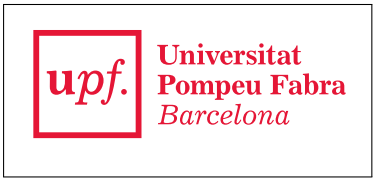
Cross-lingual Sentiment Analysis for Under-resourced Languages

Jeremy Barnes

TESI DOCTORAL UPF / ANY 2018

DIRECTOR DE LA TESI

Patrik Lambert and Toni Badia
Departament Traducció i Ciències del Llenguatge



Dedicated to Itziar. Without you, none of this would have been possible.

Acknowledgments

First of all, I would like to thank my wife, Itziar. You have been my source of stability, the one who convinced me that it was possible, and the one who brought a smile to my face after many a long night's work. You have endured changing countries, changing jobs, and so many inconveniences and I can't begin to thank you enough.

I thank my supervisors Toni Badia and Patrik Lambert, who have helped and guided me through this journey. Both of you have given so much of your time and energy to help me discover what research really is and how to do it in a purposeful and exact manner.

I would also like to thank my two supervisors during my research stay at the University of Stuttgart, Sabine Schulte im Walde and Roman Klinger. Sabine, thank you for supporting me during my stay, for all of the great conversations, and for pushing me to be better. Roman, you've been a great role model for how to conduct serious research. I greatly admire your enthusiasm and attention to detail and thank you for letting me into your research group.

I would like to thank all the friends I made in Stuttgart. Diego and Gabriella, first, for making our transition from Barcelona a little easier. Maximillian Köper and Kim Anh Nguyen, thanks for all of the morning coffees and conversations. Thanks to Evgeny Kim and Laura Bostan for the great times at conferences. Thanks to Jonas Kuhn and his family for giving us a place to stay when all seemed lost.

Finally, I would like to thank my parents for giving me the confidence and moral support from the time I decided to go to college. Without this, I would have never made it to where I am today.

Abstract

The amount of user-generated content available on the internet is constantly growing and with it, the opportunity to find methods which allow us to infer valuable information from this content. Sentiment Analysis is a task that allows us to calculate the polarity of this content automatically. While some languages, such as English, have a vast array of resources to enable sentiment analysis, most under-resourced languages lack them. *Cross-lingual Sentiment Analysis* (CLSA) attempts to make use of resource-rich languages in order to create or improve sentiment analysis systems in a under-resourced language. Machine translation is the most common way of transferring these resources, yet it is not always available nor the optimal solution. The objective of this thesis is to explore approaches than enable sentiment analysis in under-resourced languages, while moving from coarse- to fine-grained sentiment.

Until now, there has been little investigation into CLSA for languages that lack large amounts of parallel data. Here, we propose cross-lingual sentiment approaches that have minimal parallel data requirements, while making the best use of available monolingual data. We start by determining the characteristics of state-of-the-art monolingual sentiment models that would be interesting for this task and comparing machine translation and cross-lingual distributional representations. We propose a model to incorporate sentiment information into bilingual distributional representations, by jointly optimizing them for semantics and sentiment, showing state-of-the-art performance when combined with machine translation. We then move these approaches to aspect-level and subsequently test them on a variety of language families and domains. Finally, we show that this approach can also be suitable for domain adaptation.

Resum

La quantitat de contingut creat pels usuaris a Internet creix constantment i al mateix temps, l'oportunitat de trobar mètodes que ens permetin treure'n informació de valor. L'anàlisi de sentiment és una tasca que ens permet calcular la polaritat d'aquesta informació de manera automàtica. Mentre algunes llengües, com l'anglès per exemple, tenen una àmplia varietat de recursos per crear sistemes d'anàlisi de sentiment, n'hi ha més que els troben a faltar. *L'Anàlisi de Sentiment Cross-lingüe* (ASCL) intenta fer servir els recursos de llengües riques en recursos per crear o millorar sistemes d'anàlisi d'opinions en llengües pobres en recursos. La traducció automàtica és una de les maneres més corrents per transferir aquests recursos, però no sempre existeix ni és sempre la solució òptima. L'objectiu d'aquesta tesi és explorar mètodes que facin possible l'anàlisi de sentiment en llengües amb pocs recursos, i al mateix temps, passar de fer-ho a un nivell de document o frase a nivell d'aspect.

Fins ara, hi ha hagut poca investigació en ASCL a aquest nivell de granularitat, encara que moltes vegades seria més útil. Nosaltres proposem mètodes d'anàlisi de sentiment cross-lingües que requereixen menys data paral·lela i treuen el màxim profit de data monolingüe que tenim a l'abast. Comencem per determinar les característiques dels models que formen l'estat de l'art que podrien servir per a la nostra tasca. Després comparem la traducció automàtica i representacions distribucionals cross-lingües. Proposem un model que optimitza les representacions distribucionals cross-lingües perquè tinguin informació semàntica i també de sentiment, i que demostra ser l'estat de l'art en combinant-se amb traducció automàtica. Després passem a un nivell de granularitat més fina i examinem com canvia el rendiment dels models amb diferents llengües metes i dominis. Finalment, demostrem que aquestes tècniques també són adequats per a l'adaptació de domini.

Laburpena

Interneteko erabiltzaileek egunero sortzen duten edukia etengabe handitzen ari da, eta halaber, eduki honetatik informazio baliotsua eskuratzeko aukera. Sentimenduen analisia eduki honen polaritatea automatikoki kalkulatzeko aukera ematen digun ataza bat da. Hizkuntza batzuk, ingelesak adibidez, baliabide aukera zabala daukaten bitartean, baliabide gutxiko hizkuntza gehieni falta zaio. *Hizkuntza-arteko Sentimenduen Analisia* (HSA) baliabide handiko hizkuntza bateko anotazioak erabiltzen saiatzen da, baliabide gutxiko hizkuntza batean sentimenduen analisi automatikoa hobetzeko helburuz. Itzulpen automatikoa baliabide hauek transferitzeko modurik erabilena da, baina ez da beti existitzen ezta beti soluziorik onena izaten. Tesi honen helburua baliabide gutxiko hizkuntzetan sentimenduen analisia ahalbideratzen duten hurbilketak esploratzea da, eta era berean, dokumentuen eta fraseen sailkaketatik aspektuen sailkaketara pasatzea.

Orain arte, baliabide paraleloak falta zaizkien hizkuntzerako HSA-ko ikerketa gutxi izan da. Tesi honetan, baliabide paralelo gutxi behar duten baina eskura dauden elebakarreko baliabide anitzak erabiltzen dituzten HSA-ko hurbilketa proposatzen ditugu. Lehenik eta behin, gure atazarako interesgarriak izan litezkeen elebakarreko sentimenduen analisiaren ereduen ezaugarriak aztertzen ditugu. Gero itzulpen automatikoa eta hizkuntza-arteko bektore espazioak alderatzen ditugu. Hizkuntza-arteko bektore espazioetan sentimenduen informazioa sartzeko modelo bat proposatzen dugu, aldi berean semantika eta sentimendua sailkatzeko helburuak optimizatzearen bidez, itzulpen automatikoarekin konbinatuta arteko egoerako emaitzarik onenak lortuz. Hurbilketa hauek fraseak sailkatzetik aspektuak sailkatzera pasatzen ditugu eta hizkuntza familia eta domeinu anitzetan aztertzen ditugu. Azkenik, hurbilketa hauek sentimenduen analisirako domeinu-egokitzapenarako ere egokiak direla egiaztatzen dugu.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

> Consider that everything is opinion,
> and opinion is in thy power.
>
> *Marcus Aurelius, Meditations*

This thesis concerns the transfer of sentiment information from resource-rich languages, *i. e.* English, to under-resourced languages. This task, known as **Cross-lingual Sentiment Analysis** (CLSA), is of interest to anyone who wishes to perform sentiment analysis, but does not have the time or money to curate hand-annotated datasets to train supervised machine learning algorithms. Specifically, our goal is to develop approaches for cross-lingual sentiment that require only *small amounts of parallel data*, allowing us to perform sentiment analysis in under-resourced languages.

For the purposes of this thesis, we define an **under-resourced language** as any language which lacks annotated sentiment data and which does not have enough parallel data available with English to enable us to easily build a high-quality machine translation system. This means that most languages which are not official languages spoken in Western Europe can be considered under-resourced for our purposes. In fact, we work extensively with co-official languages spoken in Western Europe, such as Catalan and Basque, which for our purposes are under-resourced languages.

This chapter summarizes the motivation for the present line of research, summarizes the approach presented later in the thesis, as well as the aims and objectives. Finally, it also gives the reader an overview of the organization of this thesis.

sentiment target
positive sentiment phrase
negative sentiment phrase

Table 1.1: Colored example legend. These colors will be used for examples throughout the thesis.

1.1 Motivation

Opinions are everywhere in our lives. Every time we open a book, read the newspaper, or look at social media, we scan for opinions and form them ourselves. Opinions give us an idea of who our friends are and who we trust. We also love to show people our opinions, as a way to define our personality. This is true on the Internet as much as it is in our face-to-face relationships. With the wealth of opinionated material now available on the Internet, it has become feasible and interesting to harness this data in order to automatically identify opinions.

Sentiment analysis, sometimes also referred to as **opinion mining**, seeks to do exactly this: create data-driven methods to classify the polarity of a text. The information obtained from sentiment classifiers can then be used for tracking user opinions of movies (Pang et al., 2002; Socher et al., 2013), predicting the outcome of political elections (Wang et al., 2012; Bakliwal et al., 2013), fighting hate speech online (Nahar et al., 2012; Hartung et al., 2017), as well as predicting the stock market to a degree (Pagolu et al., 2016).

Supervised machine learning algorithms have become the most successful approaches to sentiment analysis. In supervised learning, a statistical model sees a set of training examples $X = \{x_1, x_2, \ldots, x_n\}$ and their corresponding labels $Y = \{y_1, y_2, \ldots, y_n\}$ and learns a function $f(X; \theta) \rightarrow Y$, where θ are the model parameters. The assumption is that future test examples will come from a similar distribution as X, allowing the model to generalize. This framework performs well for sentiment analysis, but requires a large amount of annotated examples, which must be found, compiled, and thoughtfully annotated in order to produce high-quality training data. Let us take the following hotel review as an example of what complexities lie in determining sentiment:

(1) We stayed at the hostel for two nights on the weekend. If you're looking for a cheap hostel in the center of the city, it's not a bad option. Breakfast isn't included, but there's a café on the other side of the street with incredible croissants.

As we can see from Example 1, opinions come in varying degrees of granularity. The overall polarity of the review, which we refer to as **document-level** classification, is relatively positive. The polarity of the first sentence of this review, known as **sentence-level** classification, however, is neutral as it contains only factual information which does not give any opinion. The last sentence contains a mixture of polarities regarding specific entities or characteristic of those entities. While the opinion expressed about "breakfast" at the hotel is negative, polarity expressed towards the "croissants" is positive. We refer to this last task as **aspect-level** or **targeted** sentiment analysis.

At document-level, the main difficulty is to discern which sentiment is more prevalent, as documents can easily contain a mix of positive and negative sentiment. The signal, however, is normally redundant, making it relatively easy to reach a respectable accuracy. At sentence-level, there is much less signal, and a larger amount of ambiguity. This makes sentence-level sentiment analysis more challenging. Many times, however, there is still conflicting sentiment within a sentence, which cannot easily be resolved within the sentence-level framework. The need to resolve conflicting intra-sentential sentiment motivates moving from document- or sentence-level to aspect-level. Unlike the more coarse-grained approaches, aspect-level sentiment analysis attempts to classify the polarity of an entity or characteristic of that entity. The advantage is that aspect-level sentiment analysis is a more realistic view of opinions, as opinions normally have a target, even if it is not mentioned explicitly. Aspect-level sentiment analysis is the most challenging subtask, both for sentiment classifiers as well as for annotation, because it is necessary to identify the aspect in question, the opinion phrase that refers to it, and resolve their relations.

Resource-rich languages, such as English, Spanish, or German, have high-quality annotated data for many sentiment tasks and domains. However, under-resourced languages either completely lack annotated data or have only a few resources for specific domains or sentiment tasks. The cost of annotating data can often be prohibitive as training native-speakers to annotate fine-grained sentiment is a long process. This motivates the need to develop sentiment analysis methods capable of leveraging data annotated in other languages.

Cross-lingual sentiment analysis (CLSA) offers us a way to perform sentiment analysis in an under-resourced language that does not have any annotated data available. Previous approaches to cross-lingual sentiment analysis have relied heavily on large amounts of parallel data to transfer sentiment information across languages. **Machine translation** (MT) has been the most common approach to cross-lingual sentiment analysis (Banea et al., 2013; Almeida et al., 2015; Zhang and Wallace, 2017). Accurate machine translation, however, requires millions of

parallel sentences, which places a limit on which languages can benefit from these approaches.

Although high quality machine translation systems already exist between many languages and have been shown to enable cross-lingual sentiment analysis, for the vast majority of language pairs in the world there is not enough parallel data between them to create these high quality MT systems. This lack of parallel data coupled with the computational expense of MT means that approaches to cross-lingual sentiment analysis that do not require MT are to be preferred.

MT also introduces noise through translation errors. Let us look at this example of a hotel review in Basque and an automatic translation achieved with a state-of-the-art machine translation system[1]:

(2) Hotel txukuna da, nahiko berria. Harreran zeuden langileen arreta ez zen onena izan. Tren geltoki bat du 5 minutura eta kotxez berehala iristen da baina oinez urruti samar dago.

The hotel is tidy, quite new. The care of the workers at reception was not the best. It's 5 minutes away from a train station and it's quick to reach the car, but it's a short distance away.

While the first two sentences are mostly well translated for the purposes of sentiment analysis, in the third, there are a number of reformulations and deletions that lead to a loss of information. The third sentence should read "It has a train station five minutes away and by car you can reach it quickly, but by foot it's quite a distance." We can see that one of the aspects has been deleted and the sentiment has flipped from negative to positive. These problems are quite common in MT and degrade the results of cross-lingual sentiment systems that use MT.

Recently proposed **bilingual distributional semantics models** (bilingual embeddings) provide a useful framework for cross-lingual research without requiring machine translation. They have proven to be useful features for bilingual dictionary induction (Mikolov et al., 2013b; Artetxe et al., 2016; Lample et al., 2018a), cross-lingual text classification (Prettenhofer and Stein, 2011; Chandar et al., 2014), or cross-lingual dependency parsing (Søgaard et al., 2015), among others. In this framework, a word is represented as an n-dimensional vector. These vectors are trained on large monolingual corpora in order to 1) maximize the similarity of words that appear in similar contexts and use some bilingual regularization in order to 2) maximize the similarity of translation pairs. In this thesis, we concentrate on a subset of these bilingual embedding methods that perform a post-hoc mapping

[1] `https://translate.google.com/`

to a bilingual space, which we refer to as **embedding projection methods**. A common feature of these approaches is that they first create separate monolingual vector spaces on purely monolingual data and then learn to project these into a bilingual space. One of the main advantages of these methods is that they make better use of small amounts of parallel data than MT systems. In fact, recently proposed unsupervised projection methods are able to achieve comparable results to supervised methods (Artetxe et al., 2017; Lample et al., 2018a) on bilingual word similarity tasks, as well as enabling unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018b).

These bilingual representations, however, have not been tested extensively on cross-lingual sentiment tasks. In this thesis, we compare distributional approaches to MT and identify a disadvantage of current bilingual embedding methods; namely, they do not incorporate sentiment information. We propose a joint sentiment-projection method that only requires a small bilingual dictionary and annotated data in the source language.

1.2 Aims and Research Questions

The main goal of this thesis is to enable sentiment analysis in under-resourced languages. As such, we introduce a number of sub-goals and associated research questions which we will revisit in Chapter 5.

Create sentiment resources for under-resourced languages:

1. Introduce aspect-level sentiment datasets for Catalan and Basque.
2. Create and release machine learning resources for quickly prototyping cross-lingual models.

Propose machine learning models for cross-lingual sentiment analysis that do not require machine translation:

1. Do monolingual vector spaces contain enough distributional information for a sentiment classifier to learn to both project them to a common space and learn to classify sentiment?
2. If this is possible, how much parallel data is necessary to perform the transfer?
3. How much source language annotated data is necessary to learn to classify the target language?

4. What amount of loss of accuracy does this joint model suffer when compared to monolingual models?
5. Is it possible to improve machine-translation based CLSA methods using this approach?

Move cross-lingual sentiment models beyond sentence-level:

1. Given a bilingual sentiment representation, is it better to assume that all aspects in a phrase have the same polarity, or try to predict each separately?
2. Can we predict the sentiment of aspects in a target language without using machine translation?

1.3 Structure of Thesis

Chapter 2 details the corpora, datasets, and tools which we use throughout the thesis.

Chapter 3 discusses state-of-the-art methods used in both monolingual and cross-lingual sentiment analysis and how they relate to the goals of this thesis. This chapter provides a general view, while some of the specifics of these approaches are discussed in more detail in the relevant section in Chapter 4 in order to facilitate reading.

Chapter 4 reports on a series of experiments that build upon one another. The first is an exploratory experiment to determine which state-of-the-art models work best for sentiment analysis. The second tests the feasibility of using bilingual distributional representations of words compared to machine translation and points out some of their shortcomings. Next, we propose a model that jointly learns to project monolingual vectors to a bilingual space and predict sentiment at sentence-level. The next experiment proposes methods to move from sentence- to aspect-level CLSA. We then report on the deployment of this system to large amounts of real world data. Finally, we show that our cross-lingual methods can also perform well on domain-adaptation tasks, providing evidence that the methods we develop generalize well.

As such, Sections 4.3 - 4.6 comprise the main contribution of the thesis, whereas Sections 4.1 - 4.2 contain important but preliminary work.

Finally, chapter 5 provides the conclusions of the thesis and discusses relevant areas of future work.

1.4 Publications

Part of the work presented in this dissertation has been published in peer-review conference proceedings. A list of these publications follows:

1. Jeremy Barnes, Patrik Lambert, and Toni Badia (2016). "Exploring Distributional Representations and Machine Translation for Aspect-based Cross-lingual Sentiment Classification." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1613-1623.

2. Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2017). "Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment datasets." In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 2-12.

3. Jeremy Barnes, Patrik Lambert, and Toni Badia (2018). "MultiBooked: A Corpus of Basque and Catalan Hotel Reviews Annotated for Aspect-level Sentiment Classification." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 656-660.

4. Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2018). "Bilingual Sentiment Embeddings: Joint Projection of Sentiment Across Languages." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2483-2493.

5. Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2018). "Projecting Embeddings for Domain Adaptation: Joint Modeling of Sentiment Analysis in Diverse Domains." In: *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics*, pp. 818-830.

Additionally, sections 4.3 - 4.5 will be included in a journal article which is currently under review.

1. Jeremy Barnes, and Roman Klinger (2018). "Projection-based Cross-lingual Sentiment Analysis for Under-resourced Languages" In: *The Journal of Artificial Intelligence Research*, (under review).

Chapter 2

DATA, TOOLS, AND METHODS

2.1 Datasets

As one of the goals of this thesis is to find approaches that generalize well to under-resourced languages, for the experiments in Chapter 4, we test our approach on many sentiment annotated datasets. We list these datasets and their most important characteristics in this subsection. However, of all these datasets, two are of vital importance to this thesis: namely, the OpeNER datasets and the MultiBooked datasets. These datasets are composed of hotel reviews which have been annotated at aspect-level. The main advantage of using these datasets is that they have been drawn from similar sources and annotated similarly across four languages (English, Spanish, Catalan, and Basque). This allows us to avoid problems of domain or adaptation of labeling schemes and enables us to concentrate entirely on the cross-lingual transfer in this thesis. The OpeNER dataset is described in detail in Section 2.1.1, while the annotation project MultiBooked undertaken during the course of the thesis is presented in Section 2.1.2.

