.3. As a quick recall, the system involves a preprocessing and a feature engineering step. As preprocessing we apply lowercasing, stemming, deletion of punctuations and unification of certain textual elements like emoticons and URLs. As features for the classification we use n-grams, sentiment lexicon based information, the number of sentiment indicating words like character repetitions and uppercase words and we also detect the scope of negations. Note that we perform document-level classification in these experiments thus we do not use the target-level techniques from Section 3.4.

	language	genre	avg. doc. length	#labels
Amazon	EN	review	202.31	2
SemEval	EN	tweet	17.40	3
ProdRev	HU	review	9.68	2

Table 5.1: Basic statistics about the used corpora. The columns show the corpora's language, genre, the average number of words in the documents and the number of labels in the corpus.

	Positive	Negative	Neutral	Overall
Amazon	42,713	7,287	–	50,000
SemEval	4,025	1,655	5,056	10,736
ProdRev	3,573	3,149	–	6,722

Table 5.2: Label distribution and the overall number of annotated documents in each corpora.

5.2.2 Datasets

We describe the 3 datasets which are used in this chapter below. We chose to use corpora from different sources: two of them contain English texts while the last consists of Hungarian ones; the average length of the document varies; one dataset contains noisy tweets while the rest more standard forum posts. These datasets have not been used earlier in the dissertation.

- **Amazon**: Product review sites are popular for both expressing opinions about products and for gathering information about positive and negative aspects before buying one. For our experiments we sampled $50,000$ Amazon.com reviews from the dataset introduced by Jindal and Liu (2008), which are related to DVDs (movies, music, etc.) and kitchen products. Amazon has a 5-level rating scale, 1 being the worst and 5 being the best. Relying on these ratings we treated reviews which were given a 5 or 1 as positive or negative, respectively. We ignored all reviews between these two levels. Users often write long opinions about the products on Amazon thus most of the text instances in this dataset contain more than one sentences.

- **SemEval**: The organizers of *SemEval-2013 Task 2 – Sentiment Analysis in Twitter* created a database for the document-level task (Wilson et al., 2013). The corpus consists of 20,000 English Twitter messages on a range of topics. The messages were annotated with positive, negative or neutral labels depending on their global sentiments. We downloaded the disclosed training portion of it, which consisted of 10,736 messages.

- **ProdRev**: A popular Hungarian product review site is Árukereső[1]. Reviews can be found on many product categories on this site. We downloaded reviews from the PC and electronic products (TV, digital cameras, etc.) categories. The reviewers on this site have to provide pros and cons when writing a review. Here, we used these as positive and negative texts,

[1] `http://www.arukereso.hu`

respectively. Many of these reviews were very short, i.e. not a complete sentence or just a short remark. Thus, we applied filtering on the texts in such a way that we only kept reviews that contained one sentence, i.e. at least 7 token long texts having proper punctuation at their end. The resulting database consisted of 6,722 documents.

Basic statistics of the used corpora can be seen in Tables 5.1 and 5.2. In terms of average number of tokens in documents the Amazon corpus is prominent, which has multiple sentences in a document instance unlike the other two corpora. Tweets can also contain multiple sentences although segmentation is often hard due to noisiness. The Amazon and ProdRev corpora have 2 labels while SemEval is annotated with 3 types of sentiment polarities.

5.3 Results

In the following tables we show the performance of our system on the 3 corpora using accuracy and F_1 score, which were obtained using 10-fold cross-validation on each corpus.

As in Section 3.3 we use two baseline systems: *MFC* assigns the most frequent class (in the training set) to each document and *unigram baseline* is the maximum entropy classifier employing exclusively unigram features without preprocessing to assign labels to documents. We call the above-mentioned feature rich approach *document-level*. Results can be seen in Table 5.3. We indicate the performance differences of each consecutive system in parentheses. The simple supervised classifier with unigram features significantly outperforms the MFC system, which shows the advantage of machine learning based approaches even for datasets where the label distribution is highly skewed like in the case of Amazon. Using preprocessing and additional features with the document-level system we managed to further improve the results. We achieved the highest improvement for the SemEval Twitter dataset comparing to the other two, which is largely due to the effect of our preprocessing step. In the case of the other more canonical corpora, preprocessing is less effective. We ran McNemar's significance test and found that our improvements are significant at 0.05 significance level with regard to our unigram baseline with the exception of the Amazon dataset. The reason for this is that the latter contains longer texts than the others and our system is not fine-tuned for longer documents since our goal is to point to hard aspects of corpora coming from different sources. Although the improvements are not significant, our results are comparable to other studies (Vinodhini and Chandrasekaran, 2012). We also provide a label-wise F_1 score comparison to the unigram baseline and the document-level systems in Table 5.4. It can be seen that in the case of the SemEval and ProdRev datasets, the additional features helped to detect all sentiment polarities. On the other hand, the macro-averaged F_1 score decreased in the case of Amazon but we managed to improve for the positive label. This could point out that additional features for indicating negative sentiments could be beneficial.

	MFC		Unigram baseline				Document-level			
	acc	F_1	acc		F_1		acc		F_1	
Amazon	85.43	46.70	86.63	(+1.20)	74.50	(+27.80)	87.06	(+0.43)	74.23	(–0.27)
SemEval	47.09	21.35	62.29	(+15.20)	55.71	(+34.36)	63.39	(+1.10)	56.87	(+1.16)
ProdRev	53.15	34.70	89.95	(+36.80)	89.91	(+55.21)	90.76	(+0.81)	90.73	(+0.82)

Table 5.3: Results of three systems on the used corpora. The accuracy (acc) scores and macro-averaged F_1 scores were calculated using 10-fold cross-validation. Differences among systems can be seen in parentheses. The unigram baseline was compared with the most frequent class (MFC) system, while the document-level was compared with the unigram baseline.

	macro F_1	positive F_1	negative F_1	neutral F_1
Amazon	–0.27	0.30	–0.85	–
SemEval	1.16	1.40	1.25	0.83
ProdRev	0.82	0.66	0.97	–

Table 5.4: F_1 score improvements of the document-level system compared with the unigram baseline.

5.4 Discussion

In the previous section we reported quantitative results achieved by the document-level system. In the following we discuss the results as well as the representative examples in more detail.

5.4.1 Domain Differences

Lexical knowledge is a key building block of document-level systems as they basically aggregate the polarity scores of known expressions. Gathering lexical knowledge is non-trivial as it strongly depends on the domain of the training documents. As it was mentioned before, the Amazon corpus contains two domains: DVD and kitchen appliance reviews. In these domains different expressions are used to express positive and negative sentiments or even the same expression can have different sentiment polarity. For example, the following examples show the opposite polarities of the word *loud* in the DVD and kitchen domains:

John Petruccis guitar is mixed dominant in the right rear speaker, every note comes through ***loud*** *and clear.* (5.1)

I heard the same ***loud*** *noise, and felt the same funky smells, as other reviewers.* (5.2)

As such words are present in both positive and negative documents, it is problematic for the classifier to clearly decide their polarity during training time, which explains the low performance increase of the document-level system for this dataset. One way to overcome this issue is to train separate models for each domain but this can be troublesome in some scenarios. In many cases there is an insufficient amount of annotated data for a domain, especially in some languages, to train a system.

In other cases, we might not even have the domain information for the documents like in the case of the SemEval dataset, which contains tweets in many topics but they cannot be identified since they come from the same noisy channel.

In our experiments we found that features based on the sentiment lexicons prove to be very useful for shallow sentiment analysis systems. In the following example, the unigram baseline system could not learn that the words *hero* and *hopefully* are positive because of their low frequency in the training database. With the lexicon-based features, the document-level system managed to correctly classify this tweet as positive.

Khader Adnan is a hero, he's the Palestinian spring and hopefully the spark of a 3rd intifada. #KhaderExists (5.3)

Our analysis reveals that sentiment lexicon-based features are helpful in distinguishing neutral and non-neutral texts and also positive and negative ones. Table 5.4 justifies this assumption, where in the case of the SemEval corpora, which have more than two polarity labels, the F_1 score increase is higher for the positive and negative labels than for the neutral label. On the other hand, sentiment lexicons could also have incorrect sentiment polarity for some domain sensitive words. As we show in the next chapter, it is essential to use domain specific sentiment lexicons, however, they are expensive to create for each and every scenario.

5.4.2 Genre Differences

The genre of documents coming from various sources are different, meaning that their linguistic properties are diverse. The most outstanding corpus in terms of its genre is SemEval. Tweets consist of very short and simple statements and non-textual elements, e.g. emoticons and hashtags, which are used instead of longer linguistic expressions resulting in non-grammatical texts. Furthermore, due to the fast paced communication on Twitter, the number of misspelled words are relatively high. Consider the following tweet:

@princess_saraw: @Lady_Li @jt23lfc haha he thinks ill clean n cook even when im dying #womens jobs ha x damn straight!!xx (5.4)

It contains spelling errors (*im*), slang words (*n*), user mentions (*@Lady_Li*), hashtags (#womens) and other out-of-vocabulary words (*ha*); furthermore, the sentences are not well separated. After analyzing our document-level results, we can conclude that preprocessing is one of the most important steps on the Twitter corpus while its impact on performance is lower for the other two corpora.

Although both the Amazon and ProdRev datasets come from the product review genre, their lengths are different. The Amazon reviews are relatively long texts, where people tend to express their opinions about a particular product in more detail. One might think that due to the more detailed and longer texts, it is easier to determine the sentiment of these reviews. However, in many cases the reviewer writes about both positive and negative aspects of the product and concludes at

the end. It is also possible that no conclusion is given, which makes it even harder to detect the final sentiment of the text. For example:

The specifications of the device are good. Also the price isn't the lowest, which indicates that this should not be the worst device ever. But I regret the purchase. (5.5)

Since our system calculates the rate of positive and negative features (and detects negations as well), this example was classified incorrectly. A solution to this problem could be to analyze the sentiment of each sentence individually and use the aggregated values as the document's sentiment. Zhang et al. (2009) introduced a system where they aggregated the sentence level sentiment values based on several features. They showed that sentences which are at the beginning or at the end of a document and the ones which are in first person are more important during the aggregation. In the case of the ProdRev dataset, where the texts are one sentence long, this issue is not present since the reviews are more concise.

5.4.3 Language Differences

Languages differ in many linguistic aspects beyond their vocabulary. In English, where grammatical roles are expressed by word order, there exist only a handful of morphological variations for words. On the other hand, in Hungarian, grammatical roles are defined using word suffixes, thus the language has a free word order and rich morphology. In our experiments, we found that the main challenge of adapting a sentiment classification system to a new and typologically very different language is to find the appropriate lexical representation. In the case of Hungarian, due to its morphological richness, a word can appear in many forms, which implies that in the training phase, word forms are not seen as frequently as in English where there are not so many different forms. One solution to this problem is to use lemmas instead of word forms but it has drawbacks as well. Consider the following example and its lemmatized form:

IOS jobb mint az Android (IOS good-COMP as the Android) "IOS is better than Android" (5.6)
IOS jó mint az Android (IOS good as the Android) "IOS is as good as Android"

There is a negative sentiment related to the target *Android*, but if the lemmatized sentence is used, the classifier would wrongly classify it as positive because by lemmatizing *jobb (better)* to *jó (good)*, we lose information. This indicates that it is not straightforward to just retrain a system with data from a language different to the one which was used during its development. We note that there are other aspects of languages that can differ but in the dissertation we do not intend to give a complete analysis and leave it for future work.

5.5 Related Work

Similarly to our work in this chapter, Remus (2014) analyzed domain and genre differences with the goal of supporting the development of robust sentiment analysis approaches that work well and in a predictable manner under different conditions. Since most works dealing with domain, genre or language differences aim at alleviating occurring problems, we cite related studies in the following that worked on similar problems mentioned in the previous sections. We present our work for specific problems as well as their related work in the following chapters.

The issue of domain differences is widely studied not only for sentiment analysis but for other natural language processing and machine learning tasks as well. Many simple approaches were proposed by the field of domain adaptation to exploit out-of-domain data better, like using the large annotated data from the source domain and the small annotated data from the target side jointly or weighting the importance of the target side data more. A good comparison of general domain adaptation methods can be found in (Daume, 2007). Blitzer et al. (2007) performed domain adaptation for sentiment analysis by measuring domain similarities in order to look for domains that can be exploited more easily for a given target domain.

The problem of multiple sentence long document classification was also studied by many researchers before. Two main types of approaches were developed: cascaded and joint model. In case of the first one, the sentiment of each sentence is classified and aggregated to one document-level label as a second step. In (Pang and Lee, 2004) each sentence was classified as either subjective and objective as a first step and only the subjective sentences were used as input for a second classifier to decide the global document-level polarity. Similarly, Zhang et al. (2009) employed different sentence-level features to determine which sentences should be used when classifying the whole document. They found that sentences at the beginning and end of the documents as well as those where the subject is in first person are more likely to contain sentiments of the author. Joint models combine sentence- and document-level classification in one model. A conditional random fields based method was proposed by McDonald et al. (2007) to sequentially output the sentiment of consecutive sentences in a document and a global polarity for the whole review as the last step. Although joint models tend to outperform the cascaded approaches, they are often hard to train since they require sentence and document level annotations.