2.1.1 OpeNER datasets

The OpeNER dataset (Agerri et al., 2013) are datasets of hotel reviews annotated at aspect-level available in Dutch, German, English, French, Italian, and Spanish. In this thesis, however, we will only make use of the English and Spanish datasets. Each hotel review is sentence- and word-tokenized, POS-tagged, chunked, and

	English	Spanish
Number of Reviews	396	409
Average length in tokens	93.3	86.8
Number of Targets	3850	3980
Number of Expressions	4150	4388
Number of Holders	413	255

Table 2.1: Corpus Statistics

		Binary		Multiclass			
		+	−	++	+	−	−−
Number of Opinions	EN	1658	661	472	1132	556	105
	ES	2404	446	813	1591	387	59

Table 2.2: Distribution of class labels for the OpeNER datasets

parsed using Ixa-pipes (Agerri et al., 2014). The annotation scheme follows the full opinion aspect-level formulation described later in Section 3.1.4.

For each sentence in the reviews, annotators first determine if the sentence contains an opinion or not. If so, they annotate the opinion phrase, as well as the opinion holder and opinion target if they are mentioned. This leads to opinion triplets where the only mandatory information is the opinion phrase.

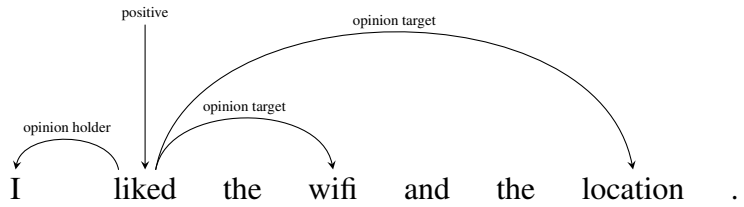


Figure 2.1: An example of an annotated opinion from the English OpeNER dataset

Note that a single opinion phrase, “liked”, can have multiple targets, “the wifi” and “the location”. One-to-many and many-to-one relationships are quite common in these datasets. During training and testing, each triplet is considered a separate occurrence.

The statistics for these datasets are shown in Tables 2.1 and 2.2. One interesting fact to point out that will be relevant for cross-lingual approaches is the difference in the number of opinion holders in English and in Spanish (413 compared to 255)

despite a larger number of reviews and opinion targets and expressions. This is due to the fact that Spanish is a pro-drop language, meaning that if the subject of the main verb is clear, it is not normally expressed overtly, which leads to a much smaller number of opinion holders. These two datasets are used extensively in a number of experiments in Chapter 4.

Jeremy Barnes, Patrik Lambert, and Toni Badia (2018). "MultiBooked: A Corpus of Basque and Catalan Hotel Reviews Annotated for Aspect-level Sentiment Classification." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 656-660. http://www.lrec-conf.org/proceedings/lrec2018/pdf/217.pdf.

2.1.2 MultiBooked datasets

Introduction and Motivation

The movement towards multi-lingual datasets for sentiment analysis is important because many languages offer different challenges, such as complex morphology or highly productive word formation, which can not be overcome by focusing only on English data.

In English there are many datasets available for document- and sentence-level sentiment analysis across different domains and at different levels of annotation (Pang et al., 2002; Hu and Liu, 2004; Blitzer et al., 2007; Socher et al., 2013; Nakov et al., 2013). These resources have been built up over a period of more than a decade and are currently necessary to achieve state-of-the-art performance.

Corpora annotated at fine-grained levels (opinion- or aspect-level) require more effort from annotators, but are able to capture information which is not present at document- or sentence-level, such as nested opinions or differing polarities of different aspects of a single entity. In English, the MPQA corpus (Wiebe et al., 2005) has been widely used in fine-grained opinion research. More recently, a number of SemEval tasks have concentrated on aspect-level sentiment analysis (Pontiki et al., 2014, 2015, 2016).

The Iberian peninsula contains two official languages (Portuguese and Spanish), as well as three co-official languages (Basque, Catalan, and Galician) and several smaller languages (Aragonese, Gascon). The two official languages do have available resources for sentiment at tweet-level (Villena-Román et al., 2013; Arruda et al., 2015), as well as at aspect-level (Agerri et al., 2013; Villena-Román et al., 2015; Almeida et al., 2015). The co-official languages, however, have almost none.

We are aware of a small discourse-related sentiment corpus available in Basque (Alkorta et al., 2015), as well as a stance corpus in Catalan (Bosco et al., 2016). These resources, however, are limited in size and scope.

In this section, we describe how we created corpora which cover both Basque and Catalan languages and are annotated in such a way that they are compatible with the OpeNER datasets (Agerri et al., 2013). This resource allows us to control for domain-differences while performing research into cross-lingual sentiment analysis, as well as introducing the first resource for aspect-level sentiment analysis in Catalan and Basque. The corpora are freely available at `https://jbarnesspain.github.io/resources/`. The content of this section derives directly from the paper accepted at LREC 2018, mentioned in

Section 1.4 (Barnes et al., 2018c).

Data Collection

To collect suitable corpora, we crawl hotel reviews from www.booking.com. Booking.com allows you to search for reviews in Catalan, but it does not include Basque. Therefore, for Basque we crawled reviews from a number of other websites that allow users to comment on their stay[1].

Many of the reviews that we found through crawling are either 1) in Spanish, 2) include a mix of Spanish and the target language, or 3) do not contain any sentiment phrases. Therefore, we use a simple language identification method[2] in order to remove any Spanish or mixed reviews and also remove any reviews that are shorter than 7 tokens. This finally gave us a total of 568 reviews in Catalan and 343 reviews in Basque, collected from November 2015 to January 2016.

We preprocess them through a light normalization, after which we perform tokenization, POS-tagging and lemmatization using Ixa-pipes (Agerri et al., 2014) for Basque and Freeling (Padró and Stanilovsky, 2012) for Catalan.

Our final documents are in KAF/NAF format (Bosma et al., 2009; Fokkens et al., 2014). This is a stand-off xml format originally from the Kyoto project (Bosma et al., 2009) and allows us to enrich our documents with many layers of linguistic information, such as the POS-tag of a word, its lemma, whether it is a polar word, and if so, if it has an opinion holder or target. The advantage of this format is that we do not have to change the original text in any way.

Annotation For annotation, we adopt the approach taken in the OpeNER project (Agerri et al., 2013), where annotators are free to choose both the span and label for any part of the text.

Guidelines In the OpeNER annotation scheme[3] (see Table 2.3 for a short summary), an annotator reads a review and must first decide if there is any positive or negative attitudes in the sentence. If there are, they then decide if the sentence is

[1] We took reviews from a total of 35 different websites, including www.airbnb.com, www.atrapalo.com, www.nekatur.net, www.rentalia.es, www.toprural.es, and www.tripadvisor.com.

[2] We use the count of stopwords to predict the probability that a review is written in Spanish, Catalan, or Basque.

[3] http://www.opener-project.eu/

Is there a positive / negative attitude?	yes/no
Is the sentence on topic ?	yes/no
Is it to the point?	yes/no
IF YES TO ALL, ANNOTATE:	
What is the span of the expression?	choose span
Is the expression positive or negative?	choose
Is the expression strong?	choose
Is there an explicit target?	yes/no
If yes, what is the span?	choose span
Is there an explicit opinion holder	yes/no
If yes, what is the span?	choose span

Table 2.3: Simplified annotation guidelines.

on topic. Since these reviews are about hotels, we constrain the opinion targets and opinion phrases to those that deal with aspects of the hotel. Annotators should annotate the span of text which refers to:

- **opinion holders**,
- **opinion targets**,
- and **opinion phrases**.

If any opinion phrase is found, the annotators must then also determine the polarity of the expression, which can be STRONG NEGATIVE, NEGATIVE, POSITIVE, or STRONG POSITIVE. As the opinion holder and targets are often implicit, we only require that each review has at least one annotated opinion phrase.

For the strong positive and strong negative labels, annotators must use clues such as adverbial modifiers ('very bad'), inherently strong adjectives ('horrible'), and any use of capitalization, repetition, or punctuation ('BAAAAD!!!!!') in order to decide between the default polarity and the strong version.

Process We used the KafAnnotator Tool (Agerri et al., 2013) to annotate each review. This tool allows the user to select a span of tokens and to annotate them as any of the four labels mentioned in this section.

The annotation of each corpus was performed in three phases: first, each annotator

	Catalan	Basque
Number of Reviews	567	343
Average length in tokens	45	46.9
Number of Targets	2762	1775
Number of Expressions	2346	2328
Number of Holders	236	296

Table 2.4: Corpus Statistics

annotated a small number of reviews (20-50), after which they compared annotations and discussed any differences. Second, the annotators annotated half of the remaining reviews and met again to discuss any new differences. Finally, they annotated the remaining reviews. For cases of conflict after the final iteration, a third annotator decided between the two.

The final Catalan corpus contains 567 annotated reviews and the final Basque corpus 343.

Dataset Characteristics The reviews are typical hotel reviews, which often mention various aspects of the hotel or experience and the polarity towards these aspects. An example is shown in Figure 2.2.

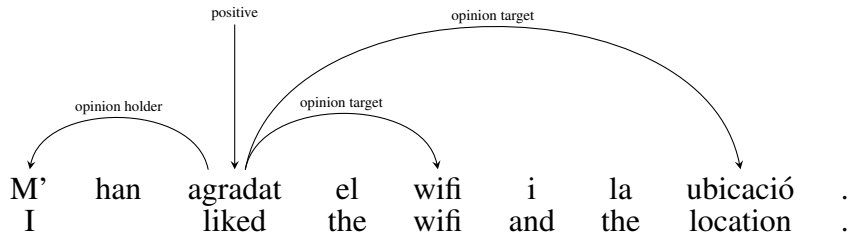


Figure 2.2: An opinion annotation following the annotation scheme detailed in this section.

Statistics for the two corpora are shown in Tables 2.4 and 2.5.

Agreement Scores Common metrics for determining inter-annotator agreement, e.g. Cohen's Kappa (Cohen, 1960) or Fleiss' Kappa (Fleiss, 1971), can not be applied when annotating sequences, as the annotators are free to choose which

		Binary		Multiclass			
		+	−	++	+	−	−−
Number of Opinions	CA	1453	883	685	808	741	142
	EU	1416	314	686	775	273	41

Table 2.5: Aspect-level statistics for the MultiBooked Catalan (CA) and Basque (EU) datasets.

parts of a sequence to include. Therefore, we use the *agr* metric (Wiebe et al., 2005), which is defined as:

$$\text{agr}(a||b) = \frac{|A \text{ matching } B|}{|A|} \tag{2.1}$$

where a and b are annotators and A and B are the set of annotations for each annotator. If we consider a to be the gold standard, agr corresponds to the recall of the system, and precision if b is the gold standard. For each pair of annotations, we report the average of the agr metric with both annotators as the temporary gold standard,

$$\text{AvgAgr}(a, b) = \frac{1}{2}\big[\text{agr}(a||b) + \text{agr}(b||a)\big] \tag{2.2}$$

Perfect agreement, therefore, is 1.0 and no agreement whatsoever is 0.0. Similar annotation projects (Wiebe et al., 2005) report $AvgAgr$ scores that range between 0.6 and 0.8 in general.

For polarity, we assign integers to each label (Strong Negative: 0, Negative: 1, Positive: 2, Strong Positive: 3). For each sentence of length n, we take the mean squared error (MSE),

$$\text{Mean Squared Error} = \frac{1}{n}\sum_{i=1}^{n}(A - B)^2 \tag{2.3}$$

where A and B are the sets of annotations for the sentence in question. This approach punishes larger discrepancies in polarity more than small discrepancies, i.e. if annotator 1 decides an opinion phrase is STRONG NEGATIVE and annotator two that the same expression is POSITIVE, this will be reflected in a larger MSE score than if annotator 2 had chosen NEGATIVE. Perfect agreement between annotators would lead to a MSE of 0.0, with the maximum depending on the

	Catalan	Basque
Number of Reviews	567	343
Targets	.767	.739
Expressions	.716	.714
Holders	.121	.259
Polarity (MSE)	1.53	2.7

Table 2.6: Inter-annotator agreement scores. *AvgAgr* score is reported for targets, expressions and holders and averaged mean squared error is reported for polarity.

length of the phrase. For a phrase of ten words, the worst MSE possible (assuming annotator 1 labeled all words STRONG POSITIVE and annotator 2 labeled them STRONG NEGATIVE) would be a 9.0. We take the mean of all the MSE scores in the corpus.

Inter-annotator agreement is reported in Table 2.6.

The inter-annotator agreement for target and expressions is high and in line with previous annotation efforts (Wiebe et al., 2005), given the fact that annotators could choose any span for these labels and were not limited to the number of annotations they could make. This reflects the clarity of the guidelines used to guide the annotation process.

The agreement score for opinion holders is lower and stems from the fact that there were relatively few instances of explicit opinion holders. Additionally, Catalan and Basque both have agreement features for verbs, which could be considered an implicit mention of the opinion holder. This is not always clear, however. Finally, the mean squared error of the polarity scores shows that annotators generally agree on where and which polarity score should be given. Again, the mean squared error in this annotation scheme requires both annotators to choose the same span and the same polarity to achieve perfect agreement.

Difficult Examples

During annotation, there were certain sentences which presented a great deal of problems for the annotators. Many of these are difficult because of 1) **nested opinions**, 2) **implicit opinions reported only through the presence or absence of certain aspects**, or 3) **the difficulty to identify the span of an expression**. Here, we give examples of each difficulty and detail how these were resolved during the annotation process.

(3) Hotela bikaina zen , nahiz eta bertako langileak ez
Hotel.ABS.SG great.ABS.SG be , although there.from workers.ABS.PL not
bereziki jatorrak izan.
particularly friendly.ABS.PL were
'The hotel was great, although the workers there were not particularly friendly.'

In the Basque sentence in Example 3, we can see that there are two distinct levels of aspects. First, the aspect 'hotel', which has a positive polarity and then the sub-aspect 'workers'. We avoid the problem of deciding which is the opinion target by treating these as two separate opinions, whose targets are 'hotel' and 'workers'.

(4) Igerilekua zegoen.
pool.ABS.SG was
'There was a pool.'

If there was an implicit opinion based on the presence or absence of a desirable aspect, such as the one seen in Example 4, we asked annotators to identify the phrase that indicates presence or absence, i.e. 'there was', as the opinion phrase.

(5) Langileek emandako arreta bikaina zen .
workers.ERG.PL given.COMP attention.ABS.SG excellent.ABS.SG was
'The attention that the staff gave was excellent.'

Finally, in order to improve overlap in span selection, we instructed annotators to choose the smallest span possible that retains the necessary information. Even after several iterations, however, there were still discrepancies with difficult examples, such as the one shown in Example 5, where the opinion target could be either 'attention', 'the attention', or 'the attention that the staff gave'.

Benchmarks

In order to provide a simple baseline, we frame the extraction of opinion holders, targets, and phrases as a sequence labeling task and map the NAF tags to BIO tags for the opinions in each review. These tags serve as the gold labels which will need to be predicted at test time. We also perform classification of the polarity of opinion phrases.

For the extraction of opinion holders, targets, and expressions we train a Conditional Random Field[4] (CRF) on standard features for supervised sequence labeling

[4]We use the implementation available in *sklearn_crfsuite*.

	Catalan	Basque
Targets	.64	.57
Expressions	.52	.54
Holders	.56	.54
Polarity	.80	.84

Table 2.7: Weighted F_1 scores for extraction of opinion targets, expressions and holders, as well as the weighted F_1 for classification of polarity.

(word-, subword-, and part-of-speech information of the current word and previous words). For the classification of the polarity of opinion phrases, we use a Bag-of-Words approach to extract features and then train a linear SVM classifier[5]

For evaluation, we perform a 10-fold cross-validation with 80 percent of the data reserved for training during each fold. For extraction and classification, we report the weighted F_1 score. The results of the benchmark experiment (shown in Table 2.7) show that these simple baselines achieve results which are somewhat lower but still comparable to similar tasks in English (Irsoy and Cardie, 2014). The drop is not surprising given that we use a relatively simple baseline system and due to the fact that Catalan and Basque have richer morphological systems than English, which were not exploited.

Conclusion

While the inter-annotator agreement for the opinion holders does not reach a satisfactory level, the agreement for opinion targets, opinion phrases, and polarity are good. The fact that a simple CRF trained on the annotated data gets relatively good F_1 scores indicates that they contain a useful source of information. Other research in emotion detection shows that despite low inter-annotator agreement scores, datasets can still be useful (Schuff et al., 2017). In fact, joint modeling of all three variables (opinion targets, expressions, and holders) has been shown to improve F_1 scores for emotion analysis (Kim and Klinger, 2018).

These datasets will serve as a test bed for the approaches proposed later in the thesis.

[5]We use the liblinear implementation from *sklearn*.

2.1.3 Other datasets

These datasets are used as test data in experiments in Chapter 4, but are less important to the thesis than the two datasets previously mentioned. Further details about these datasets are given in the relevant methodological sections.

The **Stanford Sentiment Treebank** is a dataset of English movie reviews which have been annotated for sentiment at each node of a parse tree (Socher et al., 2013).

The **Sentube dataset** consists of youtube comments in English and Italian annotated for sentiment, as well as other phenomena (Severyn and Moschitti, 2015). We only make use of the English dataset.

The **SemEval 2013 Tweet-based dataset** are English tweets annotated for sentiment (Nakov et al., 2013).

The **SemEval 2016 Tweet-based dataset** are English tweets annotated for sentiment (Nakov et al., 2016).

The **SemEval 2016 Aspect-based datasets** are English and Spanish tweets annotated for sentiment at aspect-level (Pontiki et al., 2016).

The **USAGE dataset** are English and German product-reviews annotated for sentiment at aspect-level (Klinger and Cimiano, 2014).

The **Amazon Domain Sentiment datasets** are English product reviews annotated for sentiment at review-level in four different domains (Books, DVDs, Electronics,)(Blitzer et al., 2006).

2.2 Corpora

In this section, we describe the general corpora which are used in several of the experiments in order to create the monolingual word embeddings in Chapter 4. This does not imply that they are the only corpora used and in the methodology sections of each experiment, we will make it clear which resources were used. However, as these corpora are used repeatedly, we describe them here and refer to them later.

	Spanish	Catalan	Basque
Sentences	23 M	9.6 M	0.7 M
Tokens	610 M	183 M	25 M
Embeddings	0.83 M	0.4 M	0.14 M

Table 2.8: Statistics for the Wikipedia corpora and monolingual vector spaces.

2.2.1 Wikipedia Corpora and Embeddings

For all of the monolingual embeddings, we require relatively large monolingual corpora to serve as input for the embedding algorithms. The easiest, largely accessible, completely comparable corpora is Wikipedia.

We download 2016 Wikipedia dumps for Spanish, Catalan and Basque and remove HTML markup with Wikiextractor[6]. We then perform sentence- and word-tokenization with Freeling (Padró and Stanilovsky, 2012) in the case of Spanish and Catalan or IXA pipes (Agerri et al., 2014) for Basque and finally lowercased all sentences. We do not lemmatize or perform any further tagging upon the corpora, as the input for the algorithms that we use do not require any further processing.

For all of the experiments we require monolingual vector spaces. For English, we use the publicly available GoogleNews vectors[7]. For Spanish, Catalan, Basque, and German we train skip-gram embeddings using the Word2Vec toolkit with 300 dimensions, subsampling of 10^{-4}, window of 5, negative sampling of 15. The statistics of the Wikipedia corpora and embeddings are shown in Table 2.8.

2.3 Projection Lexicons

The bilingual embedding methods introduced in Chapter 4 require a bilingual lexicon. We use the sentiment lexicon from Hu and Liu (2004) (to which we refer in the following as Hu and Liu) and its translation into each target language. We translate the lexicon using Google Translate and exclude multi-word expressions.[8] This leaves a dictionary of 5700 translations in Spanish, 5271 in Catalan, and 4577 in Basque. We set aside ten percent of the translation pairs as a development set in

[6] `http://attardi.github.io/wikiextractor/`

[7] `https://code.google.com/archive/p/word2vec/`

[8] Note that we only do that for convenience. Using a machine translation service to generate this list could easily be replaced by a manual translation, as the lexicon is comparably small.

order to check that the distances between translation pairs not seen during training are also minimized during training.

We also translate the NRC hashtag sentiment lexicon (Mohammad et al., 2013) to Spanish, which gives a much larger bilingual lexicon (22985 translation pairs).

2.4 Tools

The work presented in this thesis is built on the backs of those who have done a great deal of preliminary work. Without the tools presented in this section, most of the work in the thesis would have been impossible or desperately slow.

2.4.1 Freeling

Freeling (Padró et al., 2010) is a C++ library for natural language processing developed at the Polytechnic University of Catalonia. It performs sentence- and word-tokenization, lemmatization, morphological analysis, POS-tagging, parsing, word sense disambiguation, and semantic role labeling. It is available for a number of languages, including English, Spanish, Portuguese, Italian, French, German, Russian, Catalan, Galician, Croatian, and Slovene.

We use this library for sentence- and word-tokenization for Catalan in all experiments.

2.4.2 Ixa pipes

Ixa pipes (Agerri et al., 2014) is a Java library for natural language processing developed at the University of the Basque Country. It performs sentence- and word-tokenization, lemmatization, morphological analysis, POS-tagging, parsing, named entity recognition, and chunking. It is available for a number of languages, including English, Spanish, Basque, Dutch, Galician, German, Italian, and French.

We use this library for sentence- and word-tokenization for English, Basque and Galician in all experiments.

2.4.3 Natural Language Toolkit

The Natural Language Toolkit (NLTK) (Loper and Bird, 2002) is a python platform for performing and teaching natural language processing. It contains corpora, annotated datasets, as well as pretrained models for sentence- and word-tokenization in a number of languages, including English, Norwegian, Swedish, and Danish.

We use this library for sentence- and word-tokenization for Norwegian, Swedish, and Danish in Section 4.5.3.

2.4.4 Theano

Theano (Theano Development Team, 2016) is a deep learning framework that was created in collaboration with the University of Montreal to allow users to create deep learning architectures by providing automatic differentiation of arbitrary functions. This framework allows a researcher to quickly prototype models without having to worry about the implementation details. We used this framework for the experiments mentioned only in Section 4.2.

It has since been discontinued.

2.4.5 Keras

Keras (Chollet, 2015) is a high-level deep learning framework that builds on top of either Theano or Tensorflow. It allows for rapid prototyping of the most common variations of deep learning models, but has the disadvantage that any large modifications require a larger investment of developing time. As it creates a static graph that cannot be updated, padding training examples of differing lengths is necessary for training.

We use Keras for determining the state of the art for monolingual sentiment models in Section 4.1.

2.4.6 Pytorch

Pytorch (Paszke et al., 2016) is a highly flexible machine learning framework which has two main advantages over the previously mentioned deep learning frameworks: (1) it creates the computational graph dynamically, and (2) it is highly compatible with Numpy (Oliphant, 2006) syntax. The first advantage means that each batch of

training examples does not need to have the same length, effectively reducing the amount of preprocessing necessary to train a model, while the second means that it is fast to learn if one knows Numpy.

We use Pytorch for the experiments in Sections 4.3, 4.4, 4.5, and 4.6.

2.4.7 Sklearn

Scikit Learn (Pedregosa et al., 2011) is a general machine learning framework available for Python that includes a number of implementations of supervised and unsupervised algorithms, including Support Vector Machines and Logistic Regression for classification. These two algorithms are used as baselines in many of the experiments throughout the thesis.

2.4.8 Word2Vec

The word2vec toolkit[9] is a well known word embedding toolkit which implements the Continuous Bag-of-words and Skip-gram embeddings. It is written in an optimized C code, which makes it possible to quickly train word embeddings. We use embeddings created with this toolkit in all experiments in the thesis.

2.5 Evaluation Metrics

As for any empirical approach to natural language processing, it is important to choose evaluation metrics that allow us to quickly and correctly identify models which perform well. While accuracy has often been used in the past, this metric is not as informative when there is a class imbalance, as we have seen in our data. Therefore, we will mainly consider the macro-averaged F_1 score (macro F_1).

F_1 score is the harmonic mean of Precision and Recall scores.

Precision (Pr) measures how accurate a model's predictions are by comparing the ratio of true positives (TP) to all of it's predictions ($TP + FP$),

$$Pr = \frac{TP}{TP + FP} \tag{2.4}$$

[9] `https://code.google.com/archive/p/word2vec/`

while simultaneously ignoring any false negatives.

Recall (Rec) on the other hand looks at the coverage of a model. Of all of the specific data points found in the data, recall calculates how many the model correctly identifies.

$$Rec = \frac{TP}{TP + FN} \tag{2.5}$$

The F_1 score calculates the harmonic mean of these two measures as a way of simultaneously testing both the accuracy of a model's predictions as well as its coverage.

$$F_1 = 2 \cdot \frac{Pr \cdot Rec}{Pr + Rec} \tag{2.6}$$

In a multiclass setup, however, it is possible to calculate the F_1 score either by macro- or micro-averaging. These two approaches have different implications, depending on your data and what you want to know.

The macro-averaged F_1 first finds the individual F_1 score for each class $c \in C$ and then averages these to calculate the final score.

$$\text{macro } F_1 = \sum_{c}^{C} \frac{F_1 c}{|C|} \tag{2.7}$$

This gives the same importance to all classes, regardless of the number of examples found in each.

The micro-averaged F_1 instead combines all true positives, false positives, and false negatives from all the classes and then calculates F_1 with these. Here, the number of examples in a class is a factor, as smaller classes contribute less to the overall F_1 score.

In this thesis, we prefer to use the macro-averaged F_1, as we are interested in knowing how well models perform on the minority classes, as well as the majority.

2.6 Statistical Significance Testing

Another important aspect of comparing two or more models is to determine whether differences between model outputs are due to real differences or if they are simply

noise. Luckily, there are several good surveys for determining the best practices for statistical significance testing in natural language processing (Yeh, 2000; Søgaard et al., 2014; Dror et al., 2018) which we can draw from.

The basic notion of statistical significance testing in natural language processing is to establish a null hypothesis which assumes that there is no difference between two models. You then compare the two distributions using either parametric or non-parametric methods (Dror et al., 2018) and if the difference is more than a certain threshold, known as a p-value (usually 0.05 or 0.01), you can dismiss the null hypothesis and confirm that there is a statistically significant difference between the two models.

Non-parametric approaches do not require you to assume a certain distribution in the data, which is helpful as often in natural language processing this distribution is not known. Approximate randomization testing (Noreen, 1989) is a non-parametric approach to statistical significance testing, which uses computationally intensive randomization testing. With this test, the null hypothesis supposes that there is no real difference between the output of two models, so any output produced by one model could just as easily come from the other model. In this case, we shuffle the outputs and randomly assign each response to one of the models (each being equally likely) and then determine what difference this causes in our metric of interest (normally macro F_1 in this paper). We perform this procedure n times and the percentage of the tests where the difference between metrics is greater than the original difference is the probability of statistical difference.

The one disadvantage of approximate randomization tests is that they are computationally expensive. However, since all of the test data we use in this thesis are relatively small, this is an acceptable trade-off.

Chapter 3

STATE OF THE ART

Sentiment analysis is an immensely popular task, which has led to a proliferation of research and publishings. In a single conference, ACL 2018 for example, there were nearly 90 submissions in the sentiment and opinion mining track (Committee, 2018). With such a large interest and large number of publications on the topic, it is nearly impossible to read all papers. During the course of the thesis, we have concentrated our reading efforts on machine learning and cross-lingual methods. Although we will try to broadly cover most relevant areas of sentiment analysis in the present review of the literature, this bias will be reflected to some extent.

The remainder of this section is divided into state-of-the-art approaches to monolingual sentiment analysis, cross-lingual sentiment analysis, and distributional semantics.

3.1 Monolingual Sentiment Analysis

3.1.1 Knowledge-based Approaches

Knowledge-based approaches to sentiment analysis work on the premise that words, especially adjectives (but also nouns, verbs, and adverbs), can be clustered into groups of the same **semantic orientation**. **Semantic orientation** "refers to the polarity and strength of words, phrases, or texts" (Taboada et al., 2011). Most of these researchers start by creating a **sentiment lexicon**, which is a list of words with their prior polarity. These can either be categorical, *i. e.*, positive or negative, or real valued, *i. e.*, a number from -1 to 1 where -1 is the most negative and 1 is the most positive.

Early work on semantic orientation laid the groundwork for sentiment analysis as we know it today. The research first concentrated on the semantic orientation of single words, using distributional features (Hatzivassiloglou and McKeown, 1997), Pointwise Mutual Information of co-occurrence with known polar words (Turney and Littman, 2003), information from WordNet (Kamps et al., 2004; Esuli and Sebastiani, 2006), and the glosses from dictionaries (Esuli and Sebastiani, 2005).

Research has also investigated the prediction of the semantic orientation of phrases and sentences. Turney (2002) proposed a mutual information approach to determine the semantic orientation of phrases in a review. The author used the co-occurrence patterns with clearly polar words, in this case "excellent" and "poor", and determined the average semantic orientation of the phrases in the review. Hu and Liu (2004) create feature summaries for reviews by finding subjective sentences that refer to a feature of a product, and determining the polarity of these sentences using a sentiment lexicon.

More recently, knowledge-based sentiment analysis has moved towards incorporating linguistic elements, such as negation, diminishers, and intensification, into the final classification. Early attempts at handling negation simply reversed the polarity (Hu and Liu, 2004) or used special negated versions of words (Das and Chen, 2007). Polanyi and Zaenen (2006) propose a method that takes both negation and intensification into account by changing the prior semantic orientation of words in context. Taboada et al. (2011) demonstrated that their lexicon-based system, which incorporated intensification and negation, performed well across domains and unseen datasets.

While knowledge-based approaches provide the foundation and motivation for sentiment analysis, we are aware of no sentiment analysis subtask where knowledge-based approaches by themselves achieve state-of-the-art results. This is best summed up by stating "[a] sentiment lexicon is necessary but not sufficient for sentiment analysis" (Liu, 2012).

3.1.2 Machine-learning Approaches

Machine learning is a statistical approach to classification, as well as other tasks, that uses annotated data to learn to model a phenomenon. Background knowledge on machine learning, *i. e.* training, objective functions, and inference, is assumed in this section. For the reader interested in finding more details, we suggest Mitchell (1997) or Bishop (2006).

Pang et al. (2002) proposed the first machine-learning based approach to sentiment analysis, as well as creating a dataset that is still one of the benchmarks for sentiment analysis. They crawled positive and negative reviews from the Internet Movie Database (IMDb), and compared the use of Naive Bayes, Maximum Entropy, and Support Vector Machine classifiers, finding that machine learning approaches outperform human-generated baselines and that Support Vector Machines generally gave the best results.

Early work following this first approach looked into sentiment-specific feature engineering, such as the use of low-frequency words (Wiebe et al., 2004; Yang et al., 2006), positional information (Kim and Hovy, 2006), N-gram information (Pang et al., 2002; Dave et al., 2003), part of speech tags (Mullen and Collier, 2004; Whitelaw et al., 2005), and negation features (Na et al., 2004; Wilson et al., 2005; Kennedy and Inkpen, 2006; Das and Chen, 2007; Zhu et al., 2014).

The main findings were that Support Vector Machines generally gave the best results for supervised classification, and that unigrams were the single most important feature for determining the polarity of a text. These unigrams, however, were highly domain- and even corpus-dependent (Aue and Gamon, 2005). While lexicon-based methods at the time were better able to handle domain discrepancies, recent approaches using neural networks and domain adaptation techniques (Blitzer et al., 2007; Bollegala et al., 2014; Ziser and Reichart, 2017) or transfer learning (Ruder and Plank, 2017) have proven more effective.

Given that all current state-of-the-art methods rely on machine learning to some degree, in this thesis we will concentrate only on the machine learning approaches.

3.1.3 Neural Networks

Neural networks are a family of machine learning algorithms that have become omnipresent in sentiment analysis in the last five years. They reach state-of-the-art results on almost all tasks and datasets which are commonly used for testing. A full overview of neural networks is beyond the scope of this thesis, but the reader may refer to Goodfellow et al. (2016) for further details. Therefore, we will concentrate only on models used in the experiments in Chapter 4.

Neural networks, like all machine learning algorithms, require large amounts of training data in order to generalize well. However, since they have a larger number of free parameters, they are more prone to overfitting than linear models such as SVMs or maximum entropy models. Regularization and early stopping are often used to counteract this problem.

Feed forward neural networks: Feed forward neural networks (FF) are the first and simplest kind of neural network. Information flows from the input layer, through an optional number of hidden layers, finally coming to the output layer, which has a dimensionality of the number of classes k that we want to classify. A general figure of a deep FF network (Feedforward Neural Networks, 2018) is shown in Figure 3.1.

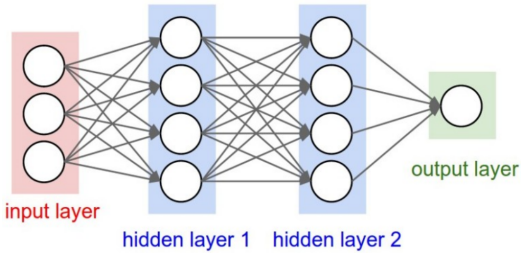


Figure 3.1: Depiction of a feed forward neural network. Taken from Feedforward Neural Networks (2018).

More formally we can define a FF network with a single hidden layer as:

$$Z = \sigma(W_1^\intercal X + b_1) \tag{3.1}$$
$$\hat{y} = \sigma(W_2^\intercal Z + b_2) \tag{3.2}$$

where W_i and b_i are the weight matrix and bias of layer i, Z is the hidden representation, and σ is a non-linearity function. Common choices for non-linearities include:

- sigmoid$(x) = \frac{1}{1+e^{-x}}$
- tanh$(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$
- rectified linear unit$(x) = max(0, x)$
- softmax$(x) = \frac{e^x}{\sum_{i=1}^{K} e^x}$

For the non-linearity of the hidden layers, the most commonly used non-linearity is the **rectified linear unit** (RELU) (Nair and Hinton, 2010a). The advantage of the RELU is that the derivative is always linear, which helps to avoid the exploding/vanishing gradient problem in deep networks (Hochreiter et al., 2001). We normally use the softmax function in the output layer for classification, as the output is a natural probability distribution over the class labels.

During training, for each (x, y) pair in an annotated corpus, we calculate the prediction $\hat{y}$, find the loss $H(y, \hat{y})$, and calculate the derivative of each weight with

respect to the input. The weights are updated using the back propagation algorithm (Allen, 1987).

Backpropagation is a method used in deep neural networks to calculate the gradient for all of the weights given a loss function. There are two phases which alternate; propagation and weight update. During propagation, a feature vector is propagated through the network until it reaches the output layer. The error for each neuron is then calculated from its contribution to the overall loss. During the weight update, each weight is changed to minimize the loss.

Some variant of gradient descent (Cauchy, 1847) is then employed to train the full network (see Ruder (2016) for a survey of gradient descent methods). By far the most common flavor of gradient descent is mini-batch stochastic gradient descent (Bengio, 2012). In this approach, the neural network receives small batches of n training examples at a time, where n is a small subset of the training corpus N. The model finds the overall error for the n examples and updates the weights accordingly. The batch size serves as a way to speed up training, as weights are not optimized for single examples, and therefore must try to move towards a more overall optimal configuration.

Cross entropy loss is most commonly used to train a FF network, although there are other options available, such as mean squared error. For binary classification, we define cross entropy as

$$H(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \tag{3.3}$$

where $y \in \{0, 1\}$ is the true label and $\hat{y} \in [0, 1]$ is the estimated label from the softmax layer of the FF network.

Typically, regularization is needed to keep the deep network from overfitting. L_2 **regularization**, which assumes that less complex models are preferable, is a common choice. We define L_2 regularization as

$$L_2 \text{ regularization term} = ||W||_2^2 \tag{3.4}$$

where W are the weights that we would like to learn. This regularization term is added to the main objective function and effectively penalizes any model that tries to give high weights.

Dropout (Srivastava et al., 2014) has become more popular recently and has been shown to avoid co-adaptation of weights. This method randomly "disconnects" a

percentage of the neurons during training. During testing, all weights are used, after being weighted to counteract the effects of learning with half of the weights.

Researchers have found that using both L_2 regularization and dropout in conjunction is also helpful (Flekova and Gurevych, 2016; Merity et al., 2018).

The main disadvantage that FF networks have is that they do not take into account the dependencies or structure of the data. Unlike more advanced models, such as Recurrent Neural Networks or Convolutional Neural Networks, FF networks do not incorporate information about word order, either locally or globally. They also cannot easily incorporate information about hierarchical linguistic structures, such as a parse tree of a sentence, unlike Recursive Neural Networks. While in theory these hindrances should lead to a preference for advanced models, in practice properly tuned FF networks can often perform nearly as well, while requiring less time to train.

Iyyer et al. (2015) propose to use a bag-of-embeddings representation as input to a deep feed forward network. They find that, despite ignoring structural and dependency information, their model was able to perform nearly as well as more informed models. This suggests that for many text classification tasks, order is not critical for good performance.

Recurrent Neural Networks: Recurrent neural networks (RNNS) are a family of neural networks whose nodes form a directed graph across a sequence (Rumelhart et al., 1986). RNNS introduce the crucial time element t, which allows this class of models to process any kind of data that can be represented sequentially.

There are a number of models within this family, but in this thesis we focus primarily on the **Long Short-Term Memory network** (LSTM) (Hochreiter and Schmidhuber, 1997) and **Bidirectional Long Short-Term Memory network** (BILSTM) (Schuster and Paliwal, 1997). A typical LSTM cell is shown in Figure 3.2.

An LSTM takes three inputs, the input at time t (x_t), the previous hidden state (C_{t-1}), and the previous output (h_{t-1}). It also has three **gating mechanisms**, a **forget gate** (f), an **input gate** (i) and an **output gate** (o), which allow information to pass through the model in time, depending on it's importance for the task. These gates each have a weight matrix and bias which allows the model to learn how much of this information to incorporate and pass through. More formally, an LSTM is defined as

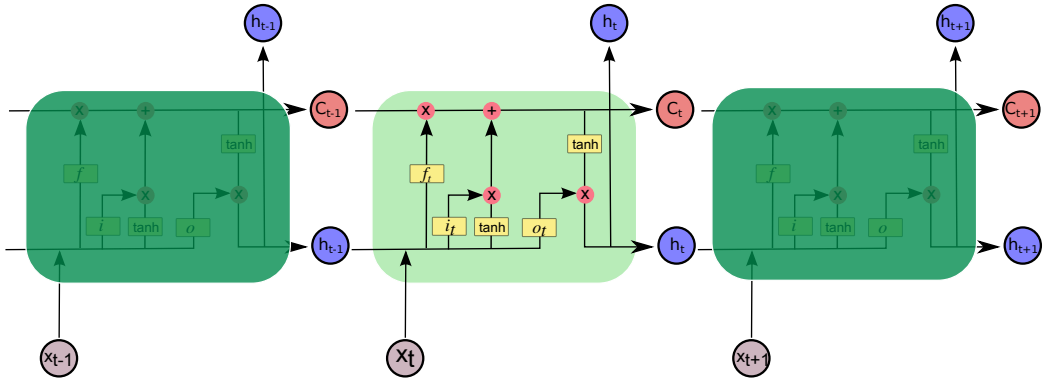


Figure 3.2: Depiction of a Long Short-Term Memory neural network. Based on that of Olah (2018).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.5}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.6}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.7}$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.8}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.9}$$
$$h_t = o_t * tanh(C_t) \tag{3.10}$$

where $\tilde{C}_t$ is the candidate hidden state.