As we saw earlier, sentiment analysis on the Twitter genre was studied by many researchers. Yearly shared tasks were organized in order to encourage research in this field (Wilson et al., 2013; Rosenthal et al., 2014, 2015; Nakov et al., 2016; Rosenthal et al., 2017). Furthermore, an in-depth analysis of the impact of preprocessing on tweet classification was performed by Krouska et al. (2016). Product reviews are often annotated with stars by the authors thus it is easy to convert them to sentiment labels and use them as additional training data. As Twitter does not have such mechanism, it is more difficult to get a large amount of annotated data without costly manual labor. On the other hand, emoticons are often used in tweets which can indicate sentiment. In

(Go et al., 2009) positive and negative emoticons were used as a noisy source of automatic labels in order to leverage additional data during training. It was shown that by using distant supervision, performance could be increased using this additional data.

Most studies dealing with sentiment analysis for multiple languages focus on the question of how to use data from a resource rich source language to improve performance on a resource poor target language. Using machine translation to translate source language training data to train a classifier on the target side or to translate test data to the source language is a straightforward method to overcome language barriers. In (Wan, 2008) and (Brooke et al., 2009) it was shown that this approach gives good results even in the case of distant language pairs like English and Chinese. On the other hand, machine translation is only applicable if large parallel data are available for the given language pair, which is often not the case for resource poor languages. More recently, bilingual word embeddings were proposed which can be used for cross-lingual transfer learning. Upadhyay et al. (2016) empirically compared methods for bilingual word embeddings requiring different levels of bilingual signal and show that these resources are useful for overcoming the language barrier.

A couple of authors published their work in the field of opinion mining and sentiment analysis specifically for Hungarian. Machine learning based approaches were employed for detecting opinions of users in their posts in (Berend and Farkas, 2008). A rule-based system was developed by the *OpinHu* project (Miháltz, 2010) for various languages, including Hungarian, using sentiment lexicons. The same authors compiled the *OpinHuBank* corpus for training Hungarian sentiment classifiers (Miháltz, 2013), which was also used in the dissertation as well as in (Hangya et al., 2015). In addition, a corpus annotated on aspect-level is also available for Hungarian (Szabó et al., 2016). For further work on Hungarian natural language processing, we point the reader to the annual conference in the field, called *Magyar Számítógépes Nyelvészeti Konferencia*.

5.6 Summary of the Thesis

Social media provides a big amount of user-generated text in a wide variety of domains, genres and languages. The analysis of these opinionated texts is important in many areas, which explains why sentiment analysis has become a popular research area. In this chapter we performed document-level sentiment analysis on three corpora coming from various sources. We comparatively analyzed our results in more detail and discussed the hard aspects of sentiment classification on texts with different domains, genre and language.

On the Twitter corpora first we applied a preprocessing step to unify Twitter specific notations. We found that this step is very important to decrease the number of unknown and infrequent word forms thus improving performance of the classifier. With the document-level system we extracted shallow information from the documents that may indicate the polarity of the texts. These features were used in a supervised machine learning-based system to classify documents into sentiment

polarity classes. We found that among the document-level features, those based on the sentiment lexicon were the most useful. With these features we managed to distinguish positive and negative sentiments more effectively. On the other hand, domain differences can cause issues when using these lexicons. Furthermore, we looked at the problems of the Hungarian language coming from its morphological richness. We showed examples to emphasize the reasons why our system developed for English is not directly applicable to Hungarian by training it using annotated Hungarian texts.

In summary we can conclude that the accuracy which can be expected from a state-of-the-art sentiment analyzer is highly dependent on:

- Diversity of the domain: lexical knowledge from corpora containing texts from multiple domains could be hard to acquire since words can have different sentiment polarity in different domains,
- Genre: various genres use various grammatical complexity and they could require special preprocessing steps,
- Length of the documents: long texts can contain mixed sentiments where the conclusion could be hard to decide while short texts are more to the point but could be too concise,
- Language of the texts: syntax and semantics can be expressed differently thus systems engineered for one language could perform lower for other languages.

Keeping these findings in mind we turn our focus to the problem of the usage of sentiment lexicons under different domains and to the problem of domain adaptation for low resource languages in the following chapters.

Chapter 6

Domain Specific Sentiment Lexicons

6.1 Introduction

In the previous chapters we showed that good performance gain can be achieved both for sentence- and target-level by using features based on sentiment lexicons. These lists, containing words together with their sentiment value, can be considered as an external knowledge for the classification task. Although lexicons are important resources, in order to achieve good performance it is essential to use knowledge from a domain that has similar properties as the domain in which we apply the classification system. In this chapter we focus on domain differences of sentiment lexicons and propose methods for creating domain specific ones.

Good indicators for the sentiment of a given text are the words in it that have positive or negative meanings. When a big amount of annotated data is available, supervised machine learning based approaches are able to learn this word-level information from the data. On the other hand, this method can be inaccurate if the size of the annotated corpus is insufficient because many word types are not frequent enough while others are not even present in the data. To overcome the lexical sparsity problem, sentiment lexicons which contain a predefined set of positive and negative words are useful. This knowledge can be used to extract various features from sentences besides n-grams, e.g. the overall number of positive and negative words.

Many general purpose sentiment lexicons are available for English (Baccianella et al., 2010; Wilson et al., 2005). The problem with these lexicons is that they were created based on general domain knowledge, which can be problematic when using them in some domains since some words may have different polarities in different domains. Consider the following examples:

*The usage of this mixer is easy and it is very **silent**.* (6.1)

*For this price it's too **silent** for me, I thought it will be louder.* (6.2)

The first example is from the domain of kitchen devices, where *silent* has a positive meaning. In contrast, the second example is from the speakers domain, where *silent* is a negative quality. This shows that it is not possible to create one ultimate lexicon which has the right word polarities in all cases, thus choosing lexicons from the appropriate domain is important. Keeping in mind that the number of text domains is basically endless and that the creation of sentiment lexicons is time consuming and expensive, automatic methods are needed to acquire the right lexicon for a given use case. In this chapter we present methods for creating and adapting sentiment lexicons automatically for a given domain. We propose language independent techniques for creating lexicons automatically. By incorporating texts from a given domain, we create lexicons from scratch which are useful for extracting features for sentiment analysis tasks. We also propose semi-automatic methods for lexicon creation by using a small number of sentiment seed words.

We evaluate lexicons by employing them in sentiment analysis tasks. We experiment on Hungarian texts, where the number of available lexicons is even lower. We use texts from two domains, one is a domain specific corpus which contains reviews about IT products, while the other contains general texts from news (Mihálta, 2013) related to various topics, like sports, politics, etc. We compare the efficiency of sentiment lexicons from different domains and show the importance of using domain-specific sentiment lexicons for different sentiment analysis use cases. The results of this chapter were published in (Hangya, 2015).

6.2 Sentiment Lexicon Creation

In this section we describe our proposed approaches for sentiment lexicon creation and adaptation. Three main approaches exist for creating sentiment lexicons: manual, dictionary-based and corpus-based (Liu, 2012). The manual approach needs good domain knowledge to select words with the right sentiment value and it is time-consuming thus expensive. Dictionary-based approaches rely on word relation knowledge such as synonym and antonym relation of words. Approaches use this knowledge to automatically collect sentiment words starting from a manually created seed lexicon. These approaches are more light-weight in terms of manual work, i.e. the initial seed lexicon which is a small set of positive and negative words can be assembled easily. On the other hand, it requires word relation information which can cause additional work if not already available. The creation of such information could be time consuming as well (Miller, 1998) since language expertise is needed. Automatic approaches producing lower quality resources (Lam et al., 2014) exist as well. In the case of the corpus-based approaches, knowledge regarding the sentiment of words is extracted from a set of documents, which makes them easily applicable in most cases.

6.2.1 Lexicon Translation

For comparison, in our experiments we use a lexicon manually translated from English to Hungarian. This method results a good quality lexicon but has some disadvantages as well. Although the translation is faster than creating a lexicon from scratch, it is still time-consuming, thus expensive especially if the original lexicon is big. Translating polysemous words can be difficult as well, due to the fact that it is unclear which meaning to use. For example, the word *terrific* has two meanings with opposite polarities, namely *awesome* and *horrible*. By using the polarity value from the original lexicon, the correct translation can be guessed, but not in all cases. The word *cool* (*cold* and *awesome*) also has two meanings where both can be positive but with different intensity. Most of the existing lexicons are for general use, so during the translation process we have to consider the domain in which the translated lexicon will be used and in the case of some words the original polarity value have to be altered.

We had access to an English lexicon already used by an in house reputation monitoring system, which we translated to Hungarian. The translated lexicon contained 3322 word forms each with its polarity level from the $[-5, 5]$ interval, where -5 is the most negative value and 5 is the most positive. The translation was carried out manually by one translator. The translation was made independent of any domain, meaning that we chose the most frequent translations of the English words and that the original sentiment values were not changed, thus resulting a general sentiment lexicon.

6.2.2 Bootstrapping Sentiment Lexicons

To overcome the above mentioned problems, we implemented methods which can automatically create sentiment lexicons. The first one is corpus-based which exploits a document set annotated with sentiment labels. The annotation can be done in various ways. The most accurate one is manual annotation. It can be done automatically as well, using an existing sentiment analysis system, hence the name bootstrapping. For example, this system can be a simple n-gram based model trained on a text from another genre or domain. The automatic method can yield a noisy annotation but a large amount of data filters noise. By annotating a large amount of data, the sentiment polarity of words that were not seen by the initial classifier during training could be decided. To calculate word polarities, we use pointwise mutual information (Turney and Littman, 2003):

$$pol(w) = PMI(w, positive) - PMI(w, negative) \tag{6.1}$$

where PMI is the pointwise mutual information for a given word w and sentiment polarity which can be calculated with the following equation:

$$PMI(w, p) = log_2 \frac{freq(w, p) * N}{freq(w) * freq(p)} \tag{6.2}$$

where $p \in \{positive, negative\}$ is the given polarity, $freq(w, p)$ is the frequency of the occurrence of word w in texts with polarity p, $freq(w)$ and $freq(p)$ are the overall frequency of word w and the number of texts with polarity p in the corpus, respectively, and N is the number of different word forms in the vocabulary. The method gives a polarity value for all words in the corpus which reflects the positiveness and negativeness of the given word in that domain. Additionally, we scale these values into the $[-5, 5]$ interval. In the following, we will refer to lexicons created with this method with the name **pmi**.

6.2.3 Extending Seed Lexicons

In this section we propose two dictionary-based methods for extending an initial seed lexicon. The input seed lexicon of these methods contains only a low number of words with their polarity values. Such an initial lexicon is easy to build due to its size by either manual or automatic methods. For our experiments we created a semi-automated method to acquire a seed lexicon. We trained a simple n-gram based sentiment analysis system with maximum entropy classifier on the training portion of the given corpus. Using the trained model, it is possible to extract those words that are most likely to occur in positive and negative texts, respectively. We manually filtered out noisy words which had high probability and used the top 20 words for both polarity classes respectively to form the seed lexicon. We scaled the polarity values of the seed words into the $[-5, 5]$ interval based on the weights of the model corresponding to these words.

Our extension methods are based on relations of words which we detail below. More precisely, we add words to the extended lexicons which are in relation with those which are already in the seed lexicon. By using word relations which reflect the aspects of a domain, we not only extend the input lexicon but also adapt it to the given domain.

WordNet

In our first extension approach we rely on word relations found in a given wordnet. Wordnets are large lexical databases which contain words grouped into sets of synonyms (synsets) (Miller, 1998). Synsets are linked by means of conceptual-semantic and lexical relations. Our hypothesis is that the sentiment polarity of a word and all of its synonyms is nearly equal. The extension process is as follows.

1. Initially each synset has a polarity value of 0.

2. We iterate over all words in the seed lexicon and assign the actual seed word's polarity value to those synsets in which it appears. Additionally, we use the relation between synsets, namely which sets have similar or opposite meanings. For this we used the *similar_to* and *hyponym* relations in the wordnet. We assign the polarity value or its inverse of a synonym set to all

related synsets depending on the relation type. If a synset is related to multiple synsets with non 0 polarity value, we calculate their average.

3. In the last step, we add all the words to the extended lexicon which are in a synset with a polarity value different than 0 using this value as the word's polarity value. A word type can be in multiple synsets with different polarity values. For instance, the word *terrific* is included in the following positive and negative synsets {*wonderful*, terrific, fantastic} and {*terrifying*, terrific}, where the seed words are *wonderful* and *terrifying*. In such cases we use the average of these polarity values as the word's polarity value.

The method can be run iteratively. The output of a step can be used as the seed lexicon of the next one further expanding the lexicon in each step. An important fact is that some words can be added to the extended lexicon with wrong polarity values in an iteration step. For example, in the IT domain if the word *silent* is used as a positive seed word, the word *uncommunicative* will be added as positive to the extended lexicon, which does not have any polarity in this domain. Because of this, after some iteration step the extended lexicon becomes too noisy. Furthermore, wordnets are general lexical resources, thus the extracted word similarities are not domain dependent. We note that domain specific wordnets exist, but only for a handful of domains and languages. For this reason it is important to start with a seed lexicon which is already domain-specific, this way the extension is aware of the specifics of a given domain. For our experiments we used the Hungarian WordNet (Miháltz et al., 2008).

Word Clusters

The disadvantage of the previous method is that WordNets could be unavailable for some low-resource languages. Furthermore, as it was mentioned above, most WordNets are not domain specific resources. To overcome these issues, we developed a method for building word relations which highlight domain specialties while only requiring non-annotated data. For this we use the Brown clustering algorithm (Brown et al., 1992). It is a hierarchical clustering of words based on the context in which they occur. The input of this method is a seed lexicon as before and an unlabeled corpus from a given domain. Similar to the previous method, our hypothesis is that the polarities of words connected by some relation, i.e. which are in the same cluster, are nearly equal. We build clusters on the unlabeled dataset and perform the following steps:

1. The initial step of the algorithm is to assign 0 polarity value to all clusters.

2. We iterate over all words in the seed lexicon and assign the polarity value of the actual seed word to the cluster which contains it. If a cluster contains multiple seed words, we calculate their average value.