Bidirectional LSTMS (Schuster and Paliwal, 1997) are an extension of LSTMS where you process the sequence once from left to right, and simultaneously from right to left. The two hidden representations can then be concatenated to give a final representation for classification. This allows the network to incorporate information from both left and right contexts to help classification.

LSTMS and BILSTMS are one of the most effective sentiment classifiers and have been successfully used for sentiment analysis (Tai et al., 2015; Barnes et al., 2017), and emotion analysis (Felbo et al., 2017).

Convolutional Neural Networks: Convolutional Neural Networks (CNNS) are a family of neural networks that were originally designed within the image recognition community. They have shown great success on many image recognition tasks, and more recently have been applied to Natural Language Processing tasks.

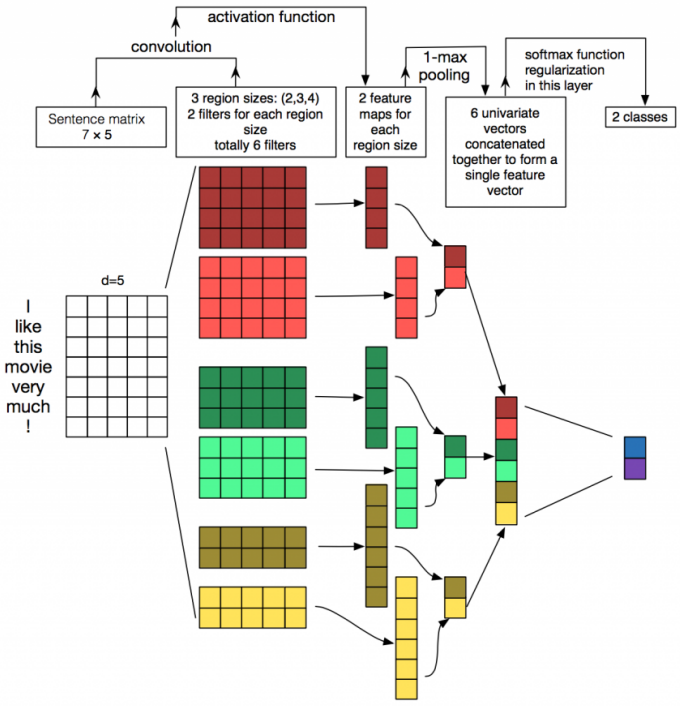


Figure 3.3: Depiction of a Convolutional neural network. Taken from (Zhang and Wallace, 2017)

CNNS are parameterized with a **kernel** or **filter**, which is a filter matrix smaller than the dimensionality of the input matrix ($\mathbb{R}^{2\times d,3\times d,4\times d}$ in the example in Figure 3.3 where d is the dimensionality of the embedding). This kernel operates in a sliding fashion over the input, creating a **feature map**, which is the output of the **convolution** on the input data. For NLP, the input is normally a set of word embeddings, which have been concatenated and padded where necessary to form a matrix. After the convolution, a non-linearity, such as RELU, is applied to each of the feature maps. Finally, there is a **pooling** step, which reduces the dimensionality of the representation by taking only the maximum or average of the pooling region. A 1-max pooling, as shown in Figure 3.3, reduces a feature vector to a vector of length 1 which takes the maximum of the original feature vector. This allows the model a certain amount of resilience to local variation.

We can think of a CNN for Natural Language Processing as a parameterized N-gram model, where the size of the kernel is equivalent to N. In practice, several kernel sizes are normally employed, which gives CNNS an advantage over other algorithms that use N-grams as input. A typical CNN for text classification is

depicted in Figure 3.3 (taken from Zhang and Wallace (2017)).

Convolutional Neural Networks (CNNS) have been applied to a number of sentiment tasks. Kim (2014) first proposed CNNS for text classification, achieving state-of-the-art results on 4 out of 7 text classification tasks. Since then, CNNS have been successfully used for classifying sentiment in tweets (Severyn and Moschitti, 2015), and short texts (dos Santos and Gatti, 2014), as well as emotions on a number of datasets (Felbo et al., 2017).

Recursive Neural Networks: Socher et al. (2013) introduced a Recursive Neural Tensor Network (RNTN) which better accounted for compositional semantics in sentiment when trained on a parsed version of the dataset from Pang and Lee (2005), where there is a sentiment label for each node. This model pushed the state of the art up to 85.4% accuracy for binary classification, outperforming feature-based baselines, as well as strong sequence-modeling baselines such as Long Short-term Memory networks (LSTMS) (Hochreiter and Schmidhuber, 1997). A depiction of a recursive neural network is shown in Figure 3.4 (Socher et al., 2013).

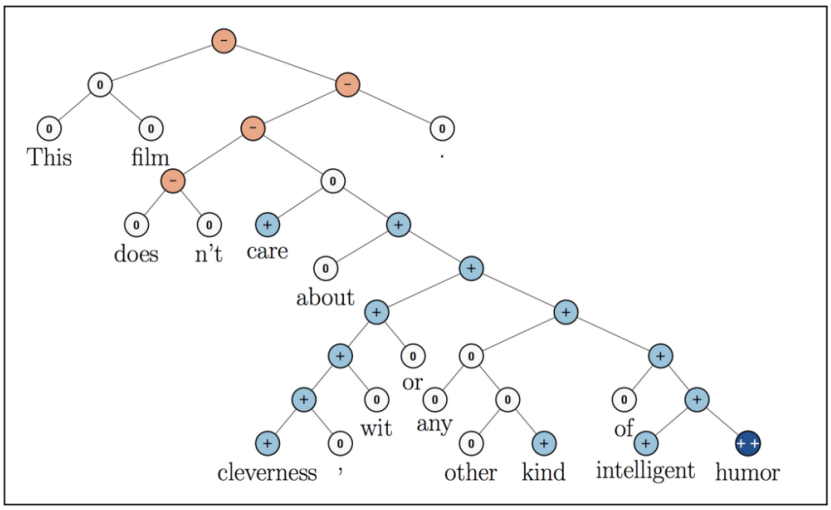


Figure 3.4: Depiction of a recursive neural network, taken from Socher et al. (2013).

Successive work built upon the idea of using tree-structured data, introducing new models, such as Deep Recursive Networks (Irsoy and Cardie, 2014), and Tree-structured LSTMS (Tai et al., 2015).

While recursive neural networks are well motivated and powerful models for sentiment analysis, they impose data requirements which are unreasonable for

under-resourced languages. For good results, you need a dataset which has been previously parsed and which is also annotated for your task at each node of the parse tree for each sentence in your training set. The only dataset annotated in such a manner is the one released by Socher et al. (2013). This means that this technique is not easily transferable to other languages.

3.1.4 Aspect-Level Sentiment Analysis

Sentence- and document-level sentiment analysis are not able to identify exactly what people like or dislike, as they only determine the polarity of a sentence or document, ignoring the opinions contained therein (Liu, 2012). In order to truly analyze the sentiment contained in these texts, it is necessary to take all opinions into account.

Aspect-level sentiment analysis (ABSA), also known as **aspect-based** or **targeted sentiment analysis**, is a fine-grained view of sentiment analysis that attempts to encounter all opinions in a text. Liu (2012) defines an opinion as a quintuple (e, t, s, h, v) where e is an entity, t is an optional aspect of the entity, s is the sentiment towards t, h is the holder of the opinion, and v is the time at which the sentiment is expressed. An **aspect** can be an entity, such as a product, service, person or event, or one of its subcomponents, such as the service or food in a restaurant (Liu, 2012). In this thesis we disregard the time v from our setup, and consider the aspect of the entity as the target to classify, if there is one. Otherwise, we consider the entity itself. This is a simplification which is common in sentiment analysis (Pontiki et al., 2014, 2015, 2016; Zhang et al., 2016) and allows us to work with triples of (t, s, h), where t is either the aspect or, in the absence of an aspect, the entity, instead of the original quintuples.

ABSA differs from sentence- and document-level sentiment analysis in several respects:

1. It is necessary to determine the aspects, either by deciding which aspects we are interested in beforehand or by finding all possible aspects.
2. There is less redundancy of sentiment information. A single word often conveys all of the sentiment towards an aspect.
3. Many aspects are mentioned only implicitly.

These differences make aspect-level sentiment analysis more difficult than document- or sentence-level sentiment analysis.

There are several possible task formulations for aspect-level sentiment analysis

and they can often be different from one another. We can define four formulations, which move from the simplest to the most complicated:

Open-set aspect classification: In this setup, an open set of aspects must be classified (Lambert, 2015). These aspects are assumed to be extracted separate from the classification step.

Closed-set aspect classification: In this setup, a small set of pre-selected aspects must be classified for polarity. While the aspects are pre-selected, they are not always explicitly mentioned in the text, which requires a separate step for resolution. This is often the format used for shared tasks on ABSA, such as the SemEval tasks (Pontiki et al., 2014, 2015, 2016).

Targeted: In this setup, the classifier must identify the polarity of an open set of targets, which are often entities (Zhang et al., 2015, 2016). These approaches attempt to identify the target and its polarity jointly or in a pipeline fashion.

Full opinions: For some tasks (Klinger and Cimiano, 2013; Agerri et al., 2013; Barnes et al., 2018a), the authors attempt to extract opinion targets, opinion phrases, and define the relationship between them.

Previous successful approaches to ABSA relied on **conditional random fields** (Lafferty et al., 2001) (Yang and Cardie, 2013; Klinger and Cimiano, 2013; Zhang et al., 2015), external knowledge bases (Mohammad et al., 2013), or by incorporating target-specific information (Jiang et al., 2011).

More recently, approaches for aspect level sentiment analysis have moved to neural networks as well: Dong et al. (2014) focus on integrating target information into recursive neural networks (Socher et al., 2013). They use dependency trees, which they then convert to have the target as the root node. They show improvements over support vector machines and more general architectures of recursive neural networks.

Zhang et al. (2015) extend CRF models by using neural networks to extract features. They show an increase in recall and F_1, while the traditional CRFs have higher precision. Zhang et al. (2016) extend this work using gated recurrent networks (Cho et al., 2014) and show improvements on the concatenation of three datasets.

Tang et al. (2016) propose a technique to perform ABSA with Long Short-Term Memory networks (LSTMS) (Hochreiter and Schmidhuber, 1997). They notice that the LSTM often 'forgets' the aspect words by the final timestep and propose to split the sentence at the aspect and use two LSTMS to create separate representations of the sentence for the left and right context. This ensures that the two LSTMS keep the relevant aspect information, as it is the last input before classification.

Sentiment can often be aspect-specific, i.e. "the battery life is long" is positive but "the movie is long" is negative. More recent approaches attempt to augment a neural network with memory to model these interactions (Chen et al., 2017; Xue and Li, 2018; Wang et al., 2018; Liu et al., 2018). Wang et al. (2017) explore methods to improve classification of multiple aspects in tweets, while Akhtar et al. (2018) attempt to use cross-lingual and multilingual data to improve aspect-based sentiment analysis in under-resourced languages.

3.2 Cross-lingual Sentiment Analysis

The lack of available data in most of the world's languages means that cross-lingual methods are of great interest. Statistical machine translation first enabled cross-lingual sentiment analysis and since has become a staple of most cross-lingual approaches. Although there are ways to perform machine translation between resource-rich languages and many under-resourced languages, *i. e.* triangulation (Cohn and Lapata, 2007), multilingual neural machine translation (Johnson et al., 2017; Lakew et al., 2018), or unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018b), the reality is that most languages are still left without reliable machine translation.

In this section we provide an overview of relevant literature which motivates the research goals of the thesis.

3.2.1 Using Machine Translation

Early work in cross-lingual sentiment analysis found that machine translation (MT) had reached a point of maturity that enabled the transfer of sentiment across languages. Researchers translated sentiment lexicons (Mihalcea et al., 2007; Meng et al., 2012) or annotated corpora and used word alignments to project sentiment annotation and create target-language annotated corpora (Banea et al., 2008; Duh et al., 2011; Demirtas and Pechenizkiy, 2013; Balahur and Turchi, 2014).

Annotation projection is a widely-used approach to cross-lingual tasks using machine translation or parallel corpora. The basic idea is to take a corpus which has been annotated for a specific task and project the labels from the source to the target corpus. This can either be done by tagging a parallel corpus, or by machine translating the source corpus to the target language. It has proven a robust approach across a number of tasks, such as POS-tagging (Buys and Botha, 2016),

Dependency Parsing (Agić et al., 2016), semantic role labeling (Padó and Lapata, 2009) or relationship extraction (Faruqui and Kumar, 2015).

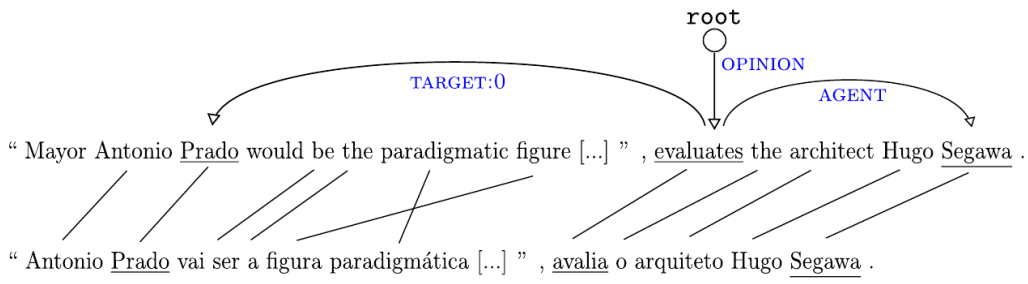


Figure 3.5: A depiction of annotation projection of sentiment annotations for aspect-level sentiment analysis using word alignments. Figure taken from (Almeida et al., 2015).

Several approaches included a multi-view representation of the data (Banea et al., 2010; Xiao and Guo, 2012) or co-training (Wan, 2009a; Demirtas and Pechenizkiy, 2013) to improve over a naive implementation of machine translation, where only the translated data is used. There are also approaches which only require parallel data (Meng et al., 2012; Zhou et al., 2016; Rasooli et al., 2017), instead of machine translation.

Another approach is to create a bilingual view of the data. The essence of this approach is to reduce the noise that translation introduces by presenting classifiers with complementary views. Wan (2009b) creates a bilingual representation of the data through SMT and then uses co-training to take advantage of classifiers that commit complementary errors. This research seems promising, but the benefits of co-training may have more to do with incorporating in-domain data than cross-lingual transfer (Demirtas and Pechenizkiy, 2013).

Pan et al. (2011) use a bi-view non-negative matrix tri-factorization approach which allows for the incorporation of sentiment lexicon information. Lu et al. (2011) incorporate a joint bilingual model which makes use of unlabeled parallel or pseudo-parallel data in order to improve sentiment classification for both languages simultaneously.

3.2.2 Without Machine Translation

There are also approaches which focus on parallel data instead of machine translation. Meng et al. (2012) make use of parallel corpora to learn a cross-lingual

mixture model, while Zhou et al. (2016) also use parallel corpora and stacked denoising autoencoders to learn a bilingual representation of documents. Rasooli et al. (2017) , instead, make use of multiple annotations and bilingual word embeddings to perform CLSA. Popat et al. (2013) use parallel corpora and learn clustering algorithms to learn useful cross-lingual features.

All of these approaches, however, require large amounts of parallel data, which are not always available between the resource-rich and under-resourced languages. A notable exception is the approach proposed by Chen et al. (2016) , an adversarial deep averaging network, which trains a joint feature extractor for two languages. They minimize the difference between these features across languages by learning to fool a language discriminator, which requires no parallel data. It does, however, require large amounts of unlabeled data.

3.2.3 Aspect-level Cross-lingual Sentiment Analysis

At document and sentence level, there is often enough redundant sentiment signal to withstand a certain amount of noise. But when we move to a more fine-grained level, *i. e.*, aspect or target level, we are often confronted with the situation that sentiment towards a specific target is expressed with a single word. If this word is mistranslated or its sentiment is incorrectly inferred, there is no way to correctly predict it. That makes the combination of cross-lingual and aspect-level sentiment analysis particularly challenging and leads to developments which aim at tackling such issues.

Annotation projection is the most common technique to perform aspect-level cross-lingual sentiment analysis. Lambert (2015) notices that when using an annotation projection approach to perform aspect-level cross-lingual sentiment analysis, aspects and sentiment-bearing phrases are shuffled or moved within the sentence, leading to uninformative annotations. To reduce this, he proposes a constrained statistical machine translation model which avoids reordering of targets and sentiment expressions during translation. The classifiers trained on this SMT data achieve comparable results to their monolingual version. However, this is a state-of-the-art SMT system[1] which is not available in most language combinations.

Almeida et al. (2015) also use annotation projection, but instead introduce dependency based opinion mining, where dependency trees are used as features for a classifier. They then use word aligned parallel text to project the dependency trees from English to Portuguese and perform aspect-level sentiment analysis. This

[1]The system achieves a *BLEU* score 45.3 in Spanish-English translation with true-case.

approach outperforms a similar delexicalized approach, as well as a model trained on a small target-language annotated corpus.

Klinger and Cimiano (2015) experiment with keeping only high-quality translated examples when using MT to perform annotation projection. They include an instance selection method which filters low-quality translations and show that this leads to improvements over using the full set of noisy translations.

Unfortunately, all of the previous approaches assume there is a high-quality machine translation system available for each language pair. While machine translation systems have improved greatly, there are still many language combinations with almost no parallel data (Balamurali et al., 2013; Popat et al., 2013; Pourdamghani and Knight, 2017; Artetxe et al., 2018; Lample et al., 2018b), which complicates the creation of machine translation systems. In fact, for the language pairs which we consider in this thesis, we are only aware of one publicly available translations service (`https://translate.google.com/`) which is able to translate all.

3.3 Distributional Semantics

Distributional semantics (Harris, 1954) takes the view that a word or phrase can be defined in large part by the contexts in which it appears, especially in a large collection of sentences. Firth (1957) famously stated that "You shall know a word by the company it keeps," which still sums up the motivation for this line of research.

While early work in distributional semantics concentrated on intuitive count-based models, more recent research has moved to what can be called predictive models (Baroni et al., 2014). In this framework, instead of counting the number of times a context word co-occurs with a target-word, the model learns a probabilistic function to either predict the context-word, given the target-word, or vice versa. Other researchers have shown that these modern approaches are really just highly tuned models that implicitly perform the same functions as the co-occurrence models (Levy et al., 2015).

Current state-of-the-art methods in sentiment analysis rely on features extracted in an unsupervised manner, mainly through one of the existing pre-trained word embedding approaches (Collobert et al., 2011; Mikolov et al., 2013a; Pennington et al., 2014). These approaches to sentiment analysis represent words as some function of their contexts, enabling the machine learning algorithms to generalize over tokens that have similar representations, arguably giving them an advantage over

previous approaches, such as bag-of-words, unigram, or bigram representations of text.

3.3.1 Monolingual Embeddings

In this subsection we will introduce the word embedding models used in experiments in Chapter 4. We concentrate primarily on the Skip-gram algorithm, but also introduce Global Vectors (GLoVe) and sentiment embeddings.

Skip-gram

Skip-gram is an approach to creating semantic word vectors which can be trained on large amounts of data quickly and which retain certain regularities (Mikolov et al., 2013c). While there is a large amount of similar research in representation learning (Bengio et al., 2003; Mikolov et al., 2010), these previous techniques primarily used neural networks to learn the word representations, which made training slow.

Mikolov et al. (2013a) propose two log-linear models: the **continuous bag of words** model (CBOW) and the **Skip-gram** model. Preliminary experiments demonstrate that Skip-gram embeddings perform better when given enough data (Mikolov et al., 2013a; Levy et al., 2015), so we will concentrate on this model.