3. Lastly, we add words with the appropriate polarity value to the extended lexicon which are in a cluster with non 0 value.

The method has one parameter which is the number of clusters to use. As the Brown clustering algorithm gives a hierarchy of clusters, one way to decide the number of clusters is to cut the hierarchy at a certain level and each subtree below this level forms a cluster. If we use a small number of clusters, words which are not similar can be in the same cluster, which causes that the extension assigns wrong polarity values to some words. Inversely, if we use too many clusters, just a small number of new words is added to the new lexicon. The main advantage of this method – in contrast with the wordnet based one – is that the clustering algorithm which uses domain-specific texts can capture word similarities which are specific to the given domain. This way it is more effective for domain-adapting the input lexicon.

6.3 Evaluation Setup

The goal of this work was to create methods to automatically assemble sentiment lexicons which are useful in sentiment analysis tasks. To comparatively evaluate lexicons, we perform sentiment analysis in which we classify sentences into positive and negative classes and the used system was strongly built upon the lexicons. We define the usefulness of a lexicon given a corpus with the accuracy of the classifier system which uses that lexicon. The higher the accuracy, the more useful the lexicon is.

6.3.1 Datasets

In this section we present the used corpora. For our experiments we used two Hungarian databases: one with texts from general and one from IT domain. We note that both corpora were used previously in Chapters 3 and 5 respectively, thus we briefly mention their main aspects here.

General domain The *OpinHuBank* (Miháltz, 2013) is a corpus created directly for sentiment analysis and contains texts from a general domain using various Hungarian news sites, blogs and forums about sports, politics, economics, etc. The sentences were annotated with positive, negative or neutral labels. For the experiment in this chapter we ignored neutral sentences and used only positive and negative instances. This way we got 882 positive and 1629 negative sentences. Note that this dataset was annotated on target level. We ignored the target information, by performing sentence-level analysis, which could cause some noise in the annotation. For example, the sentence *I cried on the back seat of my BMW!*, where *BMW* is the target, would be negative in the sentence-level scenario. However, it is neutral in the target-level case because the negative sentiment is not related to BMW. Since our goal in this chapter is not to build the best performing sentiment

Sentence	The laptop's display has **better** parameters!
Lemmatized unigrams	the, laptop, display, have, good, parameter, !
sentiment words	better
Overall values	POSITIVE=3.5, NEGATIVE=0.0, POLARITY=3.5
Neighbors	has_POSITIVE, POSITIVE_parameter

Table 6.1: An example sentence and the features extracted from it. The sentiment word in the sentence is *better*, which has 3.5 polarity value.

classifier but to comparatively analyze the quality of sentiment lexicons, it does not invalidate our experiments. We refer to this dataset as **OpinHu** henceforward in this chapter.

IT specific domain As the domain specific dataset, we used the corpus from the Hungarian site called *árukereső*. The data contains reviews about electronic products. The dataset consists of $3,573$ positive and $3,149$ negative instances. We refer to this corpus as **ProdRev**.

6.3.2 Sentiment Classifier

As the sentiment analysis system, we employed a fairly simple system which was strongly built on features based on the sentiment lexicons in order to show their quality. We used a maximum entropy classifier with lemmatized unigrams and the lexicon based features detailed below. In the following we consider a word as *sentiment word* if it is included in the given lexicon and its absolute polarity value is at least 1. A sentiment word is positive or negative depending on the sign of its polarity value. The lexicon based features are the following (an example of all features can be seen in Table 6.1):

- the sentiment words in the text (in their original form)
- the overall values of positive and negative words, respectively
- the overall values of sentiment words
- pairs made of the polarity of a sentiment word and its preceding/following lexical neighbor

6.4 Results

In the following we present the accuracy of our classifier by performing 10-fold cross-validation. The results of the system using the lexicon extending techniques can be seen in Table 6.2. In the case of both OpinHu and ProdRev databases we created a seed lexicon with the semi-automatic method which was presented earlier. The notation *wn* indicates the usage of the wordnet-based

lexicon	acc
OpinHu-seed	86.2
OpinHu-seed wn-1	86.4
OpinHu-seed wn-2	85.9
OpinHu-seed wn-3	86.3
OpinHu-seed wn-4	86.0
OpinHu-seed cluster-15	86.7
OpinHu-seed cluster-15 t3	86.8

lexicon	acc
ProdRev-seed	90.7
ProdRev-seed wn-1	90.8
ProdRev-seed wn-2	90.5
ProdRev-seed wn-3	90.8
ProdRev-seed wn-4	90.9
ProdRev-seed cluster-18	90.8
ProdRev-seed cluster-19 t3	90.8

Table 6.2: Extension of seed lexicon with wordnet (wn) and cluster based methods. The accuracies on OpinHu and ProdRev corpora were measured using 10-fold cross-validation.

lexicon	acc	
	OpinHu	ProdRev
OpinHu-baseline	86.1	70.1
ProdRev-baseline	61.6	90.0
OpinHu-seed cluster-15 t3	86.8	90.1
ProdRev-seed wn-4	86.2	90.9
translated	88.4	90.2
OpinHu-pmi	96.3	90.0
ProdRev-pmi	84.3	91.9
ProdRev2-pmi	-	91.0

Table 6.3: Achieved accuracies using different lexicons on OpinHu and ProdRev. Data used for training baseline systems, initializing seed or pmi lexicons are indicated in the first column.

word relations and the number after that gives the number of iterations we ran. In the case of the OpinHu corpus, we achieved the highest increase in accuracy with 1 iteration while in the case of ProdRev 4 iterations proved to be the best. In both cases, after the 5^{th} iteration the lexicons became too noisy and the results begun to decrease.

In the last two rows of the tables, the results of the clustering based extension method can be seen. The number at the end of the lines shows the level where the cluster hierarchy was cut, which was tuned on a small held out dataset, and *t3* indicates that we filtered out words from the lexicon which have a frequency of at most 3 in the corpus. This technique performed better in the case of the OpinHu corpus, and comparably in the case of ProdRev compared to the wordnet based extension.

In Table 6.3, the results of the baseline systems which used only lemmatized unigrams as features can be seen for both corpora, along with the best extended lexicons, the bootstrapped (*pmi*) lexicons and the manually *translated* lexicon. Two baseline systems were created, the first was trained on the OpinHu corpus and the second on ProdRev (the used training datasets are indicated in the first column; all other systems were trained using in-domain data). The results show that the system not being trained on the same domain as the test corpus resulted in a significantly lower accuracy score. Furthermore, it can be seen that an increase can be achieved with the extending techniques compared to the baselines if the lexicon is in the appropriate domain, otherwise this increase is much smaller.

The *translated* lexicon caused 2.3% increase in case of the OpinHu corpus and only 0.2% in case of the ProdRev, which is less than the effect of the extended lexicons in the case of the latter. The reason for this is that OpinHu contains more general texts and the lexicon which was translated is for general purpose. The IT specific ProdRev corpus benefits less from a non domain specific sentiment lexicon.

The last 3 rows of Table 6.3 show the results for the bootstrapped lexicons. The prefix of each line indicates the annotated corpus which was used to create the lexicon. In those cases where the corpus used for the creation of the lexicon is the same as the corpus on which the sentiment analysis system was evaluated, the results show a theoretical maximum. This maximum shows the accuracy which can be achieved if a perfect lexicon for that corpus is available. It can be seen that these lexicons are not useful for the other domains as they can even decrease the results such as in the case of the ProdRev-pmi lexicon on the OpinHu corpus.

We created a lexicon for the IT domain using similar texts contained in the ProdRev dataset. For this, we created the *ProdRev2* corpus, which consists of those positive and negative reviews from the árukereső site that are not one sentence long, i.e. they are shorter or longer. Using this to create a lexicon we managed to outperform the lexicons based on the extension methods.

6.5 Related Work

One of the most important indicators of sentiment in a given text are sentiment words (Liu, 2012). In the early years of sentiment analysis research many systems were based on lexicons, i.e. counting the ratio of positive and negative words in texts. The Sentiment Orientation CALculator uses lexicons with sentiment words, intensifiers and negators to classify texts (Taboada et al., 2011). Due to their popularity, many projects started to create good quality general purpose lexicons. A list of 6,800 positive and negative words were created by Hu and Liu (2004a) over many years. In the SentiWordNet positivity, negativity and objectivity scores were assigned to synsets of the English WordNet (Baccianella et al., 2010). The work is based on quantitative analysis of the glosses associated to the synsets. The MPQA (Multi-Perspective Question Answering) Subjectivity Lexicon is a list of subjective clues that can be used for analysis (Wilson et al., 2005). In the WordStat Sentiment Dictionary, word patterns are annotated (Pollach, 2011). A lot of work has been done in the field for English but the number of available lexicons are smaller for other languages. Hungarian, on which our experiments were performed, is a low-resource language in terms of sentiment lexicons. In (Szabó, 2014) a Hungarian lexicon was created based on an English resource. Beside translating existing lexicons, a different approach was followed in (Chen and Skiena, 2014) where lexicons for 136 major languages were created. A variety of linguistic resources from multiple languages were used to produce an immense knowledge graph and propagating information from an English lexicon to create sentiment lexicons for each language.

All of the above works indicate the need for automatically creating sentiment lexicons. Two main approaches exist (Liu, 2012), namely the dictionary-based and the corpus-based ones. The former method is based on linguistic relation of words. Mohammad et al. (2009) exploited many antonym-generating affix patterns like X and $disX$, e.g. *honest–dishonest*, to increase the coverage of a given lexicon. Wordnet was used to determine the distance of adjectives in (Kamps et al., 2004). The polarity of a given adjective was calculated based on the distance of the adjective and two reference points. Corpus-based approaches use texts from a given domain to create lexicons. In one of the earliest work, a corpus and some seed adjectives were used for extension by applying rules, e.g. if two adjectives are conjoined, their polarity should be similar. The propagation was based on the hypothesis that neighboring sentences have similar sentiment orientation in (Kanayama and Nasukawa, 2006). Since these methods strongly rely on the corpus in a given domain, the resulting lexicons are also domain specific. On the other hand, it was shown that further work is required because word polarity could also depend on the context besides the domain. Ding et al. (2008) use pairs of words and aspects, and determines sentiment words and their orientations together with the aspects that they modify.

6.6 Summary of the Thesis

Sentiment lexicons are important resources to improve performance of classifiers especially in low-resource setups. They can be considered as external knowledge, which can be used for feature engineering during sentiment classification. On the other hand, they have to be used with caution since words may have different polarities in different domains. Most of the available lexicons are created based on general domain knowledge, thus they do not fit all use cases. There is a need to create domain specific lexicons which is time consuming and expensive due to the number of possible domains and the required domain knowledge.

In this chapter we focused on how to create sentiment lexicons automatically, which are useful for sentiment analysis. We presented a technique to create lexicons from scratch by using either annotated or non-annotated texts. We proposed methods for extending an initial seed lexicon by propagating sentiment polarity values based on word relation information. To build the input seed lexicons we proposed a semi-automatic method but we argue that these lexicons can be easily created manually as well if no annotated data is available. In addition, we proposed a method that creates sentiment lexicons using statistical information in annotated data. For this we used point-wise mutual information.

Although our proposed methods are language independent, we ran experiments on Hungarian corpora due to its lack of sentiment lexicons. Our results empirically underpin that it is important to use lexicons which are aware of the specificities of the domain by achieving better results with an automatic lexicon compared to the manually created one on the IT domain. We also showed that

by using a lexicon from a different domain the results can even decrease. Although we achieved an increase in accuracy with the automatically created lexicons in the general domain, the best results were given by the manually assembled (and translated) lexicon. From this we can conclude that the manually created lexicons are better in quality, but they are much more expensive and it is hard to create one for all domains, thus automatic methods are needed. The results show that the proposed automatic methods are useful for increasing the performance of sentiment analysis systems in all domains.

Chapter 7

Cross-Lingual Domain Adaptation for Sentiment Analysis

7.1 Introduction

Recent advancements of neural networks made it possible to create systems which can understand language better. With the ability to learning sophisticated feature templates without the need of their manual engineering, neural networks are able to detect high level language phenomena. Using these techniques, the performance of sentiment classifiers have also been increased. An important technique which made this performance increase possible is word embeddings, i.e. vector representations of words, which allows us to measure the similarity of word meanings. In a way this resource can be considered similar to sentiment lexicons since by using them we can incorporate external knowledge to our system. Word embeddings for a given language are easy to build due to the requirement of only a large unlabeled corpus. It is also possible to build bilingual word embeddings (BWEs) where words from a source and a target language are embedded in the same vector space by representing source and target words which have similar or the same meaning with similar vectors. Such resources allow us to perform bilingual transfer learning where only source language data is used to train a model which is then applied on the target language. Bilingual tasks are crucial for overcoming data sparsity in the target language as we also showed in the previous chapter for Hungarian. On the other hand, resources required for such tasks are often out-of-domain, thus domain adaptation is an important problem here as well.

In this chapter we propose two methods for domain adaptation of bilingual tasks. Previously, task specific approaches were proposed. In contrast, our methods are task and language independent. We show experimentally that a simple adaptation process of BWEs involving only unlabeled text is highly effective. We then show that a semi-supervised classification method from computer vision

can be applied successfully for further gains in cross-lingual classification.