The Skip-gram model tries to predict all of the context tokens in a window using just the center token. More formally, given a sequence of training words $w_1, w_2, \ldots w_T$, the Skip-gram model attempts to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, c\neq 0}\log p(w_{t+j}|w_t) \tag{3.11}$$

where c is the size of the training window. Intuitively, for each token t in the corpus, we create a context window around this token. We then try to predict each word in the context window, given the token t.

Using a softmax over the whole vocabulary to calculate the probability of a word given a context can be prohibitive when the vocabulary is large. The authors instead propose **negative sampling**, a simplification of noise contrastive estimation, in order to eliminate the need for softmax and to reduce training time. The authors justify this choice by pointing out the difference in objectives; namely, Skip-gram

is only interested in creating high quality vector representations, not computing a full probability distribution. They therefore replace $p(w_{t+j}|w_t)$ with

$$\log \sigma(v_{wO}^{'\intercal} v_{wI}) + \sum_{i=1}^{k} \mathbb{E}\left[\log \sigma(-v_{wi}^{'\intercal} v_{wI})\right] \tag{3.12}$$

at every step. Basically, the task is reduced to distinguishing the current true vector $v_{wO}^{'}$ from other vectors taken from the noise distribution using logistic regression.

These embeddings have been empirically shown to perform well for word analogy tasks (Mikolov et al., 2013a), as well as text classification tasks (Kim, 2014; dos Santos and Gatti, 2014; Irsoy and Cardie, 2014).

Global Vectors for Word Representations (GloVe)

While the Skip-gram embedding algorithm is able to make good use of local statistics via window co-occurrence, it does not utilize the overall statistics as well as previous methods. Given that one of the interesting properties of the word2vec embeddings is their syntactic and semantic regularities,Pennington et al. (2014) attempt to model this property explicitly by combining the advantages of global matrix factorization and local context windows. They find that much of the relevant information can be determined by ratios of co-occurrence, rather than the raw counts and propose a log bilinear regression model to take advantage of this fact.

Let X be a word-word co-occurrence matrix, where $X_{i,j}$ is the co-occurrence count of words i and j. Instead of relying directly on the co-occurrence counts, they make use of the ratio of co-occurrences given a context, $P_{i,k}/P_{j,k}$, where k is a context word. Their formulation leads to a general model

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{i,k}}{P_{j,k}} \tag{3.13}$$

where w_i and w_j are the word embedding vectors for words i and j respectively, and $\tilde{w}_k$ is a context vector for the word k. Their final objective function is

$$J = \sum_{i,j=1}^{V} f(X_{i,j})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{i,j})^2 \tag{3.14}$$

where V is the size of the vocabulary and f is a weighting function that effectively downsamples frequent co-occurrences. They also include two biases (b_i and $\tilde{b}_j$) and a regularization term $\log X_{i,j}$ and reduce the main function to the dot product of the word vector w_i and the context vector $\tilde{w}_j$. .

These embeddings have been empirically shown to perform well for sentiment classification and semantic relatedness (Tai et al., 2015). However, there is currently no consensus as to whether GloVe or SkipGram embeddings perform better. Therefore, in this thesis, we will mainly use SkipGram embeddings.

Retrofitting to Semantic Lexicons

There have been several proposals to improve the quality of word embeddings using semantic lexicons. Yu and Dredze (2014) propose several methods which combine the CBOW architecture (Mikolov et al., 2013a) and a second objective function which attempts to maximize the relations found within some semantic lexicon. They use both the Paraphrase Database (Ganitkevitch et al., 2013) and WordNet (Fellbaum, 1999) and test their models on language modeling and semantic similarity tasks. They report that their method leads to an improvement on both tasks.

Kiela et al. (2015) aim to improve embeddings by augmenting the context of a given word while training a skip-gram model (Mikolov et al., 2013a). They sample extra context words, taken either from a thesaurus or association data, and incorporate this into the context of the word for each update. The evaluation is both intrinsic, on word similarity and relatedness tasks, as well as extrinsic on TOEFL synonym and document classification tasks. The augmentation strategy improves the word vectors on all tasks.

Faruqui et al. (2015) propose a method to refine word vectors by using relational information from semantic lexicons (we refer to this method as RETROFIT). They require a vocabulary $V = \{w_1, \ldots, w_n\}$, its word embeddings matrix $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_n\}$, where each $\hat{q}_i$ is one vector for one word w_i and an ontology Ω, which they represent as an undirected graph (V, E) with one vertex for each word type and edges $(w_i, w_j) \in E \subseteq V \times V$. They attempt to learn the matrix $Q = \{q_1, \ldots, q_n\}$, such that q_i is similar to both $\hat{q}_i$ and $q_j \forall j$ for $(i, j) \in E$. Therefore, the objective function to minimize is

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i ||q_i - \hat{q}_i||^2 + \sum_{(i,j) \in E} \beta_{i,j} ||q_i - q_j||^2 \right],$$

where α and β control the relative strengths of associations.

They use the XL version of the Paraphrase Database (PPDB-XL) dataset (Ganitkevitch et al., 2013), which is a dataset of paraphrases as the semantic lexicon, to improve the original vectors. This dataset includes 8 million lexical paraphrases collected from bilingual corpora, where words in language A are considered paraphrases if they are consistently translated to the same word in language B. They then test on the Stanford Sentiment Treebank (Socher et al., 2013). They train an L2-regularized logistic regression classifier on the average of the word embeddings for a text and find improvements after retrofitting.

All above approaches show improvements over previous word embedding approaches (Mnih and Teh, 2012; Yu and Dredze, 2014; Xu et al., 2014) on this data set.

Sentiment embeddings

Maas et al. (2011) first explored the idea of incorporating sentiment information into semantic word vectors. They proposed a topic modeling approach similar to latent Dirichlet allocation in order to collect the semantic information in their word vectors. To incorporate the sentiment information, they included a second objective whereby they maximize the probability of the sentiment label for each word in a labeled document.

Tang et al. (2014) take a joint training approach and simultaneously incorporate syntactic[2] and sentiment information into their word embeddings (we refer to this method as JOINT). They extend the word embedding approach of Collobert et al. (2011), who use a neural network to predict whether an n-gram is a true n-gram or a "corrupted" version. They use the hinge loss

$$\text{loss}_{\text{cw}}(t, t^r) = \max(0, 1 - f^{cw}(t) + f^{cw}(t^r)) \tag{3.15}$$

and backpropagate the error to the corresponding word embeddings. Here, t is the original n-gram, t^r is the corrupted n-gram and f^{cw} is the language model score. Tang et al. (2014) add a sentiment hinge loss to the Collobert et al. (2011) model, as

$$\text{loss}_{\text{s}}(t, t^r) = \max(0, 1 - \delta_s(t) f_1^s(t) + \delta_s(t) f_1^s(t^r)) \,, \tag{3.16}$$

where f_1^s is the predicted negative score and $\delta_s(t)$ is an indicator function that reflects the sentiment of a sentence. $\delta_s(t)$ is 1 if the true sentiment is positive and

[2]We use the authors' terminology here, but make no assumptions that the distributional representation encodes information directly pertaining to syntax.

-1 if it is negative. They then use a weighted sum of both scores to create their sentiment embeddings:

$$\text{loss}_{\text{combined}}(t, t^r) = \alpha \cdot \text{loss}_{\text{cw}}(t, t^r) + (1 - \alpha) \cdot \text{loss}_{\text{s}}(t, t^r)\,. \tag{3.17}$$

This requires sentiment-annotated data for training both the syntactic and sentiment losses, which they acquire by collecting tweets associated with certain emoticons. In this way, they are able to simultaneously incorporate sentiment and semantic information relevant to their task. They test their approach on the SemEval 2013 twitter dataset (Nakov et al., 2013), changing the task from three-class to binary classification, and find that they outperform other approaches.

Overall, creating sentiment-specific embeddings shows promise for tasks with a large amount of distantly-labeled data, and intuitively deals with the problem of antonyms having similar representations.

Open Problems

While word embedding algorithms have led to improvements in many NLP tasks, they still suffer from a number of shortcomings. One of the most relevant problems with using word embeddings for sentiment analysis is that *antonyms* often have similar representations. This problem stems directly from the distributional hypothesis, as antonyms are often found in similar local contexts. Compare the words "love" and "hate" in the following examples:

> I really loved that movie.
> I really hated that movie.

Given only the current examples, these words would have the exact same representation. In practice, this does not happen, but the word embeddings are still overly similar, which can impair classification. Previous research has addressed this problem, but this has not been used in sentiment analysis (Nguyen et al., 2016).

A second problem comes from the concept of a word in these embedding algorithms. They normally do not perform a significant amount of preprocessing, which means that they perform their calculations directly on the tokens. For morphologically rich languages, this is a problem, as the number of tokens is significantly higher than in morphologically poor languages like English. This leads to a scarcity of training examples for most words and a consequent loss in performance. There have been attempts to incorporate morphological information into word embeddings on a monolingual level (Bojanowski et al., 2017). However, there is still a large amount of work to be done in this line.

Finally, it is not completely clear how to combine these word vectors into larger constituents in an unsupervised way, as simple addition or averaging often functions as well as more complicated methods (Kartsaklis, 2014). More recently, research has turned to learning contextualized word representations trained on large unlabeled data, which they can then fine-tune for other tasks (Felbo et al., 2017; Peters et al., 2018; Howard and Ruder, 2018). This approach, however, is not likely to help in cross-lingual cases, as the pretraining picks up on language-specific relations that are not likely to generalize.

3.3.2 Bilingual Embeddings

Bilingual word embeddings offer an intuitive solution for low-resource languages, as they can theoretically make the best use of large amounts of monolingual data and small amounts of parallel data. Although they have been used for cross-lingual document classification, there has been little previous work on using them for cross-lingual sentiment analysis. We can subdivide bilingual embeddings according to the amount of bilingual signal that they require. From most restrictive to least restrictive:

1. **Word-aligned Corpora**
2. **Sentence-aligned Corpora**
3. **Document-aligned Corpora**
4. **Projection-based Methods**

Word-aligned Corpora

Word-level bilingual supervision is the most restrictive data requirement that bilingual embedding algorithms can have, but ensures that the learned representations are of high quality.

Zou et al. (2013) introduce bilingual word embeddings for machine translation. They require large word aligned parallel corpora and use a modified max margin loss to predict true word / context pairs. They use bilingual regularization based on the word alignment counts in order to ensure that the word embeddings for both source and target languages are similar.

Luong et al. (2015) extend the Skip-gram model to bilingual embeddings. The proposed **BiSkip** algorithm requires word-aligned parallel corpora as input, and

uses a modified loss function which, given a center word, attempts to predict both the source and target words within a context window.

Gouws and Søgaard (2015) propose a method to create a pseudo-bilingual corpus with a small task-specific bilingual lexicon, which can then be used to train bilingual embeddings (BARISTA). This approach requires a monolingual corpus in both the source and target languages and a set of translation pairs. The source and target corpora are concatenated and then every word is randomly kept or replaced by its translation with a probability of 0.5. Any kind of word embedding algorithm can be trained with this pseudo-bilingual corpus to create bilingual word embeddings.

While research indicates that bilingual embeddings created with word-level supervision perform better than sentence- or document-level supervision on cross-lingual dictionary induction and cross-lingual document classification (Upadhyay et al., 2016), it is also prohibitive for under-resourced languages, as they often do not have large word-aligned parallel corpora available.

Sentence-aligned Corpora

Sentence-aligned data is less restrictive and easier to find for some language pairs. The EuroParl (Koehn, 2005) and Open Subtitles (Lison and Tiedemann, 2016) are available in major European languages.

Bilingual embedding methods that require only sentence-aligned data attempt to learn word representations that maximize the similarity of words that commonly occur in parallel sentences, while at the same time maximizing the similarity of words that occur in similar contexts monolingually (Hermann and Blunsom, 2014; Chandar et al., 2014).

Gouws et al. (2015), on the other hand, attempt to make the best use of both monolingual and bilingual signals. They use large monolingual corpora to best create the monolingual word representations, while requiring less parallel data which acts as bilingual regularization, ensuring that words that frequently occur in parallel sentences have similar representations.

Comparable Corpora

An even less restrictive approach is to use comparable data. Wikipedia articles have been the primary source of non-aligned but comparable corpora for creating bilingual word representations. (Vulić and Moens, 2014, 2015; Vulic and Moens, 2016)

use comparable Wikipedia articles to create bilingual word embeddings. They show that these representations work well for bilingual lexicon extraction.

Søgaard et al. (2015) create multilingual word representations by using the indexes of the Wikipedia articles in which they appear. They find these representations often outperform bilingual representations using sentence-aligned data.

Mogadala and Rettinger (2016) propose both a sentence- and document-aligned regularization term while learning bilingual word embeddings for cross-lingual document classification. They then use these representations to perform cross-lingual document classification.

Nonetheless, research indicates that representations created from document-aligned data perform worse than word-level or sentence-level supervised embeddings (Upadhyay et al., 2016; Mogadala and Rettinger, 2016) for downstream tasks. In fact, when using document-alignment only, Mogadala and Rettinger (2016) report results that are roughly 20 percentage points worse than using the sentence-aligned regularization for cross-lingual document classification.

Projection-based Methods

An approach to create bilingual embeddings that has a less prohibitive data requirement is to create monolingual vector spaces on large amounts of monolingual data and then learn a projection from one to the other with a small bilingual lexicon. This approach is interesting for under-resourced languages as it is often possible to find enough monolingual data to create high-quality word representations. Small bilingual lexicons, such as bilingual dictionaries or hand-translated lexicons, are often enough signal to enable a mapping to a bilingual space.

Mikolov et al. (2013b) find that vector spaces in different languages have similar arrangements. Therefore, they propose a linear projection which consists of learning a rotation and scaling matrix. They do this with a simple ridge regression approach, which is simply least squares regression with L2 regularization

$$\min_{W} \sum_{i \in D} ||Wx_i - z_i||^2 + \lambda ||W||_2^2 \tag{3.18}$$

where W is the weight matrix, $D = (x_1, z_1), (x_2, z_2), \ldots, (x_i, z_i)$ is a bilingual dictionary that contains translation pairs, with x in the source language and z in the target language. Here λ is a parameter that determines the amount of regularization used. They then show that these representations can be used to create translation dictionaries automatically.

VECMAP: Artetxe et al. (2016, 2017) propose an approach (VECMAP) that improves on the work of Mikolov et al. (2013b) by requiring the projection to be orthogonal, thereby preserving the monolingual quality of the original word vectors.

Given source embeddings S, target embeddings T, and a bilingual lexicon L, Artetxe et al. (2016) learn a projection matrix W by minimizing the square of Euclidean distances

$$\arg\min_{W} \sum_{i} ||S'W - T'||_F^2 \,, \tag{3.19}$$

where $S' \in S$ and $T' \in T$ are the word embedding matrices for the tokens in the bilingual lexicon L. This is solved using the Moore-Penrose pseudoinverse $S'^{+} = (S'^{T}S')^{-1}S'^{T}$ as $W = S'^{+}T'$, which can be computed using SVD. We refer to this approach as VECMAP.

Multilingual Unsupervised and Supervised Embeddings (MUSE): Lample et al. (2018a) propose a similar refined orthogonal projection method to that of Artetxe et al. (2017), but include an adversarial discriminator, which seeks to discriminate samples from the projected space WS, and the target T, while the projection matrix W attempts to prevent this making the projection from the source space WS as similar to the target space T as possible.

They further refine their projection matrix by reducing the hubness problem (Dinu et al., 2015), which is commonly found in high-dimensional spaces. For each projected embedding Wx, they define the k nearest neighbors in the target space, $\mathcal{N}_T$, suggesting $k = 10$. They consider the mean cosine similarity $r_T(Wx)$ between a projected embedding Wx and its k nearest neighbors

$$r_T(Wx) = \frac{1}{k} \sum_{y \in \mathcal{N}_T(Wx)} \cos(Wx, y) \tag{3.20}$$

as well as the mean cosine of a target word y to its neighborhood, which they denote by r_S.

In order to decrease similarity between mapped vectors lying in dense areas, they introduce a cross-domain similarity local scaling term (CSLS)

$$\text{CSLS}(Wx, y) = 2\cos(Wx, y) - r_T(Wx) - r_S(y) \,, \tag{3.21}$$

which they find improves accuracy, while not requiring any parameter tuning.

These last techniques have the advantage of requiring relatively little parallel training data while taking advantage of larger amounts of monolingual data. However, they are not optimized for sentiment.

Bilingual Sentiment Embeddings

Given that task-specific embeddings are generally desirable, research has moved towards creating sentiment-specific bilingual embeddings.

Zhou et al. (2015) propose a two step approach to create bilingual sentiment embeddings by translating all source data to the target language and vice versa. They first create bilingual representations using denoising autoencoders, in line with the approach of Chandar et al. (2014). Instead of using the entire vocabulary, however, they only attempt to model 2000 words taken from a sentiment lexicon, as well as their translations to the target language. Once the autoencoder learns a bilingual representation, they incorporate sentiment information by updating the parameters of the autoencoder while performing supervised classification. This optimizes the hidden representations to predict the sentiment of a document regardless of whether it is in the source or target language.

The major drawback of this approach is that it requires the existence of a machine translation system, which is a prohibitive assumption for many under-resourced languages, especially if it must be open and freely accessible. It also is only able to model a small number of sentiment bearing features (2000). While this is not a large problem for document-level classification, at sentence- or aspect-level, it is unlikely that these features would be present in all examples.

Chapter 4

EXPERIMENTS

In this chapter, we outline the main experiments of the thesis. We begin by evaluating which machine learning models are state-of-the-art in monolingual English sentiment analysis in Section 4.1. This provides us with a basis for choosing cross-lingual models later.

We then explore whether cross-lingual distributional representations are competitive with machine translation approaches in Section 4.2. These two sets of experiments motivate the approaches we propose in the rest of the thesis.

In Section 4.3 we propose a joint model to learn to project sentiment to a bilingual space for sentence-level cross-lingual sentiment analysis. We then propose several techniques to transfer this to aspect-level in Section 4.4.

We also include a more qualitative deployment experiment in Section 4.5, where we observe how well our approach works when applied to twitter sentiment analysis in ten European languages.

Finally, we transfer our sentence-level approach from a cross-lingual to a cross-domain task in Section 4.6.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2017). "Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment datasets." In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 2-12. http://aclweb.org/anthology/W/W17/W17-5202.pdf

4.1 Assessing State-of-the-art Monolingual Sentiment Models

Given that the main goal of this thesis is to create machine-learning models that are able to perform cross-lingual sentiment analysis, it is important to understand the properties of state-of-the-art machine learning models. In order to evaluate these models mono-lingually, we often use a number of datasets. However, not every model is tested on every dataset, and it is not clear that a model that performs well on one certain dataset will transfer well to other datasets with different properties.

The work described in this section aims at discovering if there are certain models that generally perform better or if there are certain models that are better adapted to certain kinds of datasets. Ultimately, the goal of these experiments is to *discover the best overall models in monolingual sentiment analysis*, so that we might see if these models perform equally well cross-lingually, or if they require modifications.

In the following sections we compare seven approaches to English sentiment analysis on six benchmark datasets[1]. We find that

- bidirectional LSTMS perform well across datasets,
- both LSTMS and bidirectional LSTMS are particularly good at fine-grained sentiment tasks,
- and embeddings trained jointly for semantics and sentiment perform well on datasets that are similar to the training data.

4.1.1 Datasets

We choose to evaluate the approaches presented in Section 4.1.2 on a number of different datasets from different domains, which also have differing levels of granularity of class labels. The Stanford Sentiment Treebank and SemEval 2013 shared-task dataset have already been used as benchmarks for some of the approaches mentioned in Section 3. Table 4.1 shows which approaches have been tested on which datasets and Table 4.2 gives an overview of the statistics for each dataset.