Our BWE adaptation method is delightfully simple. We begin by adapting monolingual word embeddings (MWEs) to the target domain for source and target languages by simply building them using both general and target-domain unlabeled data. As a second step we use post-hoc mapping to transform the word embeddings of the two languages into the same vector space (Mikolov et al., 2013b). We show experimentally for the first time that the domain-adapted bilingual word embeddings we produce using this technique are highly effective. In previous work, task-dependent approaches were used for this type of domain adaptation. Our approach is simple and task independent.

Second, we adapt the semi-supervised image classification system of Häusser et al. (2017) for NLP problems. This approach is broadly applicable to many NLP classification tasks where unlabeled data are available. The system exploits unlabeled data during the training of classifiers by learning similar features for similar labeled and unlabeled training examples, thereby extracting information from unlabeled examples as well. As we show experimentally, the system further improves cross-lingual knowledge transfer for our tasks.

Although our focus is on sentiment analysis in this thesis, we study two quite different tasks and domains, where resources are lacking, showing that our methods are universal and perform well. We show results for cross-lingual sentiment classification (CLSC) of tweets. In addition, we also present results for bilingual lexicon induction (BLI) on the medical domain. After combining both of our techniques, the results of sentiment analysis are competitive with systems that use annotated data in the target language, an impressive result considering that we require no target-language annotated data. The method also yields impressive improvements for bilingual lexicon induction compared with baselines trained on in-domain data. We show that high-quality domain-adapted bilingual word embeddings are required in order to use unlabeled data well by the semi-supervised approach. The results of this chapter were published in (Hangya et al., 2018).

7.2 Domain Adaptation Approaches

In this section we introduce the two proposed approaches and show task specific setups and results after.

7.2.1 Adaptation of Bilingual Word Embeddings

BWEs trained on *general domain* texts usually result in lower performance when used in a system for a *specific domain*. There are two reasons for this. (i) Vocabularies of specific domains contain words that are not used in the general case, e.g., names of medicines or diseases. (ii) The meaning of a word varies across domains; e.g., "apple" mostly refers to a fruit in general domains, but it is an electronic device in many product reviews. Figure 7.1 depicts the domain adaptation problem of

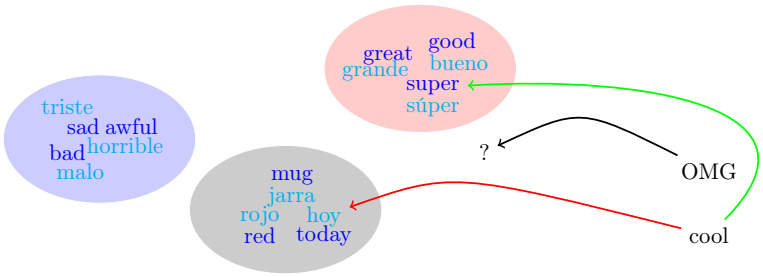

Figure 7.1: Domain adaptation of BWEs. Red, gray and blue word clusters correspond to words with positive, neutral and negative sentiment, respectively. The polarity of word *cool* is not clear because it depends on the target domain. *OMG* is a domain specific word and was not seen in general domains.

BWEs for sentiment classification.

The delightfully simple method adapts general domain BWEs in a way that preserves the semantic knowledge from general domain data and leverages *monolingual* domain specific data to create domain-specific BWEs. Our domain-adaptation approach is applicable to any language pair in which monolingual data are available. Unlike other methods, our approach is task independent: it only requires unlabeled in-domain target language text.

To create domain adapted BWEs, we first train MWEs in both languages and then map those into the same space using post-hoc mapping (Mikolov et al., 2013b). We train MWEs for both languages by concatenating monolingual out-of-domain and in-domain data. The out-of-domain data allows us to create accurate distributed representations of common vocabulary while the in-domain data embeds domain specific words. We then map the two MWEs using a small seed lexicon to create the adapted BWEs. As post-hoc mapping only requires a seed lexicon as bilingual signal, it can easily be used with (cheap) monolingual data.

For **post-hoc mapping**, we use Mikolov et al. (2013b)'s approach. This model assumes a $W \in \mathbb{R}^{d_1 \times d_2}$ matrix which maps vectors from the source to the target MWEs, where d_1 and d_2 are the embedding space dimensions. A seed lexicon of $(x_i, y_i) \in L \subseteq \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ pairs is needed where x_i and y_i are source and target MWEs. W can be learned using ridge regression by minimizing the L_2-regularized mapping error between the source x_i and the target y_i vectors:

$$\min_W \sum_i ||Wx_i - y_i||_2^2 + \lambda ||W||_2^2 \tag{7.1}$$

where λ is the regularization weight. Based on the source embedding x, we then compute a target embedding as Wx.

We create MWEs with word2vec skipgram (Mikolov et al., 2013a)[1] and estimate W with *scikit-learn* (Pedregosa et al., 2011). We use the default parameters of these tools.

[1] `https://github.com/dav/word2vec`

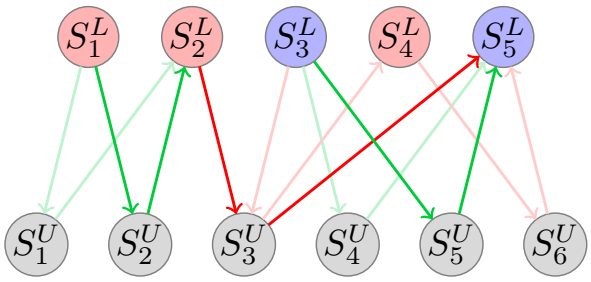

Figure 7.2: Walking cycles of the semi-supervised approach. Red and blue nodes illustrate labeled samples with 2 possible labels while gray depicts unlabeled ones. Green and red arrows show correct and incorrect cycles, respectively while opacity shows step probabilities.

7.2.2 Semi-Supervised Learning

In addition to the BWE-adaptation method, we investigate ways to further incorporate unlabeled data from the target domain to overcome data sparsity. Häusser et al. (2017) introduce a semi-supervised method for neural networks that makes associations from the vector representation of labeled samples to those of unlabeled ones and back. This lets the learning exploit unlabeled samples as well. While Häusser et al. (2017) use their model for image classification, we adapt it to NLP tasks. We show that our semi-supervised model requires adapted BWEs to be effective and yields significant improvements. This innovative method is general and can be applied to any classification task when unlabeled text is available.

Häusser et al. (2017)'s basic assumption is that the embeddings of labeled and unlabeled samples, i.e. the representations in the neural network on which the classification layer is applied, are similar within the same class. To achieve this, walking cycles are introduced: a cycle starts from a labeled sample, goes to an unlabeled one and ends at a labeled one. A cycle is correct if the start and end samples are in the same class. The probability of going from sample A to B is proportional to the cosine similarity of their embeddings. Figure 7.2 illustrates the idea of the approach. To maximize the number of correct cycles, two loss functions are employed: Walker loss and Visit loss. In the following, we briefly introduce the used loss functions and refer to (Häusser et al., 2017) for more details.

Walker loss penalizes incorrect walks and encourages a uniform probability distribution of walks to the correct class. It is defined as:

$$\mathcal{L}_{walker} := H(T, P^{aba}) \tag{7.2}$$

where H is the cross-entropy function, P_{ij}^{aba} is the probability that a cycle starts from a labeled sample i and ends at a labeled sample j and T is the uniform target distribution:

$$T_{ij} := \begin{cases} \frac{1}{\#c(i)} & \text{if } c(i) = c(j) \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

where $c(i)$ is the class of sample i and $\#c(i)$ is the number of occurrences of $c(i)$ in the labeled set.

Visit loss encourages cycles to visit all unlabeled samples, rather than just those which are the most similar to labeled samples. It is defined as:

$$\begin{aligned} \mathcal{L}_{visit} &:= H(V, P^{visit}) \\ P_j^{visit} &:= \langle P_{ij}^{ab} \rangle_i \\ V_j &:= \frac{1}{U} \end{aligned} \tag{7.4}$$

where H is the cross-entropy function, P_{ij}^{ab} is the probability that a cycle starts from labeled sample i and goes to an unlabeled sample j and U is the number of unlabeled samples.

Classification loss is necessary to learn to map representations of samples to class labels. For this, we use a fully connected feed-forward layer as the last layer of our network and calculate the cross-entropy between predicted and gold labels:

$$\mathcal{L}_{class} := -\sum_i y_i * log(\hat{y}_i) \tag{7.5}$$

where y_i is a vector of zeros except having 1 at the position corresponding to the gold label of sample i, $\hat{y}_i$ is a vector containing the predicted probabilities of each possible label and $log(\cdot)$ is the element-wise natural log operation.

The total loss during training is the weighted sum of the walker, visit and classification losses which is minimized using Adam (Kingma and Ba, 2015).

We adapt this model to sentiment classification and bilingual lexicon induction. We call these systems **semisup**[2]. Due to the fact that we initialize the embedding layers for both classifiers with BWEs, the models are able to make some correct cycles at the beginning of the training and improve them later on. We describe the used task specific systems and the labeled/unlabeled datasets in the subsequent sections below.

We use Häusser et al. (2017)'s implementation of the losses, with 1.0, 0.5 and 1.0 weights for the walker, visit and classification losses respectively for CLSC based on preliminary experiments. We fine-tuned the weights for BLI on the development set for each experiment.

[2]Our implementation can be found publicly at: `https://github.com/hangyav/biadapt`

7.3 Cross-Lingual Sentiment Classification

To show the performance of our proposed methods on sentiment classification we run experiments on the Twitter domain. As resource rich source language we use English data and evaluate on Spanish, which we simulate as resource poor.

In CLSC, an important application of BWEs, we train a supervised sentiment model on training data available in the source language and apply it to the resource poor target language, for which there is typically no or not adequate amount of training data available. Because BWEs embed source and target words in the same space, annotations in the source (represented as BWEs) enable transfer learning. The trained model can then be applied to the target language. For CLSC of tweets, a drawback of BWEs trained on non-Twitter data is that they do not produce embeddings for Twitter-specific vocabulary, e.g. slang words like English *coool* and Spanish *chido*, resulting in lost information when a sentiment classifier uses them.

In the following, we introduce the used data in our experiments and the neural network architectures for the classification.

7.3.1 Datasets

Training Data for Twitter Specific BWEs As comparable non-Twitter data, we use OpenSubtitles (Lison and Tiedemann, 2016), which contains 49.2M English and Spanish subtitle sentences, respectively (**Subtitle**). The reason behind choosing Subtitles is that although it is out-of-domain, it contains slang words similar to tweets thus serving as a strong baseline in our setup. We experiment with two monolingual Twitter datasets:

(i) **22M_tweets**: Downloaded[3] English (17.2M) and Spanish (4.8M) tweets using the public *Twitter Streaming API*[4] with language filters *en* and *es*.

(ii) A **BACKGROUND** corpus of 296K English and 150K Spanish (non-annotated) tweets released with the test data of the RepLab task (Amigó et al., 2013).

All Twitter data was tokenized using Bird et al. (2009) and lowercased. User names, URLs, numbers, emoticons and punctuation were removed.

As lexicon for the mapping, we used the BNC word frequency list (Kilgarriff, 1997), a list of 6,318 frequent English lemmas and their Spanish translations, obtained from Google Translate. Note that we do not need a domain-specific lexicon in order to get good quality adapted BWEs.

Training Data for Sentiment Classifiers For sentiment classification, we use data from the *RepLab 2013* shared task (Amigó et al., 2013), which was already introduced and used in Chapter 3.

[3]We downloaded for a month starting on 2016-10-15.
[4]`dev.twitter.com/streaming/overview`

As a quick recap, the data is annotated with positive, neutral and negative labels and contains English and Spanish tweets. Here we used the official English training set (26.6K tweets[5]) and the Spanish test set (14.9K) in the resource-poor setup. We only use the 7.2K Spanish labeled training data for comparison reasons in Section 7.4.2, which we will discuss later. The shared task was on target-level sentiment analysis. The reason for using this dataset instead of a sentence-level corpus is that it contains comparable English and Spanish tweets annotated for sentiment. There are other Twitter datasets for English (Nakov et al., 2016) and Spanish (García-Cumbreras et al., 2016), but they were downloaded at different times and were annotated using different annotation methodologies, thus impeding a clean and consistent evaluation.

7.3.2 Systems

For evaluating our adapted BWEs on the *RepLab* dataset, we used a **target-aware** sentiment classifier introduced by Zhang et al. (2016). The network first embeds input words using pre-trained BWEs and feeds them to a bi-directional gated neural network. Pooling is applied on the hidden representations of the left and right contexts of the target mention, respectively. Finally, gated neurons are used to model the interaction between the target mention and its surrounding context. During training, we held our pre-trained BWEs fixed and kept the default parameters of the model.

We also implemented Kim (2014)'s *CNN-non-static* system, which does not use the target information in a given document (**target-ignorant**). The network first embeds input words using pre-trained BWEs and feeds them to a convolutional layer with multiple window sizes. Max pooling is applied on top of convolution followed by a fully connected network with one hidden layer. We used this system as well because it performed comparably to the target-aware system. The reason for this lies in the simplicity of this system. Since the target-aware network has more parameters, it also requires more training data to find the right weights of the network. We used the default parameters as described in (Kim, 2014) with the exception of using 1000 feature maps and 30 epochs, based on our initial experiments. Word embeddings are fixed during the training just as for the target-aware classifier.