[1]The code and embeddings for the best models are available at `http://www.ims.uni-stuttgart.de/data/sota_sentiment`

	BOW	AVE	RETROFIT	JOINT	LSTM	BILSTM	CNN
SST-fine	–	–	–	–	+	+	+
SST-binary	–	+	+	–	+	+	+
OpeNER	+	–	–	–	–	–	–
SenTube-A	+	–	–	–	–	–	–
SenTube-T	+	–	–	–	–	–	–
SemEval	–	–	–	+	–	–	–

Table 4.1: Mapping of previous state-of-the-art methods to previous evaluations on state-of-the-art datasets. An + indicates that we are aware of a publication which reports on this combination and a – indicates our assumption that no reported results are available.

	Train	Dev.	Test	# Labels	Avg. Length	Vocab. Size
SST-fine	8,544	1,101	2,210	5	19.53	19,500
SST-binary	6,920	872	1,821	2	19.67	17,539
OpeNER	2,780	186	743	4	4.28	2,447
SenTube-A	3,381	225	903	2	28.54	18,569
SenTube-T	4,997	333	1,334	2	28.73	20,276
SemEval	6,021	890	2,376	3	22.40	21,163

Table 4.2: Statistics of datasets. Train, Dev., and Test refer to the number of examples for each subsection of a dataset. The number of labels corresponds to the annotation scheme, where: two is positive and negative; three is positive, neutral, negative; four is strong positive, positive, negative, strong negative; five is strong positive, positive, neutral, negative, strong negative.

Stanford Sentiment

The Stanford Sentiment Treebank (*SST-fine*) (Socher et al., 2013) is a dataset of movie reviews which was annotated for 5 levels of sentiment: strong negative, negative, neutral, positive, and strong positive. It is annotated both at the clause level, where each node in a binary tree is given a sentiment score, as well as at sentence level. We use the standard split of 8544/1102/2210 for training, validation and testing. In order to compare with Faruqui et al. (2015) , we also adapt the dataset to the task of binary sentiment analysis, where strong negative and negative are mapped to one label, and strong positive and positive are mapped to another label, and the neutral examples are dropped. This leads to a slightly different split of 6920/872/1821 (we refer to this dataset as *SST-binary*).

OpeNER

The *OpeNER* English dataset (Agerri et al., 2013) is a dataset of hotel reviews in which each review is annotated for opinions. An opinion includes sentiment holders, targets, and phrases, of which only the sentiment phrase is obligatory. Additionally, sentiment phrases are annotated for four levels of sentiment: strong negative, negative, positive and strong positive. We use a split of 2780/186/734 examples.

Sentube datasets

The SenTube datasets (Uryupina et al., 2014) are texts that are taken from YouTube comments regarding automobiles and tablets. These comments are normally directed towards a commercial or a video that contains information about the product. We take only those comments that have some polarity towards the target product in the video. For the automobile dataset (*SenTube-A*), this gives a 3381/225/903 training, validation, and test split. For the tablets dataset (*SenTube-T*) the splits are 4997/333/1334. These are annotated for positive, negative, and neutral sentiment.

Semeval 2013

The SemEval 2013 Twitter dataset (*SemEval*) (Nakov et al., 2013) is a dataset that contains tweets collected for the 2013 SemEval shared task B. Each tweet was annotated for three levels of sentiment: positive, negative, or neutral. There were

originally 9684/1654/3813 tweets annotated, but when we downloaded the dataset, we were only able to download 6021/890/2376 due to many of the tweets no longer being available.

4.1.2 Methodology

We compare seven approaches, five of which are already mentioned in Chapter 3, as well as two baselines. In this section, we describe the models and parameters and test them on the benchmark datasets mentioned in Section 4.1.1.

Models

Baselines We compare our models against two baselines. First, we train an L2-regularized logistic regression on a bag-of-words representation (BOW) of the training examples, where each example is represented as a vector of size n, with $n = |V|$ and V the vocabulary. This is a standard baseline for text classification.

Our second baseline is an L2-regularized logistic regression classifier trained on the average of the word vectors in the training example (AVE). We train word embeddings using the skip-gram with negative sampling algorithm (Mikolov et al., 2013a) on a 2016 Wikipedia dump, using 50-, 100-, 200-, and 600-dimensional vectors, a window size of 10, 5 negative samples, and we set the subsampling parameter to 10^{-4}. Additionally, we use the publicly available 300-dimensional GoogleNews vectors[2] in order to compare to previous work.

Retrofitting We apply the approach by Faruqui et al. (2015) and make use of the code[3] released in combination with the PPDB-XL lexicon, as this gave the best results for sentiment analysis in their experiments. We train for 10 iterations. Following the authors' setup, for testing we train an L2-regularized logistic regression classifier on the average word embeddings for a phrase (RETROFIT).

Joint Training For the joint method, we use the 50-dimensional sentiment embeddings provided by Tang et al. (2014). Additionally, we create 100-, 200-, and 300-dimensional embeddings using the code that is publicly available[4]. We use

[2] `https://code.google.com/archive/p/word2vec/`
[3] `https://github.com/mfaruqui/retrofitting`
[4] `http://ir.hit.edu.cn/~dytang`

the same hyperparameters as Tang et al. (2014): five million positive and negative tweets crawled using hashtags as proxies for sentiment, a 20-dimensional hidden layer, and a window size of three. Following the authors' setup, we concatenate the maximum, minimum and average vectors of the word embeddings for each phrase. We then train a linear SVM on these representations (JOINT).

Supervised Training We implement a standard LSTM which has an embedding layer that maps the input to a 50-, 100-, 200-, 300-, or 600-dimensional vector, depending on the embeddings used to initialize the layer. These vectors then pass to an LSTM layer. We feed the final hidden state to a standard fully-connected 50-dimensional dense layer and then to a softmax layer, which gives us a probability distribution over our classes. As a regularizer, we use a dropout (Srivastava et al., 2014) of 0.5 before the LSTM layer.

The BIDIRECTIONAL LSTM (BILSTM) has the same architecture as the normal LSTM, but includes an additional layer which runs from the end of the text to the front. We use the same parameters as the LSTM, but concatenate the two hidden layers before passing them to the dense layer[5].

We also train a simple one-layer CNN with one convolutional layer on top of pre-trained word embeddings. The first layer is an embeddings layer that maps the input of length *n* (padded when needed) to an $n \times R$ dimensional matrix, where R is the dimensionality of the word embeddings. The embedding matrix is then convoluted with filter sizes of 2, 3, and 4, followed by a pooling layer of length 2. This is then fed to a fully connected dense layer with ReLU activations (Nair and Hinton, 2010b) and finally to the softmax layer. We again use dropout (0.5), this time before and after the convolutional layers.

For all neural models, we initialize our word representations with the skip-gram algorithm with negative sampling (Mikolov et al., 2013a). For the 300-dimensional vectors, we use the publicly available GoogleNews vectors. For the other dimensions (50, 100, 200, 600), we create skip-gram vectors with a window size of 10, 5 negative samples and run 5 iterations. For out-of-vocabulary words, we use vectors initialized randomly between -0.25 and 0.25 to approximate the variance of the pre-trained vectors. We train our models using ADAM (Kingma and Ba, 2014) and a minibatch size of 32 and tune the hidden layer dimension and number of training epochs on the validation set.

[5]For the neural models on the *SST-fine* and *SST-binary* datasets, we do not achieve results as high as (Tai et al., 2015) and (Kim, 2014) , because we train our models only on sentence representations, not on the labeled phrase representations. We do this to be able to compare across datasets.

4.1.3 Results

Table 4.3 shows the results for the seven models across all datasets, as well as the macro-averaged results. We performed random approximation tests (Yeh, 2000) using the *sigf* package (Padó, 2006) with 10,000 iterations to determine the statistical significance of differences between models. Since the reported accuracies for the neural models are the means over five runs, we cannot use this technique in a straightforward manner. Therefore, we perform the random approximation tests between the runs[6] and consider the models statistically different if a majority (at least 3) of the runs are statistically different ($p < 0.01$, which corresponds to $p < 0.05$ with Bonferroni correction for 5 hypotheses). For interested readers, the results of statistical testing are summarized in Table 6.1.

Obviously, BOW continues to be a strong baseline: Though it never provides the best result on a dataset, it gives better results than AVE on *OpeNER*, *SenTube-T*, and *SemEval*. Surprisingly, it also performs better than JOINT on the same sets except for *SenTube-T*. Similarly, it outperforms RETROFIT on *SenTube-T* and *SemEval*.

RETROFIT performs better than CNN on *SST-fine* and JOINT on *SST-fine*, *SST-binary*, and *OpeNER*. It also improves the results of AVE across all datasets but *SenTube-A* and *SemEval* datasets.

Although JOINT does not perform well across datasets and, in fact, does not surpass the baselines on some datasets, it does lead to good results on *SemEval* and to state-of-the-art results on *SenTube-A* and *SenTube-T*.

Similarly to RETROFIT, CNN does not outperform any of the other methods on any dataset. As said, this method does not beat the baseline on *SST-fine*, *SenTube-A*, and *SenTube-T*. However, it outperforms the AVE baseline on *SST-binary* and *OpeNER*.

The best models are LSTM and BILSTM. The best overall model is BILSTM, which outperforms the other models on half of the tasks (*SST-fine*, *Opener*, and *SemEval*) and consistently beats the baseline. This is in line with other research (Plank et al., 2016; Kiperwasser and Goldberg, 2016; Zhou et al., 2016), which suggests that this model is robust across tasks as well as datasets. The differences in performance between LSTM and BILSTM, however, are only significant ($p < 0.01$) on the *SemEval* dataset, as shown in Table 6.1.

[6]We compare the results from the first run of model A with the first run of model B, then the second from A with the second from B, an so forth. An alternative would have been to use a t-test, which is common in such setting. However, we opted against this as the independence assumptions for such test do not hold.

	Model	Dim.	SST-fine	SST-binary	OpeNER	SenTube-A	SenTube-T	SemEval	Macro-Avg.
Baselines	BOW		40.3	80.7	77.1[4]	60.6[5]	66.0[5]	65.5	65.0
	AVE	50	38.9	74.1	59.5	62.0	61.7	58.1	59.0
		100	39.7	76.7	67.2	61.5	61.8	58.8	60.9
		200	40.7	78.2	69.3	60.6	62.8	61.1	62.1
		300	41.6	80.3[3]	76.3	61.5	64.3	63.6	64.6
		600	40.6	79.1	77.0	56.4	62.9	61.8	63.0
State-of-the-Art Methods	RETROFIT	50	39.2	75.3	63.9	60.6	62.3	58.1	59.9
		100	39.7	76.7	70.0	61.4	62.8	59.5	61.7
		200	41.8	78.3	73.5	60.0	63.2	61.2	63.0
		300	42.2	81.2[3]	75.9	61.7	63.6	61.8	64.4
		600	42.9	81.1	78.3	60.0	65.5	62.4	65.0
	JOINT	50	35.8	70.6	72.9	65.1	**68.1**	66.8[6]	63.2
		100	34.3	70.8	67.0	64.3	66.4	60.1	60.5
		200	33.7	72.3	68.6	**66.2**	66.6	58.4	61.0
		300	36.0	71.6	70.1	64.7	67.6	60.8	61.8
		600	36.9	74.0	75.8	63.7	64.2	60.9	62.6
	LSTM	50	43.3 (1.0)	80.5 (0.4)	81.1 (0.4)	58.9 (0.8)	63.4 (3.1)	63.9 (1.7)	65.2 (1.2)
		100	44.1 (0.8)	79.5 (0.6)	82.4 (0.5)	58.9 (1.1)	63.1 (0.4)	67.3 (1.1)	65.9 (0.7)
		200	44.1 (1.6)	80.9 (0.6)	82.0 (0.6)	58.6 (0.6)	65.2 (1.6)	66.8 (1.3)	66.3 (1.1)
		300	45.3[1] (1.9)	81.7[1] (0.7)	82.3 (0.6)	57.4 (1.3)	63.6 (0.7)	67.6 (0.6)	66.3 (1.0)
		600	44.5 (1.4)	**83.1** (0.9)	81.2 (0.8)	57.4 (1.1)	65.7 (1.2)	67.5 (0.7)	66.5 (1)
	BILSTM	50	43.6 (1.2)	82.9 (0.7)	79.2 (0.8)	59.5 (1.1)	65.6 (1.2)	64.3 (1.2)	65.9 (1.0)
		100	43.8 (1.1)	79.8 (1.0)	82.4 (0.6)	58.6 (0.8)	66.4 (1.4)	65.2 (0.6)	66.0 (0.9)
		200	44.0 (0.9)	80.1 (0.6)	81.7 (0.5)	58.9 (0.3)	63.3 (1.0)	66.4 (0.3)	65.7 (0.6)
		300	**45.6**[1] (1.6)	82.6[1] (0.7)	**82.5** (0.6)	59.3 (1.0)	66.2 (1.5)	65.1 (0.9)	**66.9** (1.1)
		600	43.2 (1.1)	83 (0.4)	81.5 (0.5)	59.2 (1.6)	66.4 (1.1)	**68.5** (0.7)	**66.9** (0.9)
	CNN	50	39.9 (0.7)	81.7 (0.3)	80.0 (0.9)	55.2 (0.7)	57.4 (3.1)	65.7 (1.0)	63.3 (1.1)
		100	40.1 (1.0)	81.6 (0.5)	79.5 (0.9)	56.0 (2.2)	61.5 (1.1)	64.2 (0.8)	63.8 (1.1)
		200	39.1 (1.1)	80.7 (0.4)	79.8 (0.7)	56.3 (1.8)	64.1 (1.1)	65.3 (0.8)	64.2 (1.0)
		300	39.8[2] (0.7)	81.3[2] (1.1)	80.3 (0.9)	57.3 (0.5)	62.1 (1.0)	63.5 (1.3)	64.0 (0.9)
		600	40.7 (2.6)	82.7 (1.2)	79.2 (1.4)	56.6 (0.6)	61.3 (2)	65.9 (1.8)	64.4 (1.5)

Table 4.3: Accuracy on the test sets. For all neural models we perform 5 runs and show the mean and standard deviation. The best results for each dataset is given in bold and results that have been previously reported are highlighted. All results derive from our reimplementation of the methods. We describe significance values in the text and appendix. Footnotes refer to the work where a method was previously tested on a specific dataset, although not necessarily with the same results: [1] (Tai et al., 2015) [2] (Kim, 2014) [3] (Faruqui et al., 2015) [4] (Lambert, 2015) [5] (Uryupina et al., 2014) [6] (Tang et al., 2014) .

We also see that the difference in performance between the two LSTM models and the others is larger on datasets with fine-grained labels (BILSTM 45.6 and LSTM 45.3 vs. an average of 40 for all others on the *SST-fine* and BILSTM 82.5 and LSTM 82.3 vs. an average of 76.5 on *OpeNER*). These differences between the LSTM models and other models are statistically significant, except for the difference between BILSTM and CNN at 50 dimensions on the *OpeNER* dataset.

Our analysis of different dimensionalities as input for the classification models reveals that, typically, the higher dimensional vectors (300 or 600) outperform lower dimensions. The only differences are in JOINT for *SenTube-T* and *SemEval* and LSTM for *SenTube-A* and AVE on all datasets except *OpeNER*.

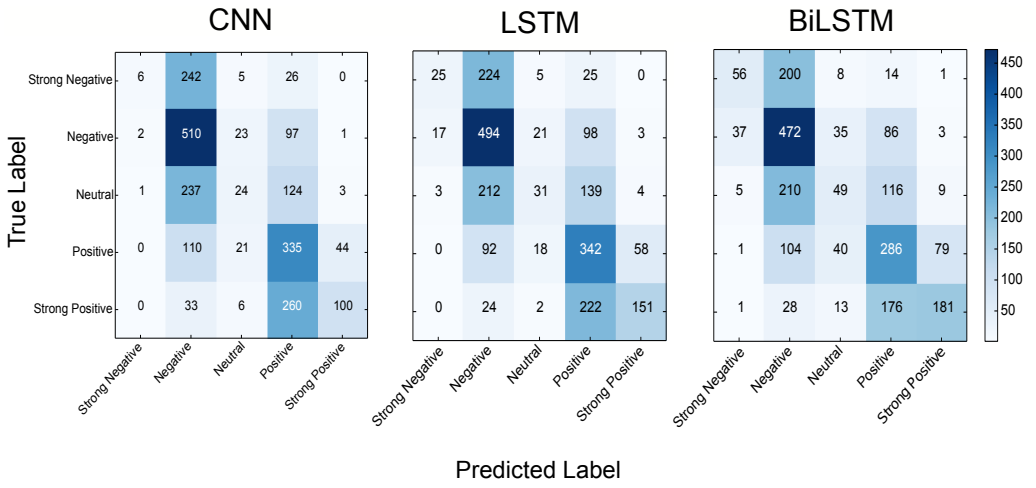


Figure 4.1: Confusion matrices of CNN, LSTM, and BILSTM on *SST-fine* dataset. We can see that both LSTM and BILSTM perform much better than CNN on strong negative, neutral, and strong positive classes.

While approaches that average the word embeddings for a sentence are comparable to state-of-the-art results (Iyyer et al., 2015), AVE and RETROFIT do not perform particularly well. This is likely due to the fact that logistic regression lacks the non-linearities which (Iyyer et al., 2015) found helped, especially at deeper layers. Averaging all of the embeddings for longer phrases also seems to lead to representations that do not contain enough information for the classifier.

We also experimented with using large sentiment lexicons as the semantic lexicon for retrofitting, but found that this hurt the representation more than it helped. We believe this is because there are not enough kinds of relationships to exploit the graph structure and by trying to collapse all words towards either a positive or negative center, too much information is lost.

χ^2 with SemEval	χ^2	p-value
SST-fine	19.408	0.002
SST-binary	19.408	0.002
OpeNER	19.408	0.002
SenTube-A	9.305	0.097
SenTube-T	7.377	0.194

Table 4.4: χ^2 statistics comparing the frequency of the following emoticons over the different datasets, :), :(, :-), :-(, :D, =). The difference in frequency of emoticons between the SemEval and SenTube datasets is not significant ($p > 0.05$), while for SST and OpeNER it is ($p < 0.05$).

We expected that JOINT would perform well on *SemEval*, given that it was designed for this task, but it was surprising that it performed so well on the *SenTube* datasets. It might be due to the fact that comments for these three datasets are comparably informal and make use of emoticons and Internet jargon. We performed a short analysis of datasets (shown in Table 4.4), where we take frequency of emoticons usage as an indirect indicator of informal speech and found that, indeed, the frequency of emoticons in the *SemEval* and *SenTube* datasets diverges significantly from the other datasets. The fact that JOINT is distantly trained on similar data gives it an advantage over other models on these datasets. This leads us to believe that this approach would transfer well to novel sentiment analysis tasks with similar properties.

The fact that CNN performs much better on *OpeNER* may be due to the smaller size of the phrases (an average of 4.28 vs. 20+ for other datasets), however, further analyses to prove this are needed.

The good results that both LSTM models achieved on the more fine-grained sentiment datasets (*SST-fine* and *OpeNER*) seem to indicate that LSTMS are able to learn dependencies that help to differentiate strong and weak versions of sentiment better than other models. This is supported by the confusion matrices shown in Figure 4.1. This makes them natural candidates for fine-grained sentiment analysis tasks.

LSTM performs better than BILSTM on two datasets but these differences are not statistically significant.

4.1.4 Discussion

In this section, we have shown that BiLSTMs are good general models for sentiment analysis in English, that both LSTMs and BiLSTMs work particularly well for fine-grained sentiment analysis, and that sentiment embeddings perform well on datasets similar to the data they are trained on.

Although sequence models, *i. e.* LSTMs and BiLSTMs, show the strongest results on monolingual sentiment analysis, this unfortunately does not make them ideal candidates for cross-lingual sentiment analysis.

LSTMs are sensitive to dependencies between words and to word order (Socher et al., 2013; Linzen et al., 2016), which is beneficial for monolingual sentiment analysis, as the classifier is able to do more than just identify keywords, incorporating such things as high-level negation or adverb modification. However, if we use a classifier trained on English data on machine-translated Spanish data, the translated data will exhibit characteristics different from those of the source language (Mohammad et al., 2016).