7.4 Results

7.4.1 Embedding Adaptation

As we previously explained, we evaluate our adaptation method on the task of target-level sentiment classification using both target-aware and target-ignorant classifiers. For all experiments, our two baselines are the off-the-shelf classifiers using non-adapted BWEs, i.e. BWEs trained only using Subtitles. Our goal is to show that our BWE adaptation method can improve the performance of

[5]Note that the number of instances is slightly lower compared to Chapter 3 due to the unavailability of some tweets during re-downloading for these experiments.

		target-	
		aware	ignorant
oracle	MWE_Subtitle	62.17	63.27
	BWE_Subtitle	62.46	63.50
domain adaptation	Baseline	55.14	59.05
	BACKGROUND	56.79	58.50
	22M_tweets	59.44	61.14
	Subtitle+BACKGROUND	58.64	59.34
	Subtitle+22M_tweets	60.99	61.06

Table 7.1: Accuracy of the BWE adaptation approach on the target-level sentiment classification task. The oracle systems used Spanish sentiment training data instead of English.

such classifiers. We trained our adapted BWEs on the concatenation of Subtitle and 22M_tweets or BACKGROUND, respectively. In addition, we also report results with BWEs trained only on tweets.

To train the sentiment classifiers, we used the English RepLab training set and we evaluated on the Spanish test set by presenting the accuracy of the systems. To show the performance that can be reached in a monolingual setup, we report results obtained by using annotated Spanish sentiment data instead of English. We call this setup oracle since it shows the performance of the used systems in a non resource-poor setup. We trained two oracle sentiment classifiers using (i) MWEs trained on only the Spanish part of Subtitle and (ii) BWEs trained on Subtitle using posthoc mapping. The difference between the two is that the embeddings of (ii) are enriched with English words, which can be beneficial for the classification of Spanish tweets because they often contain a few English words as well.

We do not compare with word embedding adaptation methods relying on specialized resources. The point of our work is to study task-independent methods and to the best of our knowledge ours is the first such attempt. Similarly, we do not compare against machine translation based sentiment classifiers, e.g. (Zhou et al., 2016), because for their adaptation, in-domain parallel data would be needed.

Table 7.1 gives the results for both classifiers. It shows that the adaptation of Subtitle based BWEs with data from Twitter (22M_tweets and BACKGROUND) clearly outperforms the Baseline in all cases. The target-aware system performed poorly with the baseline BWEs and could benefit significantly from the adaptation approach. The target-ignorant one performed better with the baseline BWEs but could also benefit from the adaptation. Comparing results with the Twitter-dataset-only based BWEs, the 22M_tweets performed better even though the BACKGROUND dataset is from the same topic as the RepLab train and test sets. Our conjecture is that the latter is too small to create good BWEs. In combination with Subtitles, 22M_tweets also yields better results than when combined with BACKGROUND. Although the best accuracy was reached using the 22M_tweets-only based BWEs, it is only slightly better than the adapted Subtitles+22M_tweets

		semisup
domain adaptation	Baseline	58.67 (-0.38)
	BACKGROUND	57.41 (-1.09)
	22M_tweets	60.19 (-0.95)
	Subtitle+BACKGROUND	60.31 (0.97)
	Subtitle+22M_tweets	63.23 (2.17)

Table 7.2: Accuracy on CLSC of the adapted BWE approach with the semisup (target-ignorant with additional loss functions) system compared to the target-ignorant in brackets.

based BWEs. In the following section we show that both the semantic knowledge from Subtitles and the domain-specific information from tweets are needed to further improve results.

Comparing the two classifiers we can say that they performed similarly in terms of their best results. On the other hand, the target-ignorant system had better results on average. This might seem surprising at first because the system does not use the target as information. As we argued before, the reason is that the target-aware system needs more training samples to be trained efficiently.

Although we did not focus on the impact of the seed lexicon size, we ran post-hoc mapping with different sizes during our preliminary experiments. With 1,000 and 100 word pairs in the lexicon, the target-ignorant system suffered 0.5% and 4.0% drop on average of our setups, respectively.

7.4.2 Semi-Supervised Method

As before, we used pre-trained BWEs to initialize the classifier and employed the English sentiment training data as the labeled set in our semi-supervised experiments. Furthermore, we used the Spanish sentiment training data as the unlabeled set, ignoring its annotation. This setup is very similar to real-word low-resource scenarios: unlabeled target-language tweets are easy to download while labeled English ones are available. We used the target-ignorant system with the additional *walker* and *visit* losses as the *semisup* system.

Table 7.2 gives results for adapted BWEs and shows that semisup helps only when word embeddings are adapted to the Twitter domain. As mentioned earlier, semisup compares labeled and unlabeled samples based on their vector representations. By using BWEs based on only Subtitles, we lose too much domain specific information, thus the embeddings of similar English and Spanish tweets are not close enough. On the other hand, if we use only tweet-based BWEs, we lose good quality semantic knowledge which can be learned from more standard text domains. By combining the two domains we were able to capture both sides. For Subtitle+22M_tweets, we even got very close to the best oracle (BWE_Subtitle) in Table 7.1 getting only 0.27% less accuracy – an impressive result keeping in mind that we did not use labeled Spanish data.

The RepLab dataset contains tweets from 4 topics: automotive, banking, university, and music. We manually analyzed similar tweets from the labeled and unlabeled sets. We found that when using semisup, English and Spanish tweets from the same topics are more similar in the embedding

		lang	target-aware	target-ignorant
oracle	MWE_Subtitle	Es	62.17	63.27
	BWE_Subtitle	Es	62.46	63.50
domain adaptation	Subtitle+BACKGROUND	En	58.64	59.34
	Subtitle+BACKGROUND	En+Es	64.01	62.92 (2.61)
	Subtitle+22M_tweets	En	60.99	61.06
	Subtitle+22M_tweets	En+Es	64.24	63.82 (0.59)

Table 7.3: Accuracy on CLSC of both target-aware and target-ignorant systems using English or/and Spanish labeled sentiment training data. Column *lang* shows the language of the used training data. Differences compared to semisup are indicated in brackets where available.

space than without the additional losses. Topics differ in how they express sentiment, which may explain why semisup increases performance for RepLab.

Adding supervision

To show how effectively semisup can exploit the unlabeled data, we used both English and Spanish labeled sentiment training data together to train the sentiment classifiers. Table 7.3 shows that by using annotated data in both languages, we get clearly better results than when using only one language. Tables 7.2 and 7.3 show that for Subtitle+22M_tweets based BWEs, the semisup approach achieved high improvement (2.17%) compared to target-ignorant with English training data only, while it achieved lower improvement (0.97%) with the Subtitle+BACKGROUND based BWEs. On the other hand, adding labeled Spanish data caused just a slight increase compared to semisup with Subtitle+22M_tweets based BWEs (0.59%), while in the case of Subtitle+BACKGROUND, we got significant additional improvement (2.61%). This means that with higher quality BWEs, unlabeled target-language data can be exploited better.

It can also be seen that the target-aware system outperformed the target-ignorant system using additional labeled target-language data. The reason could be that by using the additional data, the architectural advantage of the target-aware system compared to the target-ignorant could be exploited.

The results in Table 7.3 are impressive: our target-level system is strongly competitive with the official shared task results. We achieved high accuracy on the Spanish test set by using only English training data. Comparing our best system which used all training data to the official results (Amigó et al., 2013), we would rank 2^{nd} even though our system is not fine-tuned for the RepLab dataset. Furthermore, we also outperformed the oracles when using annotated data from both languages, which shows the additional advantage of using BWEs.

7.5 Medical Bilingual Lexicon Induction

In this chapter we apply our proposed adaptation methods to an auxiliary task to show the universality of our approaches. BLI is another interesting downstream task for BWEs, which is useful for many applications such as machine translation. Given a list of words in a source language, the goal of BLI is to mine translations for each word in a chosen target language. The medical bilingual lexicon induction task proposed in (Heyman et al., 2017) aims to mine medical words using BWEs trained on a very small amount of English and Dutch monolingual medical data. Due to the lack of resources in this domain, good quality BWEs are hard to build using in-domain data only. We show that by enriching BWEs with general domain knowledge, better results can be achieved on this medical domain task.

7.5.1 Experimental Setup

We evaluate our adaptation methods on the dataset provided by Heyman et al. (2017). The monolingual medical data consists of English and Dutch medical articles from Wikipedia. The English (Dutch) articles contain 52,336 (21,374) sentences. In addition, a total of 7,368 manually annotated word translation pairs occurring in the English (source) and Dutch (target) monolingual corpora are provided as gold data. This set is split 64%/16%/20% into train/development/test sets. In addition, 20% of the English words have multiple translations. Given an English word, the task is to find the correct Dutch translation.

As monolingual general-domain data we use the English and Dutch data from Europarl (v7) (Koehn, 2005), a corpus of 2 million sentence pairs. Although Europarl is a parallel corpus, we use it in a monolingual way and shuffle each side of the corpus before training. By using massive cheap data, we create high-quality MWEs in each language which are still domain-specific (due to the inclusion of medical data). To obtain an out-of-domain seed lexicon, we translated the English words in BNC to Dutch using Google Translate (just as we did before for the Twitter CLSC task). We then used the out-of-domain BNC and the in-domain medical seed lexicons in separate experiments to create BWEs with post-hoc mapping. Note that we did not concatenate the two lexicons because (i) they have a small common subset of source words which have different target words, thus having a negative effect on the mapping and (ii) we did not want to modify the medical seed lexicon because it was taken from previous work.

7.5.2 Bilingual Lexicon Induction Systems

To perform BLI we employed two methods. Because BWEs represent words from different languages in a shared space, BLI can be performed via *cosine similarity* in this space. In other words, given a BWE representing two languages V_s and V_t, the translation of each word $s \in V_s$ can be induced by taking the word $t \in V_t$ whose representation x_t in the BWE is closest to the representation x_s.

	cosine similarity		classifier	
	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
Baseline medical lexicon	13.43	9.84	37.73	36.61
Baseline BNC lexicon	–	–	20.73	21.78
Adapted medical lexicon	14.18	14.15	40.71	38.09
Adapted BNC lexicon	16.29	16.71	22.10	21.50

Table 7.4: Results (F_1) for medical BLI with the cosine similarity and the classifier based systems. We present baseline and our proposed domain adaptation method using both general (BNC) and medical lexicons.

As the second approach we used a *classifier based system* proposed by Heyman et al. (2017). This neural network based system is comprised of two main modules. The first is a character-level LSTM which aims to learn orthographic similarity of word pairs. The other is the concatenation of the embeddings of the two words using embedding layers with the aim of learning the similarity among semantic representations of the words. Dense layers are applied on top of the two modules before the output soft-max layer. The classifier is trained using positive and negative word pair examples and a pre-trained word embedding model. Negative examples are randomly sampled by picking a random non-translation pair for the English words in the training lexicon. We used default parameters as reported by Heyman et al. (2017) except for the t classification thresholds which is used at prediction time to decide if a candidate word pair is positive. We fine-tuned these on the development set. We note that the system works with pre-trained MWEs as well (and report these as official baseline results) but it requires BWEs for candidate generation at prediction time, thus we use BWEs for the system's input for all experiments. In preliminary work, we had found that MWE and BWE results are similar.

7.5.3 Results

Embedding Adaptation

Heyman et al. (2017)'s results are our *baseline*. Table 7.4 compares its performance with our adapted BWEs, with both cosine similarity and classification based systems. We show F_1 scores where "top" F_1 scores are based on the most probable word as prediction only, while "all" F_1 scores use all words as prediction whose probability is above the threshold. It can be seen that the cosine similarity based system using adapted BWEs clearly outperforms the non-adapted BWEs which were trained in a resource poor setup.[6] Moreover, the best performance was reached using the general seed lexicon for the mapping, which is due to the fact that general domain words have better quality embeddings in the MWE models, which in turn gives a better quality mapping.

The classification based system performs significantly better compared to cosine similarity by

[6]The results for cosine similarity in (Heyman et al., 2017) are based on BWESG-based BWEs (Vulić and Moens, 2016) trained on a small document aligned parallel corpus without using a seed lexicon.

	F_1 (top)	F_1 (all)
Baseline+BNC	35.04 (-0.66)	34.98 (-1.40)
Baseline+medical	36.20 (0.50)	36.55 (0.16)
Adapted+BNC	41.01 (0.30)	39.04 (0.95)
Adapted+medical	41.44 (0.73)	37.51 (-0.57)

Table 7.5: Results with the semi-supervised system for BLI. Differences compared to previous results are indicated in brackets. Baseline results are compared to rerun experiments of Heyman et al. (2017) using BWEs instead of MWEs. We use the medical labeled lexicon in all cases.

exploiting the seed lexicon better. Using adapted BWEs as input word embeddings for the system, further improvements were achieved, which shows the better quality of our BWEs. Simulating an even poorer setup by using a general lexicon, the performance gain of the classifier is lower. This shows the significance of the medical seed lexicon for this system. On the other hand, adapted BWEs have better performance compared to non-adapted ones using the best translation while they have just slightly lower F_1 using multiple translations. This result shows that while with adapted BWEs the system predicts better "top" translations, it has a harder time when predicting "all" due to the increased vocabulary size.