If we follow the goal of this thesis and decide not to use machine translation, in an attempt to design approaches which help under-resourced languages, the results will be even worse. The difference in word order means that currently, sequence models are not the best choice for our aim.

Given that embedding averaging methods do not perform much worse, we will use these as a substitution for sequence models.

Jeremy Barnes, Patrik Lambert, and Toni Badia (2016). "Exploring Distributional Representations and Machine Translation for Aspect-based Cross-lingual Sentiment Classification." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1613-1623. http://aclweb.org/anthology/C/C16/C16-1152.pdf

4.2 Exploring Distributional Representations and Machine Translation for Cross-lingual Sentiment Analysis

The previous section introduced state-of-the-art machine learning models and revealed that sequence models which use distributional representations as features are a good starting place for monolingual sentiment analysis. In fact, this approach to sentiment analysis leads to state-of-the-art results for most tasks. For the task of cross-lingual sentiment analysis, however, it is not as clear that distributional representations can carry both the bilingual and sentiment signals necessary to classify sentiment in a target language.

In this section we attempt to determine if distributional representations, whether that be *word embeddings* or the *latent hidden state* of autoencoders, can provide enough signal to enable cross-lingual sentiment analysis. We do this by performing experiments on a single language pair for which we have sufficient parallel data, as well as high-quality machine translation (English - Spanish), and testing on a single dataset available in the same domain in both languages (OpeNER). In this way we are able to fairly compare several different approaches, which have different data requirements, without introducing any domain differences.

Most research in cross-lingual sentiment analysis has used Statistical Machine Translation (SMT) as a way of bridging the gap between languages, but there are drawbacks to this. First, an SMT system must be available for the language combination at hand. This requires a great deal of development and the quality of the sentiment analysis system used afterwards depends heavily on the quality of the SMT system. Secondly, study shows that even high quality SMT introduces noise into the data (Mohammad et al., 2016). Finally, there are tasks in which systems which use distributed semantic representations to map between languages outperform SMT systems, e.g. cross-lingual document classification (Klementiev et al., 2012; Chandar et al., 2014).

For this reason, a different representation of words and phrases, *e. g.* distributional vector representations, could prove to be a more effective approach and enable us to leverage information from resource-rich languages (English) to perform CLSA in a target language that lacks these resources (e.g. Spanish, Catalan, Basque).

This section makes the following contributions:

- According to our knowledge, this is the most complete comparison of several types of distributed representations and machine translation for cross-lingual sentiment analysis.

OpeNER Corpora	English	Spanish
Training Examples	2780	2991
Strong Pos	23.38%	29%
Pos	46.08%	50.34%
Neg	25.61%	17.41%
Strong Neg	4.93%	3.01%
Test examples	929	999
Strong Pos	23.36%	29.23%
Pos	46.07%	50.34%
Neg	25.62%	17.42%
Strong Neg	4.95%	3.00%

Table 4.5: Statistics of OpeNER Corpora

- We demonstrate that distributed representations can be competitive with machine translation for Cross-lingual Sentiment Classification tasks.

The content of this section derives directly from the paper accepted at COLING 2016, mentioned in Section 1.4 (Barnes et al., 2016).

4.2.1 Methodology

Datasets

The data used to train the sentiment analysis models are the English and Spanish OpeNER sentiment corpora (Agerri et al., 2013). We take a subset of these corpora which deal only with hotel reviews. Each review has annotations for opinion holders, opinion targets and opinion sentiment. We refer to this triplet (opinion holder, opinion target, opinion sentiment) as an opinion unit. The sentiment can be strong positive, positive, negative, or strong negative. A neutral category is not included. As such, when training a classifier, rather than training on the complete sentence, we use the opinion unit. Table 4.5 shows the statistics for these corpora.

The corpora used to create the word embeddings are an English and Spanish Wikipedia corpus. These were taken from Wikipedia dumps in January 2016 and preprocessed to remove html markup and lowercase all words. We then performed sentence and word tokenization. We did not remove punctuation because this is often useful information for sentiment analysis. Table 4.6 gives the statistics for

Wikipedia Corpora	English	Spanish
Number of sentences	118,900,197	26,777,415
Number of tokens	2,055,786,401	506,612,108

Table 4.6: Statistics of Wikipedia Corpora

Europarl v7 Corpus	English	Spanish
Number of sentences	1,965,734	1,965,734
Number of tokens	49,093,806	51,575,784

Table 4.7: Statistics of Europarl v7 Corpus

these corpora.

The English-Spanish part of the Europarl v7 corpus[7] (Koehn, 2005) is used as parallel data. It contains around 2 million aligned sentences from the European Parliament. Table 4.7 shows the statistics for this corpus.

Representation of Training and Test Data for Sentiment Classification: For all experiments we use the same train and test split shown in Table 4.5. For each experiment, we trained a classifier on the English training data, performed the cross-lingual transfer on the Spanish test data and used this new data to test the classifier. A depiction of this setup is shown in Figure 4.2.

One difficulty encountered when using vector representations is that the opinion units are variable length. This means that to train a classifier either we find a fixed-length representation for all opinion units or we use a classifier that accepts variable-length input. We decided to take an averaging approach, which has shown promise in other works (Iyyer et al., 2015). For each opinion unit we took the arithmetic mean of the words that compose the opinion unit in order to create a fixed-length vector representation for each sentence (shown in Figure 4.3). We then use these vectors to train a classifier. For the SMT transfer methods, we trained the classifier on unigram features. In all experiments, we used the sequential minimal optimization (SMO) classifier from the WEKA toolkit (Hall et al., 2009).

Vector Space Projection: Following Mikolov et al. (2013b) we create two sets of monolingual word embeddings using the Europarl v7 corpus (Koehn, 2005). We

[7] http://www.statmt.org/europarl

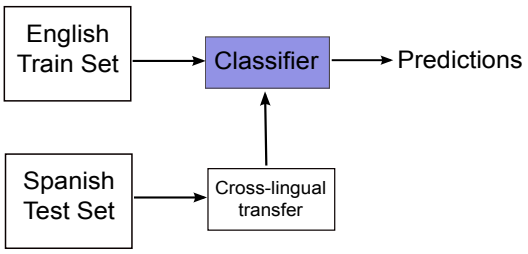


Figure 4.2: The process of cross-lingual sentiment classification. We assume that the opinion units have already been determined. The English train set is used to train a classifier. The Spanish test set is mapped accordingly and the classifier is tested on this cross-lingual test set.

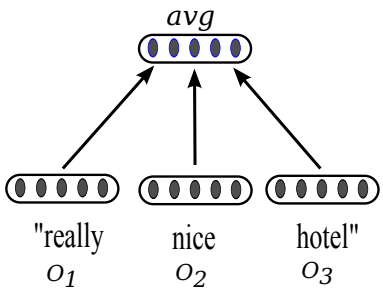


Figure 4.3: The representation of an opinion unit. For each word in the opinion unit, we take its vector representation and average these vectors in order to create a fixed-length vector which we use to train a sentiment classifier

use the Skip-gram model (Mikolov et al., 2013a) and create 300 dimensional vectors using a window of 5 words, and 10 negative samples. We compile a bilingual dictionary by taking the 8000 most common words in the English Wikipedia and translating them using Bing Translator[8]. Although Bing gives several options, we take only the first translation for use in our bilingual dictionary. We then remove errors and ambiguous words manually and arrive at a final number of 4518 word pairs to train the matrix. Finally, we use ridge regression to optimize the translation matrix W. After creating the transition matrix W, we test the effectiveness of this matrix translation to enable Cross-lingual Sentiment Classification.

For each opinion unit in the corpora, we create a fixed-length vector representation, as shown in Figure 4.3. We now have a dataset with training instances such as $\{x_i, y_i\}$, where x_i is a 300 dimension vector and y_i is its corresponding label (Strong Positive, Positive, Negative, Strong Negative). As a baseline, we train and

[8]http://www.microsofttranslator.com

test an SVM on the Spanish data from the OpeNER corpus as the Spanish test set has the same opinion units as the cross-lingual test set. We do the same with the English data, although this is not truly comparable.

We then create the cross-lingual test set by applying our translation matrix W to the embeddings for each sentence Spanish test set. In order to find the most similar vector from the English word embeddings, we use a k-nearest neighbor algorithm with cosine as the distance metric. Finally, we use the mean of the word embeddings as mentioned above to create the final fixed-length representation. We test on the cross-lingual test set. The results are shown in Table 4.8.

Bilingual Word Embeddings: The next set of experiments required the use of parallel sentences to create bilingual word embeddings. Following the work of Luong et al. (2015), we create bilingual word embeddings using the Bilingual Skip-gram (BISKIP) algorithm, which uses the Skip-gram model (Mikolov et al., 2013a) with an added bilingual objective. This algorithm creates vector representations in which words that appear in parallel sentences have similar representations. We use the BISKIP algorithm to train English and Spanish word vectors on the Wikipedia corpora and the Europarl corpus (Koehn, 2005). We create 300 dimensional vectors with a window of 5 words on either side, 10 negative samples and run the algorithm for 3 epochs. This process gives us two sets of word embeddings in which words that often appear in parallel sentences have similar vector representations.

To train our classifier, we use our learned English embeddings and take the average of the vectors in each opinion unit in the English train set. We perform the same procedure with the learned Spanish embeddings and the Spanish test set. The results are shown in Table 4.8.

Stacked Bilingual Denoising Autoencoders: Following the work of Zhou et al. (2016) we train a Stacked Bilingual Denoising Autoencoder (SBDA) on parallel sentences from the Europarl corpus. This approach aims to encode the parallel sentences into a common latent space. Given a vocabulary of length *n*, SBDA maps the sentences, which are represented as n-dimensional one-hot vectors, to a lower dimensional representation. These representations are then used to reconstruct the original sentences. In order to keep SBDA from simply learning the identity function, the lower dimensional representation of one of the sentences is corrupted, which causes SBDA to look for discriminative features to help reconstruct the original sentences. In this way, SBDA learns to find a lower dimensional representation that encodes as much information as possible needed to reconstruct the bilingual sentences.

We create source and target language autoencoders with 1000 hidden units, which are then mapped to 500 hidden units. We set the corruption level to 0.5 and concatenate and normalize the 500 source and 500 target hidden units before feeding them to the bilingual autoencoder. After SBDA has been trained, we use the learned weights to represent our data in the latent bilingual space.

We then create training data by mapping the opinion units from the English train set to a latent 500 dimensional vector in bilingual space. We train a classifier on the mapped English training set. We test on the similarly mapped Spanish test set. The results are shown in Table 4.8.

Statistical Machine Translation For the final experiment we use statistical machine translation as a means of bridging the gap between languages. We translate our target language data using Google Translate[9], a highly developed SMT system, as well as Constrained SMT. Specifically, with Google Translate we translate only the opinion phrases. This has the disadvantage that translation is done without context, but the advantage that the opinion phrases are not mixed with the rest of the text. We compared this approach with a Constrained SMT approach (Lambert, 2015). Constrained SMT allows us to translate the opinion units in context, but without reordering or scrambling them. The language model used in this approach is trained with data from the hotel domain, which improves the quality of translation and results in more accurate Cross-lingual Sentiment Classification results.

Finally, we train an SVM classifier on unigram features from the monolingual English training set. We create test sets by translating the Spanish test data with each SMT system. The results are shown in Table 4.8.

Equal amounts of parallel data: Each of the previous experiments rely on different amounts of parallel data for optimal performance. Since we are interested in their performance on under-resourced languages, we run all experiments again with the minimal amount of parallel data (measured at 15.9M English words).

4.2.2 Results

From the results it is clear that the projection-search approach did not yield any effective results. This may be a result of several factors. Mikolov et al. (2013b) were able to leverage a simple mapping strategy between word embeddings created

[9]http://translate.google.com/

	EN	ES	VSP	CSMT	BWE	SDBA	Google
Parallel Data[10]	-	-	4518	15.9M	15.9M	15.9M	-
Precision	-	-	45.3	77.9	49.0	25.4	-
Recall	-	-	31.0	75.8	46.8	40.0	-
F1 Score	-	-	35.1	75.5	47.3	33.8	-
Accuracy	-	-	48.0	75.8	62.0	55.0	-
Parallel Data	0	0	8000	15.9M	49M	49M	?
Precision	82.4	80.3	25.4	77.9	43.7	68.2	67.0
Recall	82.2	80.9	25.1	75.8	46.8	59.0	56.8
F1 Score	82.0	80.0	25.2	75.5	45.3	63.3	61.5
Accuracy	82.2	80.9	33.5	75.8	56.0	74.5	72.8

Table 4.8: Results of Crosslingual Experiments: Precision, recall and F_1 are the weighted averages of all classes. EN (English) and ES (Spanish) show the results of training and testing a classifier on the monolingual data. Vector Space Projection (VSP), Constrained SMT (CSMT), Bilingual Word Embeddings (BWE), Stacked Bilingual Denoising Autoencoders (SDA), and Google SMT (Google).

from large monolingual datasets in order to fill the gaps in translation dictionaries. Given the poor results of this experiment, it seems unlikely that using nearest neighbor search is useful for cross-lingual sentiment analysis.

In Mikolov et al. (2013b), the success of this technique depended largely on using a small subset of the vocabulary and pairing it with other approaches. In our approach, however, all of the weight of correctly classifying a phrase fell on the accuracy of the mapping scheme. Therefore, it seems that any error in the mapping resulted in the propagation of error during classification. Another problem that arose is that there were some words whose vector representation always appeared as the nearest neighbor of many other words, although they were not semantically similar with any of them. This problem is known as hubness and is an intrinsic problem with high-dimensional vector space. Our work seems to confirm the research of Lazaridou et al. (2015) and Dinu et al. (2015), who showed that hubness is compounded when trying to create a linear mapping between two sets of word embeddings.

The constrained SMT approach is the most accurate approach and shows that, given a more refined treatment of less parallel data, one can achieve CLSA systems which are comparable to monolingual ones. It is interesting that Google Translate has a better BLEU score than constrained SMT[11], but the performance on the

[11]Google Translate achieves 48.6 BLEU in English-Spanish, versus 45.3 for constrained SMT.

classification task was lower. It also showed poorer results than the bilingual stacked denoising autoencoder.

The results given by the bilingual word embeddings are not optimal, but are promising enough to warrant more research. There are problems with bilingual word embeddings which would need to be addressed in order to improve their usefulness for Cross-lingual Sentiment Classification. First, there is the problem of ambiguity that affects all word embeddings. This is not often taken into consideration in the literature. One way to correct this problem would be to disambiguate the word senses prior to creating the word embeddings. Cheng et al. (2014) show that this technique improves the performance of distributional models for learning compositional models of meaning and it may improve the performance for sentiment analysis as well.

Secondly, due to the fact that they have similar distributions, antonyms are often given similar vector representations. This is not a problem for POS-taggers or parsers, but it is detrimental to sentiment analysis systems based on word embeddings because these words have opposing polarities and should therefore have different vector representations. To remedy this, one could add a classification task in the problem formulation that would better separate these antonyms into differing vector spaces (Tang et al., 2014).

The SBDA approach gives reasonably good results, despite the fact that it was designed for sentence-level CLSA. There are still ways which we could adapt this approach to aspect-based CLSA. By using word alignment, we could split sentences into parallel or pseudo-parallel n-grams and train the autoencoder with this data. This may improve its performance at aspect-level.

The results shown in Table 4.8 show that, despite a general decrease in precision, recall and F1, the performance of bilingual word embeddings remains stable with less data. SBDA, however, performs poorly with this amount of data

4.2.3 Discussion

In this section, we have compared three bilingual distributional approaches (vector space projection, bilingual word embeddings, and bilingual stacked denoising autoencoders) to two machine translation approaches (google SMT and constrained SMT). We have shown that while Stacked Bilingual Denoising Autoencoders show promise, most current distributional approaches do not perform as well as MT for cross-lingual sentiment analysis.

Role of Parallel Data: It is interesting to see that there is not a direct correlation

between the amount of parallel data used and the results. Constrained SMT uses less data than BWE or SBDA and still outperforms both. However, this approach uses higher quality, in-domain data as well as tuning parameters which adapt it to this domain. The trend within representation and distributional approaches has been to use larger and larger datasets, but these results seem to suggest that using smaller, task-specific in-domain datasets which are automatically discovered from larger datasets may be key in improving performance in cross-lingual sentiment analysis.

Representation: Besides using the average of the vectors in the opinion unit as a representation, we also experimented with summation. Summation led to results that were slightly worse than averaging. This is likely due to the fact that longer opinion units result in vectors which are a magnitude larger than shorter opinion units.

Classifiers: Apart from the SVM classifiers used in all experiments, we conducted further experiments using deep feed-forward networks and LSTMS. The feed-forward network gave similar results to the SVM, but the LSTMS performed poorly. LSTMS are powerful non-linear classifiers which are able to take word order into account. In our setup, it seems they overfit to the source side feature space and word order and are not able to generalize to the target language, despite the fact that the feature space is similar.

Parallel Data: One of the interesting findings from these experiments is that more parallel data and a closer bilingual alignment does not necessarily lead to better results in cross-lingual sentiment analysis. This corresponds to more recent progress in projection-based bilingual word embeddings which require little or no parallel data, and achieve better results than highly supervised approaches (Artetxe et al., 2016, 2017; Lample et al., 2018a). In the following experiments, we will exploit the fact that little parallel data can create useful bilingual representations to develop better cross-lingual sentiment methods.

Finally, it is important to note that none of these methods were created specifically for sentiment analysis. For the distributional approaches, it is possible and often desirable to incorporate sentiment information directly in the training objective (Tang et al., 2014; Yu et al., 2017). In the next section, we will address this problem.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2018). "Bilingual Sentiment Embeddings: Joint Projection of Sentiment Across Languages." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2483-2493. http://aclweb.org/anthology/P18-1231.

4.3 Bilingual Sentiment Embeddings

The previous section shows us that distributional representations can be used for cross-lingual sentiment analysis, but currently lack some necessary properties. Specifically, stacked denoising autoencoders, biskip embeddings, and the basic matrix projection technique do not make the best use of small amounts of parallel data. Secondly, they do not incorporate sentiment information, as they are trained only to maximize the similarity of translation pairs in vector space. This means that while they may provide useful representations for word similarity tasks (Faruqui and Dyer, 2014; Vulic and Moens, 2016; Artetxe et al., 2016, 2018), they do not perform well on cross-lingual sentiment analysis.

Cross-lingual approaches using distributed representations have shown great promise for a number of tasks, such as cross-lingual document classification (Prettenhofer and Stein, 2011; Chandar et al., 2014), part of speech tagging (Buys and Botha, 2016; Plank et al., 2016) and cross-lingual dictionary induction (Mikolov et al., 2013b; Hermann and Blunsom, 2014; Artetxe et al., 2016). For cross-lingual sentiment analysis, however, these approaches do not currently perform well. This is due in part to the fact that they have no sentiment signal during training.

In this section, we propose a novel approach to incorporate sentiment information in a cross-lingual sentiment classification model. Here we perform sentence-level classification, in order to concentrate on the projection and sentiment objectives, without having to worry about other problems that arise when working at aspect-level, such as multiple aspects in a single sentence or sparse signals, where a single word determines the polarity. We will later extend this approach to aspect-level in Section 4.4.

Our model, *Bilingual Sentiment Embeddings* (BLSE), are embeddings that are jointly optimized to represent both (1) semantic information in the source and target languages, which are bound to each other through a small bilingual dictionary, and (2) sentiment information, which is annotated on the source language only. We only need three resources: (i) a comparably small bilingual lexicon, (ii) an annotated sentiment corpus in the resource-rich language, and (iii) monolingual word embeddings for the two involved languages.