Semi-Supervised Method

For BLI experiments we used the classifier of Heyman et al. (2017) with the additional losses as the *semisup* system. We used word pairs from the medical seed lexicon as the labeled set (with negative word pairs generated as described in Section 7.5.2). As opposed to CLSC and the work of (Häusser et al., 2017), for this task we do not have a straightforward unlabeled set, and therefore we need to generate it. We developed two scenarios. For the first, **BNC**, we generated a general unlabeled set using English words from the BNC lexicon and generated 10 pairs out of each word by using the 5 most similar Dutch words based on the corresponding BWEs and 5 random Dutch words. For the second scenario, **medical**, we generated an in-domain unlabeled set by generating for each English word in the medical lexicon the 3 most similar Dutch words based on BWEs and for each of these we used the 5 most similar English words (ignoring the words which are in the original medical lexicon) and 5 negative words. The idea behind these methods is to automatically generate an unlabeled set that hopefully has a similar positive and negative word pair distribution to the distribution in the labeled set.

Results in Table 7.5 show that adding semisup to the classifier further increases performance for BLI as well. For the baseline system, when using only in-domain text for creating BWEs, only the medical unlabeled set was effective, general domain word pairs could not be exploited due to the lack of general semantic knowledge in the BWE model. On the other hand, by using our domain adapted BWEs, which contain both general domain and in-domain semantic knowledge, we can exploit word pairs from both domains. Results for adapted BWEs increased in 3 out of 4 cases, where the only

exception is when using multiple translations for a given source word, which may have been caused by the bigger vocabulary size.

These results show that adapted BWEs are needed to exploit unlabeled data well, which leads to an impressive overall 3.71 increase compared with the best result in previous work (Heyman et al., 2017), by using only unlabeled data.

7.6 Previous Work

Bilingual Word Embeddings

Many approaches have been proposed for creating high quality BWEs using different bilingual signals. Following Mikolov et al. (2013b), many authors (Faruqui and Dyer, 2014; Xing et al., 2015; Lazaridou et al., 2015; Vulić and Korhonen, 2016) map monolingual word embeddings (MWEs) into the same bilingual space. Others leverage parallel texts (Hermann and Blunsom, 2014; Gouws et al., 2015) or create artificial cross-lingual corpora using seed lexicons or document alignments (Vulić and Moens, 2015; Duong et al., 2016) to train BWEs.

In contrast, our aim is not to improve the intrinsic quality of BWEs, but to adapt BWEs to specific domains to enhance their performance on bilingual tasks in these domains. Faruqui et al. (2015), Gouws and Søgaard (2015), and Rothe et al. (2016) have previously studied domain adaptation of bilingual word embeddings, showing it to be highly effective for improving downstream tasks. However, importantly, their proposed methods are based on specialized domain lexicons (such as, e.g. sentiment lexicons) which contain task specific word relations. Our delightfully simple approach is, in contrast, effectively task independent (in that it only requires unlabeled in-domain text), which is an important strength.

Cross-Lingual Sentiment Analysis

Sentiment analysis is widely applied, and thus ideally we would have access to high quality supervised models in all human languages. Unfortunately, good quality labeled datasets are missing for many languages. Training models on resource rich languages and applying them to resource poor languages is therefore highly desirable. Cross-lingual sentiment classification tackles this problem (Mihalcea et al., 2007; Banea et al., 2010; Wan, 2009; Lu et al., 2011; Balamurali and Joshi, 2012; Gui et al., 2013). Recent CLSC approaches use BWEs as features of deep learning architectures, which allows us to use a model for target-language sentiment classification, even when the model was trained only using source-language supervised training data. Following this approach, we performed CLSC on Spanish tweets using English training data.

Xiao and Guo (2013) proposed a cross-lingual log-bilinear document model to learn distributed representations of words, which can capture both the semantic similarities of words across languages

and the predictive information with respect to the classification task. Similarly, Tang and Wan (2014) jointly embedded texts in different languages into a joint semantic space representing sentiment. Zhou et al. (2014) employed aligned sentences in the BWE learning process, but in the sentiment classification process only representations in the source language are used for training, and representations in the target language are used for predicting labels. An important weakness of these three works was that aligned sentences were required.

Some work has trained sentiment-specific BWEs using annotated sentiment information in both languages (Zhou et al., 2015, 2016), which is desirable, but this is not applicable to our scenario. Our goal was to adapt BWEs to a specific domain without requiring additional task-specific engineering or knowledge sources beyond having access to plentiful target-language in-domain unlabeled text. Both of the approaches we study in this work fit this criterion, the delightfully simple method for adapting BWEs and the broadly applicable semi-supervised approach of Häusser et al. (2017) can improve the performance of any off-the-shelf classifier that is based on BWEs.

Bilingual Lexicon Induction

BLI is an important task that has been addressed by a large amount of previous work. The goal of BLI is to automatically extract word translation pairs using BWEs. While BLI is often used to provide an intrinsic evaluation of BWEs (Lazaridou et al., 2015; Vulić and Moens, 2015; Vulić and Korhonen, 2016), it is also useful for tasks such as machine translation (Madhyastha and España Bohnet, 2017). Most work on BLI using BWEs focuses on frequent words in high-resource domains such as parliament proceedings or news texts. Recently, Heyman et al. (2017) tackled BLI of words in the medical domain. This task is useful for many applications such as terminology extraction or OOV mining for machine translation of medical texts. Heyman et al. (2017) show that when only a small amount of medical data is available, BLI using BWEs tends to perform poorly. Especially BWEs obtained using post-hoc mapping (Mikolov et al., 2013b; Lazaridou et al., 2015) fail on this task. Consequently, Heyman et al. (2017) build BWEs using aligned documents and then engineer a specialized classification-based approach to BLI. In contrast, our delightfully simple approach to create high-quality BWEs for the medical domain requires only monolingual data. We showed that our adapted BWEs yield impressive improvements over non-adapted BWEs in this task with both cosine similarity and with the classifier of Heyman et al. (2017). In addition, we showed that the broadly applicable method can push performance further using easily accessible unlabeled data.

7.7 Summary of the Thesis

Bilingual word embeddings trained on general domain data yield poor results in out-of-domain tasks. In this chapter we proposed two approaches for domain adaptation in bilingual setups. Our delightfully simple task independent method to adapt BWEs to a specific domain uses unlabeled

monolingual data only. We showed that with the support of adapted BWEs, the performance of off-the-shelf methods can be increased for cross-lingual Twitter sentiment classification. Furthermore, by adapting the broadly applicable semi-supervised approach of Häusser et al. (2017), which until now has only been applied in computer vision, we were able to effectively exploit unlabeled data to further improve performance. We showed that, when also using high-quality adapted BWEs, the performance of the semi-supervised systems can be significantly increased by using unlabeled data at classifier training time. Furthermore, CLSC results are competitive with a system that uses target-language labeled data, even when we use no such target-language labeled data. In addition, we applied our proposed methods to bilingual lexicon induction on the medical domain. Our results proved the task and domain independence of our approaches.

Contribution

This work was conducted during the author's visit at the Center for Information and Language Processing (CIS) at the University of Munich (LMU). We would like to thank Dr. Fabienne Braune, Prof. Dr. Alexander M. Fraser and Prof. Dr. Hinrich Schütze for their invaluable cooperation in the development of this work. The two proposed methods for domain adaptation were developed by the author of this dissertation with the help of invaluable ideas of all authors of the published work of (Hangya et al., 2018). The experiments related to bilingual lexicon induction was performed by Dr. Fabienne Braune. The experiments on cross-lingual sentiment analysis and the application of semi-supervised method on the classifier based BLI system were carried out by the author of the dissertation.

Chapter 8

Summary

8.1 Summary in English

The work done in the dissertation focused on pushing the boundaries of sentiment analysis techniques. Various natural language processing techniques were introduced starting from specific preprocessing steps through feature engineering till different ways and levels of analysis. Most of the approaches were based on the more traditional manual feature engineering used with linear classifiers, moreover, techniques based on neural networks were also introduced.

The dissertation dealt with two main topics: target-level sentiment analysis (Chapters 3 and 4) and domain/genre differences (Chapters 5, 6 and 7), which we summarize in the following.

8.1.1 Target-Level Sentiment Analysis

Chapter 3 introduced the task of target-level sentiment classification where instead of the detection of global sentiments in texts the task is to classify only those sentiments which are related to a given target. This application is often employed by various entities such as product owners or public figures in order to monitor users' sentiments towards their products or actions. Various difficulties have to be overcome when developing such systems, e.g. detecting relevant parts of texts in case it contains sentiments related to different targets. The author proposed a competitive system for target-level analysis and evaluated on multiple genres and languages. In order to reach good performance on tweets, a Twitter specific preprocessing was introduced. In order to make a document-level classifier target-aware, various target-level features were introduced. For detecting relevant parts of input texts, both surfaceform- and syntax-based methods were proposed. The competitiveness of the system was shown on both Twitter and more standard genres and also using Hungarian texts besides English. In addition, task-specific systems were also shown which – when participating in shared-tasks – ranked among the best systems reaching 1^{st} place in case of the RepLab 2013

shared task. Related to his publications (Hangya and Farkas, 2013a,b, 2017), the author regards the following results as his main contributions to the field:

- Surfaceform-based target-level feature engineering for emphasizing relevant content on less well formed texts;
- Syntax-based feature engineering, using both dependency and constituency parses, for target specific feature extraction on more standard texts;
- A hybrid system using both of the above techniques reaching best overall performance and its error analysis on various datasets.

8.1.2 Fine-Grained Sentiment Analysis

Chapter 4 focused on the syntactic structure of sentences. By looking at the main building blocks of sentences and how they are connected, we can understand their meaning better. Fine-grained sentiment analysis aims to predict a sentiment value for each constituent of the input sentences in a bottom-up fashion. Previous approaches for this task showed that this approach can improve the sentence-level performance but they used either expensive annotated data that is only accessible for a restricted set of domains and languages or language specific rules that are also hard to adapt to other use cases. The author together with his colleagues proposed a latent syntactic structure based system relying only on sentence-level annotations and general feature templates making the system easily applicable to any languages and domains. It was shown that the proposed system improves document-level classification. Although the system having access to fine-grained annotations obtained better performance, it was shown that the proposed method performs better on domains where such data is not accessible. Furthermore, the proposed system was exploited for the target-level sentiment classification task, by extracting target-specific features from the generated sentiment trees, resulting performance improvement for this task as well. Related to his publication (Hangya et al., 2017), the author regards the following results as his main contributions to the field:

- Introduction of latent syntactic structure-based fine-grained sentiment analysis method using only sentence-level annotations as opposed to previous work;
- Improvement of document-level sentiment classification by exploiting the sentences' syntactic structure;
- Incorporating fine-grained analysis into target-level classification by proposing sentiment tree-based features.

8.1.3 Sentiment Analysis on Various Genres

Chapter 5 is an introduction to the second main topic of the dissertation. Domain differences represent a huge problem in general for machine learning. Systems developed and trained in one set of input might not perform well in the case of domain shift. The author comparatively analyzed sentiment classification on document-level English and Hungarian datasets coming from different domains and genres. As a basis for the next chapters, he pinpointed main issues when applying systems developed and/or trained on different data then the test set. Related to his publications (Hangya et al., 2013; Hangya and Farkas, 2017), the author regards the following result as his main contribution to the field:

- In-depth quantitative and qualitative comparison of document-level sentiment analysis on various domains, genres and languages.

8.1.4 Domain Specific Sentiment Lexicons

Chapter 6 focused on sentiment lexicons which are key external resources for sentiment analyzers. By using lexicons to extract features from input texts, significant performance increase can be achieved. On the other hand, the sentiment polarity of words often depends on the domain on which the classifier is applied. Since creating sentiment lexicons is time consuming and expensive, most of them are general purpose resources. The author showed that a good quality lexicon could have negative effects on the performance if used in the wrong domain. To overcome this limitation, automatic methods were proposed to create domain specific lexicons. It was shown that by extending a cheap domain specific seed lexicon using different lexical resources or by building one from scratch relying on annotated data, better performance can be reached compared to the case when using out of domain manual lexicons. Since most lexicons are available in English, the author showed experiments for Hungarian sentiment analysis. Related to his publication (Hangya, 2015), the author regards the following results as his main contributions to the field:

- Language independent automatic methods for creating domain-specific sentiment lexicons: extending a small seed lexicon and building from scratch using annotated data;
- Showing that domain specific lexicons are crucial in order to achieve good performance on Hungarian datasets.

8.1.5 Cross-Lingual Domain Adaptation for Sentiment Analysis

Chapter 7 introduced the developed methods for domain adaptation in cross-lingual sentiment classification. Word embeddings together with neural networks achieved good performance in various sentiment classification scenarios recently. In addition, cross-lingual transfer learning made it possible to build models using annotations from a resource rich language and apply it to languages where

no annotations are available. Although these methods work well in many scenarios, they often have low quality when the training and test data are from different domains. The author together with his colleagues showed that bilingual word embeddings built using general domain resources are lacking in-domain specific words and represent the incorrect meaning of others. To overcome this issue, a simple domain adaptation method was proposed relying only on monolingual unlabeled data in contrast to other works. In addition, a second approach was proposed based on semi-supervised learning which can incorporate further domain specific unlabeled data to improve cross-lingual knowledge transfer. The performed experiments showed that the adaptation of embeddings is crucial for the second step and that the proposed methods are task and language independent. Related to his publication (Hangya et al., 2018), the author regards the following results as his main contributions to the field:

- Simple, yet effective method for domain adaptation of bilingual word embeddings;
- Semi-supervised method for incorporating unlabeled task specific target language/domain data to improve cross-language knowledge transfer;
- Domain adaptation of cross-lingual tasks using unlabeled data only.

8.2 Magyar nyelvű összefoglaló

A disszertáció célja az aktuális szentimentelemző technikák határainak kiterjesztése volt. Különböző természetesnyelv-feldolgozó technológiákat ismertettünk kezdve a szentiment specifikus előfeldolgozási lépésekkel, jellemző kinyerési technikákon át egészen a szentiment elemzés különböző szintjeiig. A legtöbb bemutatott módszer a hagyományosabbnak tekinthető jellemzőkinyerés alapú lineáris osztályozókra épült, de neurális hálózatra épülő technikákat is bemutattunk.

A disszertáció két főbb témát dolgoz fel, melyeket lentebb foglalunk össze: célorientált szentimentelemzés a 3. és 4. fejezetekben, valamint domén- és műfajkülönbségek a 5., 6. és 7. fejezetekben.

8.2.1 Célorientált szentimentelemzés

A 3. fejezetben a célorientált szentimentelemzést mutattuk be, mely esetében a globális érzelmek felderítése helyett adott célra vonatkozó szentimentek osztályozása a feladat. A módszert sokan alkalmazzák, mint például termékek gyártói vagy ismert személyiségek annak érdekében, hogy a felhasználóik vagy követőik véleményét könnyebben megismerhessék. A feladat megoldása során számos nehézséggel szembe kell nézni, mint például a kérdéses célra vonatkozó szövegrészek detektálása több célt tartalmazó mondatok esetében. A szerző egy versenyképes célorientált szentimentelemző rendszert dolgozott ki, melyet különböző stílusú és nyelvű szövegeken értékelt ki. Annak érdekében, hogy magas pontosságot érjen el tweeteken, egy Twitter-specifikus előfeldolgozó módszert mutatott be. Különböző célorientált jellemzőket dolgoztunk ki, melyekkel dokumentumszintű rendszerek célorientáltá tehetők. A szövegek releváns részeinek detektálásához felszíni jellemzőkön és szintaxison alapuló módszerek születtek. A rendszer eredményeit Twitterről származó és sztenderdebb angol és magyar nyelvű szövegeken mutattuk be. Ezenfelül versenyfeladatokra fejlesztett specifikus rendszereket is ismertettünk, melyek a legjobb rendszerek között szerepeltek. A RepLab 2013 versenyre kidolgozott rendszer első helyezést ért el. Korábbi publikációi (Hangya and Farkas, 2013a,b, 2017) alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Célorientált jellemzők kidolgozása felszíni jegyek alapján kevésbé jól formált szövegek esetében a releváns tartalom hangsúlyozása érdekében;
- Szintaxisalapú célorientált jellemzők kidolgozása jól formált szövegek esetében dependencia és konstituens elemzők felhasználásával;
- Hibrid rendszer kidolgozása a fent említett technikák kombinálásával és a rendszer hibáinak elemzése különböző adatbázisokon.

8.2.2 Aprólékos szentimentelemzés

A 4. fejezetben a mondatok szintaktikai összetétele került központba. A mondatok fő elemei és azok kapcsolata lehetőséget nyújt a tartalom jobb megértésére. Aprólékos szentimentelemzés során

a cél, hogy a mondatok minden egyes konstituenséhez egy szentimentértéket rendeljünk alulról felfelé haladva. Korábbi munkák megmutatták, hogy a módszer javítja a dokumentumszintű szentimentelemzés minőségét, azonban ezeknek a módszereknek a működéséhez vagy nehezen előállítható adatokra van szükség, melyek csak bizonyos nyelvek és domének esetében érhetőek el, vagy nyelvfüggő szabályokra épülnek, melyek nehezen adaptálhatóak más felhasználási esetekhez. A szerző és munkatársai egy rejtett szintaktikai struktúrán alapuló módszert dolgoztak ki, mely széles körben alkalmazható, mivel csak mondatszinten annotált szövegekre, illetve általános nyelvi jellemzőkre épül. Kísérleteken keresztül bemutattuk, hogy a kidolgozott módszer javítja a dokumentumszintű osztályozók pontosságát. Annak ellenére, hogy más aprólékosan annotált adatokat használó rendszerek jobban teljesítenek a megfelelő adatok rendelkezésre állása esetében, az itt kidolgozott módszer jobb eredményt ért el, amikor csak mondatszintű jelölés érhető el. Ezen felül a rendszer által előállított szentimentfák felhasználhatók célorientált elemzőrendszerekben is a fákból kinyert jellemzők segítségével, ami tovább növelte a célorientált rendszer minőségét. Korábbi publikációja (Hangya et al., 2017) alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Rejtett szintaktikai struktúrán alapuló aprólékos szentimentelemző rendszer bevezetése csak mondat szinten annotált szövegek felhasználásával;
- Dokumentumszintű szentimentelemzés pontosságának javítása mondatok összetevős elemzésével;
- Az aprólékos szentimentelemző rendszer integrálása célorientált elemzésbe szentimentfa alapú jellemzők kidolgozásával.

8.2.3 Különböző stílusú szövegek szentimentelemzése

Az 5. fejezet a disszertáció második fő témakörét, a doménkülönbségeket vezeti be, mely általánosságban véve a gépi tanulás egy nagy feladata. A probléma lényege, hogy egy adott szöveghalmazon kifejlesztett vagy betanított rendszerek nem teljesítenek jól más domének esetében. A szerző egy dokumentumszintű szentimentelemző rendszer működését hasonlította össze angol és magyar nyelvű szövegeken, melyek különböző domének és műfajok szövegeit tartalmazzák. A következő fejezeteket megalapozva, a szerző rámutatott a különböző doménekből származó szövegek használata esetén előforduló főbb problémákra. Korábbi publikációi (Hangya et al., 2013; Hangya and Farkas, 2017) alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Dokumentumszintű szentimentelemzés kvantitatív és kvalitatív összehasonlítása különböző doméneken, műfajokon és nyelveken keresztül.

8.2.4 Doménspecifikus szentimentlexikonok

A 6. fejezet középpontjában a szentimentlexikonok állnak, melyek kulcsfontosságú információforrások szentimentelemzéshez. Jelentős minőségjavulás érhető el szövegekből való extra jellemzők

kinyerésével ezen lexikonok segítségével. Viszont ahogy azt korábban már említettük, a szavak szentimentértéke erősen függ a doméntől, melyből a szövegeink származnak. Mivel ilyen lexikonok előállítása időigényes és drága folyamat, a legtöbb rendelkezésre álló lexikon általános célú. A szerző megmutatta, hogy jó minőségű lexikonok is ronthatják a rendszerek pontosságát, ha nem megfelelő domén esetében használjuk őket. A probléma orvoslására doménspecifikus lexikonokat építő módszereket mutattunk be. Megmutattuk, hogy jobb eredmények érhetőek el mind doménspecifikus alaplexikonok kiterjesztésével, mind új lexikonok létrehozásával annotált szövegek segítségével, mint kézzel előállított jó minőségű lexikon doménen kívüli használata esetén. Mivel a legtöbb rendelkezésre álló lexikon angol nyelvű, a szerző magyar nyelvű kísérleteket hajtott végre. Korábbi publikációja (Hangya, 2015) alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Doménspecifikus szentimentlexikonokat előállító nyelvfüggetlen automatikus módszerek kidolgozása: alaplexikonok kiterjesztése és új lexikonok létrehozása annotált szövegek segítségével;
- Doménspecifikus lexikonok fontosságának bemutatása különböző magyar korpuszokon a minél nagyobb pontosság elérésének érdekében.

8.2.5 Keresztnyelvi szentimentelemzés doménadaptációja

A 7. fejezet a keresztnyelvi szentimentelemzéshez kidolgozott doménadaptációs módszereket mutatta be. Szóbeágyazások és neurális hálózatok együttese jó minőségű szentimentelemző rendszerek létrehozását tette lehetővé. Ezen felül, a keresztnyelvi módszerek lehetővé tették, hogy egy erőforrásokban gazdag nyelv annotált szövegeit felhasználva olyan modellt hozzunk létre, mely alkalmazható olyan nyelvek esetében, amelyekre nem áll rendelkezésre annotált szöveg. Ezen módszerek sok esetben jól teljesítenek, de a domén változásával pontosságuk jelentősen csökkenhet. A szerző és munkatársai bemutatták, hogy általános célú erőforrásokat használó kétnyelvű szóbeágyazási modellek számos szó helytelen (a doménbe nem illeszkedő) jelentését tartalmazzák, valamint számos szó hiányzik belőlük. A problémára egy egyszerű doménadaptációs módszert ismertettünk, mely a korábbi munkákkal ellentétben csak egynyelvű jelöletlen szövegeket igényel. Ezen felül, egy további félig felügyelt módszert is bemutattunk, mely a hatékonyabb keresztnyelvi információátvitel érdekében lehetővé teszi további doménspecifikus jelöletlen adatok felhasználását. A végrehajtott kísérletek bemutatták, hogy az első módszer szükséges a második jó működése érdekében, valamint igazolták, hogy a kidolgozott módszerek feladat- és nyelvfüggetlenek. Korábbi publikációja (Hangya et al., 2018) alapján a szerző a tézis legfontosabb saját eredményeinek a következőket tekinti:

- Egyszerű, de hatékony módszer kidolgozása kétnyelvű szóbeágyazások doménadaptációja céljából;
- Feladatspecifikus célnyelvi/doménbeli adatok integrációját lehetővé tevő félig felügyelt módszer kidolgozása hatékonyabb keresztnyelvi információátvitel érdekében;
- Keresztnyelvi problémák doménadaptációja jelöletlen szövegek felhasználásával.

Bibliography

Enrique Amigó, Juio Gonzalo and Felisa Verdejo. 2012. Reliability and sensitivity: Generic evaluation measures for document organization tasks. Technical report, Universidad Nacional de Educación a Distancia.

Enrique Amigó, Jorge Carrillo de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Tamara Martín, Edgar Meij, Maarten de Rijke and Damiano Spina. 2013. Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. In *Proceedings of the 2nd Conference and Labs of the Evaluation Forum*.

Gabor Angeli, Christopher D. Manning and Daniel Jurafsky. 2012. Parsing Time: Learning to Interpret Time Expressions. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 446–455.

Nicholas Asher, Farah Benamara and Yvette Yannick Mathieu. 2009. Appraisal of Opinion Expressions in Discourse. *Lingvisticæ Investigationes*, 32(2):279–292.

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 2200–2204.

AR Balamurali and Adity Joshi. 2012. Cross-Lingual Sentiment Analysis for Indian Languages using Linked WordNets. In *Proceedings of the 25rd International Conference on Computational Linguistics*, pages 73–82.

Carmen Banea, Rada Mihalcea and Janyce Wiebe. 2010. Multilingual Subjectivity: Are More Languages Better? In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 28–36.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175.

Gábor Berend and Richárd Farkas. 2008. Opinion Mining in Hungarian based on textual and graphical clues. In *Proceedings of the 8th WSEAS International Conference on Simulation, Modelling and Optimization*, pages 23–25.

Steven Bird, Ewan Klein and Edward Loper. 2009. *Natural language processing with Python*. O'Reilly Media, Inc.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

Anders Björkelund and Jonas Kuhn. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 47–57.

David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022.

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

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97.

Julian Brooke, Milan Tofiloski and Maite Taboada. 2009. Cross-linguistic sentiment analysis: From English to Spanish. In *Proceedings of the international conference of Recent Advances in Natural Language Processing*, pages 50–54.

Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Hakan Ceylan and Rada Mihalcea. 2011. An Efficient Indexer for Large N-Gram Corpora. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 103–108.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 383–389.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 1–8.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.

Jean-Valere Cossu, Benjamin Bigot, Ludovic Bonnefoy, Mohamed Morchid, Xavier Bost, Gregory Senay, Richard Dufour, Vincent Bouvier, Juan-Manuel Torres-Moreno and Marc El-Beze. 2013. LIA@RepLab 2013. In *Working Notes of the 2nd Conference and Labs of the Evaluation Forum*.

Hal Daume. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca and Martin Jaggi. 2016. Swisscheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pages 1124–1128.

Xiaowen Ding, Bing Liu and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240.

Li Dong, Furu Wei, Shujie Liu, Ming Zhou and Ke Xu. 2015. A Statistical Parsing Framework for Sentiment Classification. *Computational Linguistics*, 41(2):265–308.

John Duchi, Elad Hazan and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird and Trevor Cohn. 2016. Learning Crosslingual Word Embeddings without Bilingual Corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1285–1295.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Manaal Faruqui and Chris Dyer. 2014. Improving Vector Space Word Representations Using Multilingual Correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471.

Joao Filgueiras and Silvio Amir. 2013. POPSTAR at RepLab 2013: Polarity for reputation classification. In *Working Notes of the 2nd Conference and Labs of the Evaluation Forum*.

G. David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Jennifer Foster, Özlem Çetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan and Josef Van Genabith. 2011. #hardtoparse: POS Tagging and Parsing the Twitterverse. In *Proceedings of the AAAI Workshop on Analyzing Microtext*, pages 20–25.

Miguel Ángel García-Cumbreras, Janine Villena-Román, Eugenio Martínez-Cámara, Manuel C. Díaz-Galiano, M. Teresa Martín-Valdivia and L. Alfonso Ureña López. 2016. Overview of TASS 2016. In *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN*, pages 13–21.

Alec Go, Richa Bhayani and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Stephan Gouws, Yoshua Bengio and Greg Corrado. 2015. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. In *Proceedings of the International Conference on Machine Learning*.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390.

Lin Gui, Ruifeng Xu, Qin Lu, Jun Xu, Jian Xu, Bin Liu and Wang Xiaolong. 2013. A Mixed Model for Cross Lingual Opinion Analysis. In *Proceedings of the Natural Language Processing and Chinese Computing*, pages 93–104.

Zhen Hai, Kuiyu Chang and Jung Jae Kim. 2011. Implicit feature identification via co-occurrence association rule mining. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 393–404.

Viktor Hangya. 2015. Automatic Construction of Domain Specific Sentiment Lexicons for Hungarian. In *Proceedings of the 18th International Conference on Text, Speech and Dialogue*, pages 201–208.

Viktor Hangya, Gábor Berend and Richárd Farkas. 2013. SZTE-NLP: Sentiment Detection on Twitter Messages. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 549–553.

Viktor Hangya, Gábor Berend, István Varga and Richárd Farkas. 2014. SZTE-NLP: Aspect level opinion mining exploiting syntactic cues. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 610–614.

Viktor Hangya, Fabienne Braune, Alexander Fraser and Hinrich Schütze. 2018. Two Methods for Domain Adaptation of Bilingual Tasks: Delightfully Simple and Broadly Applicable. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 810–820.

Viktor Hangya and Richárd Farkas. 2013a. Filtering and Polarity Detection for Reputation Management on Tweets. In *Working Notes of the 2nd Conference and Labs of the Evaluation Forum*.

Viktor Hangya and Richárd Farkas. 2013b. Target-oriented opinion mining from tweets. In *Proceedings of the 4th IEEE International Conference on Cognitive Infocommunications*, pages 251–254.

Viktor Hangya and Richárd Farkas. 2017. A comparative empirical study on social media sentiment analysis over various genres and languages. *Artificial Intelligence Review*, 47(4):485–505.

Viktor Hangya, Richárd Farkas and Gábor Berend. 2015. Entitásorientált véleménydetekció webes híranyagokból. In *XI. Magyar Számítógépes Nyelvészeti Konferencia*, pages 227–234.

Viktor Hangya, Zsolt Szántó and Richárd Farkas. 2017. Latent Syntactic Structure-Based Sentiment Analysis. In *Proceeding of the 2nd IEEE International Conference on Computational Intelligence and Applications*.

Philip Häusser, Alexander Mordvintsev and Daniel Cremers. 2017. Learning by Association - A versatile semi-supervised training method for neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual Models for Compositional Distributed Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 58–68.

Geert Heyman, Ivan Vulić and Marie-Francine Moens. 2017. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 1084–1094.

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

Minqing Hu and Bing Liu. 2004b. Mining Opinion Features in Customer Reviews. In *American Association for Artificial Intelligence*, pages 755–760.

Liang Huang. 2008. Forest Reranking: Discriminative Parsing with Non-Local Features. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 586–594.

Bernard J. Jansen, Mimi Zhang, Kate Sobel and Abdur Chowdury. 2009. Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 151–160.

Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Proceedings of the 2008 international conference on web search and data mining*, pages 219–230.

Martin Joos. 1950. Description of language design. *The Journal of the Acoustical Society of America*, 22(6):701–707.

Daniel Jurafsky and James H. Martin. 2000. An introduction to natural language processing, computational linguistics, and speech recognition.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc.

Jaap Kamps, Maarten Marx, Robert J. Mokken and Maarten De Rijke. 2004. Using WordNet to Measure Semantic Orientations of Adjectives. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1115–1118.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363.

Jason S. Kessler, Miriam Eckert, Lyndsie Clark and Nicolas Nicolov. 2010. The 2010 ICWSM JDPA Sentiment Corpus for the Automotive Domain. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge*.

Adam Kilgarriff. 1997. Putting frequencies in the dictionary. *International Journal of Lexicography*, 10(2):135–155.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry and Saif M. Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proceedings the International Workshop on Semantic Evaluation*, pages 437–442.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistic*, pages 423–430.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit*, pages 79–86.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer and Noah A. Smith. 2014. A Dependency Parser for Tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1001–1012.

Vandana Korde and C. Namrata Mahender. 2012. Text classification and classifiers: A survey. *International Journal of Artificial Intelligence and Applications*, 3(2):85.

Akrivi Krouska, Christos Troussas and Maria Virvou. 2016. The effect of preprocessing techniques on Twitter sentiment analysis. In *Proceedings of the International Conference on Information, Intelligence, Systems and Applications*, pages 1–5.

Khang Nhut Lam, Feras Al Tarouti and Jugal Kalita. 2014. Automatically constructing Wordnet Synsets. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 106–111.

Angeliki Lazaridou, Georgiana Dinu and Marco Baroni. 2015. Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 270–280.

Angeliki Lazaridou, Ivan Titov and Caroline Sporleder. 2013. A Bayesian Model for Joint Unsupervised Induction of Sentiment, Aspect and Discourse Representations. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1639.

Qi Li. 2012. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York*, pages 1–54.

Shi Li, Lina Zhou and Yijun Li. 2015. Improving aspect extraction by augmenting a frequency-based method with web-based similarity measures. *Information Processing & Management*, 51(1):58–67.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Language Resources and Evaluation Conference*, pages 923–929.

Bing Liu. 2010. Sentiment Analysis and Subjectivity. *Handbook of natural language processing*, 2:627–666.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Bin Lu, Chenhao Tan, Claire Cardie and Benjamin K. Tsou. 2011. Joint Bilingual Sentiment Classification with Unlabeled Parallel Corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 320–330.

Pranava Swaroop Madhyastha and Cristina España Bohnet. 2017. Learning Bilingual Projections of Embeddings for Vocabulary Expansion in Machine Translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 139–145.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. The MIT Press.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 55–60.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. Http://mallet.cs.umass.edu.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells and Jeff Reynar. 2007. Structured Models for Fine-to-Coarse Sentiment Analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439.

Rada Mihalcea, Carmen Banea and Janyce Wiebe. 2007. Learning Multilingual Subjective Language via Cross-Lingual Projections. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 976–983.

Márton Miháltz. 2013. OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In *IX. Magyar Számítógépes Nyelvészeti Konferencia*, pages 343–345.

Márton Miháltz, Csaba Hatvani, Judit Kuti, György Szarvas, János Csirik, Gábor Prószéky and Tamás Váradi. 2008. Methods and results of the Hungarian WordNet project. In *Proceedings of the 4th Global WordNet Conference*, pages 387–405.

Márton Miháltz. 2010. OpinHu: online szövegek többnyelvű véleményelemzése. In *VII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 14–23.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations.*

Tomas Mikolov, Quoc V. Le and Ilya Sutskever. 2013b. Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4168.

George Miller. 1998. *WordNet: An electronic lexical database.* MIT Press.

Saif Mohammad, Cody Dunne and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 599–608.

Alejandro Mosquera, Javi Fernandez, Jose M. Gomez, Patricio Martinez-Barco and Paloma Moreda. 2013. DLSI-Volvam at RepLab 2013: Polarity Classification on Twitter Data. In *Working Notes of the 2nd Conference and Labs of the Evaluation Forum.*

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani and Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 1–18.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge and Noah A. Smith. 2010. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media.*

Georgios Paltoglou and Mike Thelwall. 2010. A study of Information Retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395.

Sinno Jialin Pan, Qiang Yang and Others. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271—278.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 1(1–2):1–135.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertran Thirion, Oliver Grisel, Matthieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Irene Pollach. 2011. Software review: Wordstat 5.0. *Organizational Research Methods*, 14(4):741–744.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 27–35.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq et al. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar and Ion Androutsopoulos. 2015. Semeval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 486–495.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting Product Features and Opinion from Reviews. In *Natural language processing and text mining*, pages 9–28.

Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui and Alexander Gelbukh. 2014. A Rule-Based Approach to Aspect Extraction from Product Reviews. In *Proceedings of the 2nd Workshop on Natural Language Processing for Social Media*, pages 28–37.

Marting F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Robert Remus. 2014. *Genre and Domain Dependencies in Sentiment Analysis.* Ph.D. thesis, Universität Leipzig.

Sara Rosenthal, Noura Farra and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 502–518.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter and Veselin Stoyanov. 2015. Semeval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 451–463.

Sara Rosenthal, Preslav Nakov, Alan Ritter and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 73–80.

Sascha Rothe, Sebastian Ebert and Hinrich Schuetze. 2016. Ultradense Word Embeddings by Orthogonal Transformation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777.

Sebastian Ruder. 2017. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902.

Erik Tjong Kim Sang and Johan Bos. 2012. Predicting the 2011 Dutch Senate Election Results with Twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 53–60.

Kim Schouten and Flavius Frasincar. 2016. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Martina Katalin Szabó. 2014. Egy magyar nyelvű szentimentlexikon létrehozásának tapasztalatai. In *Proceedings of the Conference Nyelv, kultúra, társadalom*, pages 3–4.

Martina Katalin Szabó, Veronika Vincze, Katalin Simkó, Viktor Varga and Viktor Hangya. 2016. A Hungarian sentiment corpus manually annotated at aspect level. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 2873–2878.

Martina Katalin Szabó, Veronika Vincze and Viktor Hangya. 2016. Aspektusszintű annotáció és szentimentet módosító elemek egy magyar nyelvű szentimentkorpuszban. In *XII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 174–182.

Zsolt Szántó and Richárd Farkas. 2014. Special Techniques for Constituent Parsing of Morphologically Rich Languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 135–144.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Oscar Tackström and Ryan McDonald. 2011. Discovering Fine-Grained Sentiment with Latent Variable Structured Prediction Models. In *Proceedings of the European Conference on Information Retrieval*, pages 368–374.

Xuewei Tang and Xiaojun Wan. 2014. Learning bilingual embedding model for cross-language sentiment classification. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, pages 134–141.

Ann Taylor, Mitchell Marcus and Beatrice Santorini. 2003. The Penn treebank: an overview. In *Treebanks*, pages 5–22. Springer.

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

Shyam Upadhyay, Manaal Faruqui, Chris Dyer and Dan Roth. 2016. Cross-lingual Models of Word Embeddings: An Empirical Comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1661–1670.

István Varga, Motoki Sano, Kentaro Torisawa, Chikara Hashimoto, Kiyonori Ohtake, Takao Kawai, Jong-Hoon Oh and Stijn De Saeger. 2013. Aid is Out There: Looking for Help from Tweets during a Large Scale Disaster. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistic*, pages 1619–1629.

David Vilares, Miguel A. Alonso and Carlos Gómez-Rodriguez. 2013. A syntactic approach for opinion mining on Spanish reviews. *Natural Language Engineering*, 21(1):139–163.

David Vilares, Miguel A. Alonso and Carlos Gómez-Rodríguez. 2015. On the usefulness of lexical and syntactic processing in polarity classification of Twitter messages. *Journal of the Association for Information Science and Technology*, 66(9):1799–1816.

G. Vinodhini and RM. Chandrasekaran. 2012. Sentiment Analysis and Opinion Mining: A Survey. *International Journal*, 2(6):282–292.

Ivan Vulić and Anna Korhonen. 2016. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 247–257.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 719–725.

Ivan Vulić and Marie-Francine Moens. 2016. Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, 55:953–994.

Joachim Wagner, Piyoush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster and Lamia Tounsi. 2014. DCU: Aspect-based polarity classification for SemEval Task 4. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 223–229.

Xiaojun Wan. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 553–561.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pages 235–243.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow and Andrés Montoyo. 2010. A Survey on the Role of Negation in Sentiment Analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 60–68.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov and Alan Ritter. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*.

Theresa Wilson, Janyce Wiebe and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the joint Conferences on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354.

Dan Wu, Wee Sun Lee, Nan Ye and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1523–1532.

Min Xiao and Yuhong Guo. 2013. Semi-Supervised Representation Learning for Cross-Lingual Text Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1465–1475.

Chao Xing, Dong Wang, Chao Liu and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

Changli Zhang, Daniel Zeng, Jiexun Li, Fei-Yue Wang and Wanli Zuo. 2009. Sentiment analysis of Chinese documents: From sentence to document level. *Journal of the American Society for Information Science and Technology*, 60(12):2474–2487.

Meishan Zhang, Yue Zhang and Duy-Tin Vo. 2016. Gated Neural Networks for Targeted Sentiment Analysis. In *Proceedings of the 30th Conference on Artificial Intelligence*, pages 3087–3093.

Yanyan Zhao, Bing Qin, Shen Hu and Ting Liu. 2010. Generalizing syntactic structures for product attribute candidate extraction. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 377–380.

Guangyou Zhou, Tingting He and Jun Zhao. 2014. Bridging the Language Gap: Learning Distributed Semantics for Cross-Lingual Sentiment Classification. In *Proceedings of the Natural Language Processing and Chinese Computing: 3rd CCF Conference*, pages 138–149.

Huiwei Zhou, Long Chen, Fulin Shi and Degen Huang. 2015. Learning bilingual sentiment word embeddings for cross-language sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 430–440.

Xinjie Zhou, Xianjun Wan and Jianguo Xiao. 2016. Cross-Lingual Sentiment Classification with Bilingual Document Representation Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1403–1412.

Xiaodan Zhu, Svetlana Kiritchenko and Saif M. Mohammad. 2014. NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 443–447.

Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt and Michael Strube. 2011. Fine-grained sentiment analysis with structural features. In *Proceedings of International Joint Conference on Natural Language Processing*, pages 336–344.

János Zsibrita, Veronika Vincze and Richárd Farkas. 2013. Magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In *Proceedings of Conference of the Recent Advances in Natural Language Processing*, pages 763–771.