We show that our model outperforms previous state-of-the-art models in nearly all experimental settings across six benchmarks. In addition, we offer an in-depth analysis and demonstrate that our model is aware of sentiment. Finally, we provide a qualitative analysis of the joint bilingual sentiment space. Our implementation is publicly available at `https://github.com/jbarnesspain/blse`.

The content of this section derives directly from the paper accepted at ACL 2018,

mentioned in Section 1.4 (Barnes et al., 2018a).

4.3.1 Model

In order to project not only semantic similarity and relatedness but also sentiment information to our target language, we propose a new model, namely *Bilingual Sentiment Embeddings* (BLSE), which jointly learns to predict sentiment and to minimize the distance between translation pairs in vector space. In this section we detail the projection objective, the sentiment objective, and finally the full objective. A sketch of the model is depicted in Figure 4.4.

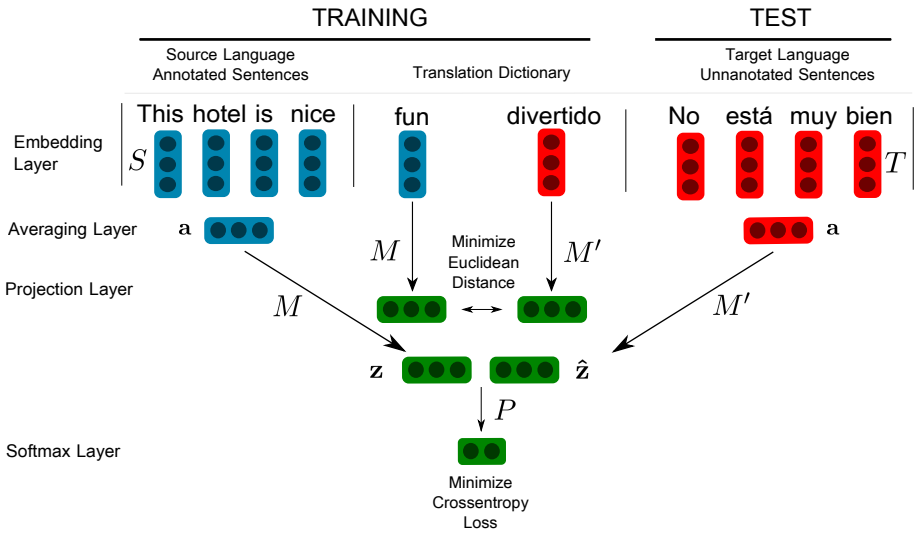


Figure 4.4: Bilingual Sentiment Embedding Model (BLSE)

Cross-lingual Projection

We assume that we have two precomputed vector spaces $S = \mathbb{R}^{v \times d}$ and $T = \mathbb{R}^{v' \times d'}$ for our source and target languages, where v (v') is the length of the source vocabulary (target vocabulary) and d (d') is the dimensionality of the embeddings. We also assume that we have a bilingual lexicon L of length n which consists of word-to-word translation pairs $L = \{(s_1, t_1), (s_2, t_2), \ldots, (s_n, t_n)\}$ which map from source to target.

In order to create a mapping from both original vector spaces S and T to shared sentiment-informed bilingual spaces $\mathbf{z}$ and $\hat{\mathbf{z}}$, we employ two linear projection

matrices, M and M'. During training, for each translation pair in L, we first look up their associated vectors, project them through their associated projection matrix and finally minimize the mean squared error of the two projected vectors. This is similar to the approach taken by Mikolov et al. (2013b) , but includes an additional target projection matrix.

The intuition for including this second matrix is that a single projection matrix does not support the transfer of sentiment information from the source language to the target language. Without M', any signal coming from the sentiment classifier would have no effect on the target embedding space T, and optimizing M to predict sentiment and projection would only be detrimental to classification of the target language. We analyze this further in Section 4.3.5. Note that in this configuration, we do not need to update the original vector spaces, which would be problematic with such small training data.

The projection quality is ensured by minimizing the mean squared error[12][13]

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{z_i} - \hat{\mathbf{z}}_\mathbf{i})^2\,, \tag{4.1}$$

where $\mathbf{z_i} = S_{s_i} \cdot M$ is the dot product of the embedding for source word s_i and the source projection matrix and $\hat{\mathbf{z}}_\mathbf{i} = T_{t_i} \cdot M'$ is the same for the target word t_i.

Sentiment Classification

We add a second training objective to optimize the projected source vectors to predict the sentiment of source phrases. This inevitably changes the projection characteristics of the matrix M, and consequently M' and encourages M' to learn to predict sentiment without any training examples in the target language.

To train M to predict sentiment, we require a source-language corpus $C_{\text{source}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$ where each sentence x_i is associated with a label y_i.

For classification, we use a two-layer feed-forward averaging network, loosely following Iyyer et al. (2015) [14]. For a sentence x_i we take the word embeddings

[12] We omit parameters in equations for better readability.

[13] We also experimented with cosine distance, but found that it performed worse than Euclidean distance.

[14] Our model employs a linear transformation after the averaging layer instead of including a non-linearity function. We choose this architecture because the weights M and M' are also used to learn a linear cross-lingual projection.

from the source embedding S and average them to $\mathbf{a}_i \in \mathbb{R}^d$. We then project this vector to the joint bilingual space $\mathbf{z}_i = \mathbf{a}_i \cdot M$. Finally, we pass $\mathbf{z}_i$ through a softmax layer P to get our prediction $\hat{y}_i = \text{softmax}(\mathbf{z}_i \cdot P)$.

To train our model to predict sentiment, we minimize the cross-entropy error of our predictions

$$H = -\sum_{i=1}^{n} y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)\,. \tag{4.2}$$

Joint Learning

In order to jointly train both the projection component and the sentiment component, we combine the two loss functions to optimize the parameter matrices M, M', and P by

$$J = \sum_{(x,y) \in C_{\text{source}}} \sum_{(s,t) \in L} \alpha H(x, y) + (1 - \alpha) \cdot \text{MSE}(s, t)\,, \tag{4.3}$$

where α is a hyperparameter that weights sentiment loss vs. projection loss.

Target-language Classification

For inference, we classify sentences from a target-language corpus C_{target}. As in the training procedure, for each sentence, we take the word embeddings from the target embeddings T and average them to $\mathbf{a}_i \in \mathbb{R}^d$. We then project this vector to the joint bilingual space $\hat{\mathbf{z}}_i = \mathbf{a}_i \cdot M'$. Finally, we pass $\hat{\mathbf{z}}_i$ through a softmax layer P to get our prediction $\hat{y}_i = \text{softmax}(\hat{\mathbf{z}}_i \cdot P)$.

4.3.2 Datasets and Resources

OpeNER and MultiBooked

To evaluate our proposed model, we conduct experiments using four benchmark datasets and three bilingual combinations. We use the OpeNER English and Spanish datasets (Agerri et al., 2013) and the MultiBooked Catalan and Basque datasets (Barnes et al., 2018c). All datasets contain hotel reviews which are annotated for aspect-level sentiment analysis. The labels include *Strong Negative* ($--$), *Negative* ($-$), *Positive* ($+$), and *Strong Positive* ($++$). We map the aspect-level annotations to sentence level by taking the most common label and remove instances of mixed polarity. We also create a binary setup by combining the strong

		EN	ES	CA	EU
Binary	+	1258	1216	718	956
	−	473	256	467	173
	Total	1731	1472	1185	1129
4-class	++	379	370	256	384
	+	879	846	462	572
	−	399	218	409	153
	−−	74	38	58	20
	Total	1731	1472	1185	1129

Table 4.9: Statistics for the OpeNER English (EN) and Spanish (ES) as well as the MultiBooked Catalan (CA) and Basque (EU) datasets.

and weak classes. This gives us a total of six experiments. The details of the sentence-level datasets are summarized in Table 4.9. For each of the experiments, we take 70 percent of the data for training, 20 percent for testing and the remaining 10 percent are used as development data for tuning.

4.3.3 Monolingual Word Embeddings

For BLSE, VECMAP, and MT, we require monolingual vector spaces for each of our languages. For English, we use the publicly available GoogleNews vectors. For Spanish, Catalan, and Basque, we train skip-gram embeddings using the Word2Vec toolkit[15] with 300 dimensions, subsampling of 10^{-4}, window of 5, negative sampling of 15 based on a 2016 Wikipedia corpus[16] (sentence-split, tokenized with IXA pipes (Agerri et al., 2014) and lowercased). The statistics of the Wikipedia corpora are given in Table 2.8.

Bilingual Lexicon

For BLSE, VECMAP, and BARISTA, we also require a bilingual lexicon. We use the sentiment lexicon from Hu and Liu (2004) (to which we refer in the following as Hu and Liu) and its translation into each target language. We translate the lexicon using Google Translate and exclude multi-word expressions.[17] This leaves

[15] `https://code.google.com/archive/p/word2vec`

[16] `http://attardi.github.io/wikiextractor/`

[17] Note that we only do that for convenience. Using a machine translation service to generate this list could easily be replaced by a manual translation, as the lexicon is comparably small.

a dictionary of 5700 translations in Spanish, 5271 in Catalan, and 4577 in Basque. We set aside ten percent of the translation pairs as a development set in order to check that the distances between translation pairs not seen during training are also minimized during training.

4.3.4 Experiments

Setting

We compare BLSE (Section 4.3.1) to VECMAP and BARISTA (Section 3.3.2) as baselines, which have similar data requirements and to machine translation (MT) and monolingual (MONO) upper bounds which request more resources. For all models (MONO, MT, VECMAP, BARISTA), we take the average of the word embeddings in the source-language training examples and train a linear SVM[18]. We report this instead of using the same feed-forward network as in BLSE as it is the stronger upper bound. We choose the regularization parameter c on the target language development set and evaluate on the target language test set.

Upper Bound MONO. We set an empirical upper bound by training and testing a linear SVM on the target language data. We train the model on the averaged embeddings from target language training data, tuning the c parameter on the development data. We test on the target language test data.

Upper Bound MT. To test the effectiveness of machine translation, we translate all of the sentiment corpora from the target language to English using the Google Translate API[19]. Note that this approach is not considered a baseline, as we assume not to have access to high-quality machine translation for low-resource languages of interest.

Baseline VECMAP. We compare with the approach proposed by Artetxe et al. (2016) which has shown promise on other tasks, such as word similarity. In order to learn the projection matrix W, we need translation pairs. We use the same word-to-word bilingual lexicon mentioned in Section 4.3.1. We then map the source vector space S to the bilingual space $\hat{S} = SW$ and use these embeddings.

Baseline BARISTA. We also compare with the approach proposed by Gouws and Søgaard (2015) . The bilingual lexicon used to create the pseudo-bilingual corpus is the same word-to-word bilingual lexicon mentioned in Section 4.3.1. We follow the authors' setup to create the pseudo-bilingual corpus. We create bilingual

[18]LinearSVC implementation from scikit-learn.

[19]`https://translate.google.com`

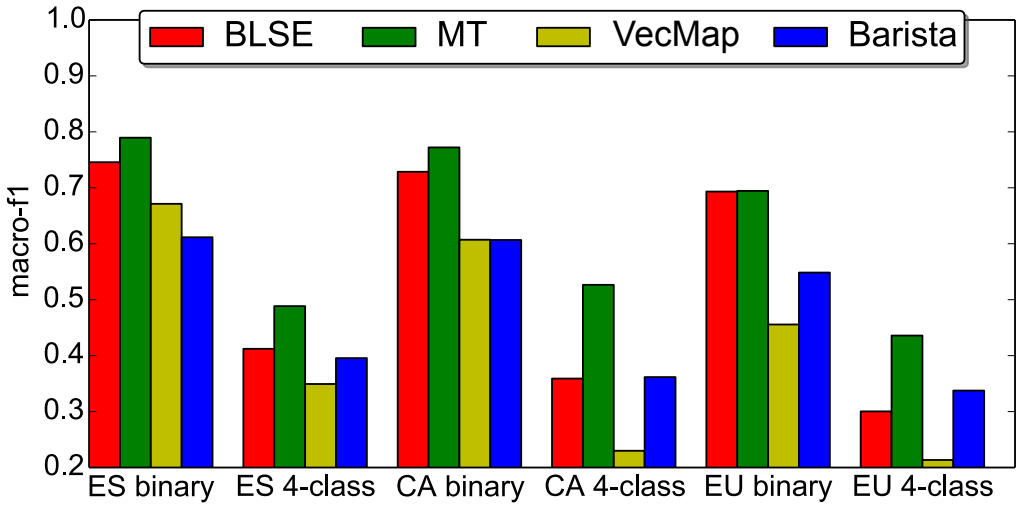


Figure 4.5: Binary and four class macro F_1 on Spanish (ES), Catalan (CA), and Basque (EU).

embeddings by training skip-gram embeddings using the Word2Vec toolkit on the pseudo-bilingual corpus using the same parameters from Section 4.3.3.

Our method: BLSE. We implement our model BLSE in Pytorch (Paszke et al., 2016) and initialize the word embeddings with the pretrained word embeddings S and T mentioned in Section 4.3.3. We use the word-to-word bilingual lexicon from Section 4.3.3, tune the hyperparameters α, training epochs, and batch size on the target development set and use the best hyperparameters achieved on the development set for testing. ADAM (Kingma and Ba, 2014) is used in order to minimize the average loss of the training batches.

Ensembles We create an ensemble of MT and each projection method (BLSE, VECMAP, BARISTA) by training a random forest classifier on the predictions from MT and each of these approaches. This allows us to evaluate to what extent each projection model adds complementary information to the machine translation approach.

Results

In Figure 4.5, we report the results of all four methods. Our method outperforms the other projection methods (the baselines VECMAP and BARISTA) on four of the six experiments substantially. It performs only slightly worse than the more resource-costly upper bounds (MT and MONO). This is especially noticeable for the binary classification task, where BLSE performs nearly as well as machine

			Binary			4-class		
			ES	CA	EU	ES	CA	EU
Upper Bounds	MONO	P	75.0	79.0	74.0	55.2	50.0	48.3
		R	72.3	79.6	67.4	42.8	50.9	46.5
		F_1	73.5	79.2	69.8	45.5	49.9	47.1
	MT	P	**82.3**	78.0	75.6	51.8	**58.9**	43.6
		R	76.6	76.8	66.5	48.5	50.5	45.2
		F_1	79.0	77.2	69.4	48.8	52.7	43.6
	BLSE	P	72.1	***72.8*	***67.5*	*****60.0***	*38.1*	**42.5*
		R	*****80.1***	***73.0*	*****72.7***	**43.4*	38.1	*37.4*
		F_1	***74.6*	***72.9*	***69.3*	**41.2*	35.9	30.0
Baselines	VecMap	P	75.0	60.1	42.2	40.1	21.6	30.0
		R	64.3	61.2	49.5	36.9	29.8	35.7
		F_1	67.1	60.7	45.6	34.9	23.0	21.3
	Barista	P	64.7	65.3	55.5	44.1	36.4	34.1
		R	59.8	61.2	54.5	37.9	38.5	34.3
		F_1	61.2	60.1	54.8	39.5	36.2	33.8
Ensemble	VecMap	P	65.3	63.1	70.4	43.5	46.5	50.1
		R	61.3	63.3	64.3	44.1	48.7	50.7
		F_1	62.6	63.2	66.4	43.8	47.6	49.9
	Barista	P	60.1	63.4	50.7	48.3	52.8	50.8
		R	55.5	62.3	50.4	46.6	53.7	49.8
		F_1	56.0	62.5	49.8	47.1	53.0	47.8
	BLSE	P	79.5	**84.7**	**80.9**	49.5	54.1	**50.3**
		R	**78.7**	**85.5**	69.9	**51.2**	**53.9**	**51.4**
		F_1	**80.3**	**85.0**	**73.5**	**50.3**	**53.9**	**50.5**

Table 4.10: Precision (P), Recall (R), and macro F_1 of four models trained on English and tested on Spanish (ES), Catalan (CA), and Basque (EU). The **bold** numbers show the best results for each metric per column and the *highlighted* numbers show where BLSE is better than the other projection methods, VECMAP and BARISTA (** $p < 0.01$, * $p < 0.05$).

translation and significantly better than the other methods. We perform approximate randomization tests (Yeh, 2000) with 10,000 runs and highlight the results that are statistically significant (**p < 0.01, *p < 0.05) in Table 4.10.

In more detail, we see that MT generally performs better than the projection methods (79–69 F_1 on binary, 52–44 on 4-class). BLSE (75–69 on binary, 41–30 on 4-class) has the best performance of the projection methods and is comparable with MT on the binary setup, with no significant difference on binary Basque. VECMAP (67–46 on binary, 35–21 on 4-class) and BARISTA (61–55 on binary, 40–34 on 4-class) are significantly worse than BLSE on all experiments except Catalan and Basque 4-class. On the binary experiment, VECMAP outperforms BARISTA on Spanish (67.1 vs. 61.2) and Catalan (60.7 vs. 60.1) but suffers more than the other methods on the four-class experiments, with a maximum F_1 of 34.9. BARISTA is relatively stable across languages.

ENSEMBLE performs the best, which shows that BLSE adds complementary information to MT. Finally, we note that all systems perform successively worse on Catalan and Basque. This is presumably due to the quality of the word embeddings, as well as the increased morphological complexity of Basque.

4.3.5 Model and Error Analysis

We analyze three aspects of our model in further detail: (i) where most mistakes originate, (ii) the effect of the bilingual lexicon, and (iii) the effect and necessity of the target-language projection matrix M'.

Phenomena

In order to analyze where each model struggles, we categorize the mistakes and annotate all of the test phrases with one of the following error classes: vocabulary (voc), adverbial modifiers (mod), negation (neg), external knowledge (know) or other. Table 4.11 shows the results.

Vocabulary: The most common way to express sentiment in hotel reviews is through the use of polar adjectives (as in "the room was great) or the mention of certain nouns that are desirable ("it had a pool"). Although this phenomenon has the largest total number of mistakes (an average of 71 per model on binary and 167 on 4-class), it is mainly due to its prevalence. MT performed the best on the test examples which according to the annotation require a correct understanding of the vocabulary (81 F_1 on binary /54 F_1 on 4-class), with BLSE (79/48) slightly

Model		voc	mod	neg	know	other	*total*
MT	bi	49	26	19	14	5	**113**
	4	147	94	19	21	12	**293**
VECMAP	bi	80	44	27	14	7	**172**
	4	182	141	19	24	19	**385**
BARISTA	bi	89	41	27	20	7	**184**
	4	191	109	24	31	15	**370**
BLSE	bi	67	45	21	15	8	**156**
	4	146	125	29	22	19	**341**

Table 4.11: Error analysis for different phenomena. See text for explanation of error classes.

worse. VECMAP (70/35) and BARISTA (67/41) perform significantly worse. This suggests that BLSE is better than VECMAP and BARISTA at transferring sentiment of the most important sentiment bearing words.

Negation: Negation is a well-studied phenomenon in sentiment analysis (Pang et al., 2002; Wiegand et al., 2010; Zhu et al., 2014; Reitan et al., 2015). Therefore, we are interested in how these four models perform on phrases that include the negation of a key element, for example "In general, this hotel isn't bad". We would like our models to recognize that the combination of two negative elements "isn't" and "bad" lead to a *Positive* label.

Given the simple classification strategy, all models perform relatively well on phrases with negation (all reach nearly 60 F_1 in the binary setting). However, while BLSE performs the best on negation in the binary setting (82.9 F_1), it has more problems with negation in the 4-class setting (36.9 F_1).

Adverbial Modifiers: Phrases that are modified by an adverb, *e. g.*, the food was *incredibly* good, are important for the four-class setup, as they often differentiate between the base and strong labels. In the binary case, all models reach more than 55 F_1. In the 4-class setup, BLSE only achieves 27.2 F_1 compared to 46.6 or 31.3 of MT and BARISTA, respectively. Therefore, presumably, our model does currently not capture the semantics of the target adverbs well. This is likely due to the fact that it assigns too much sentiment to functional words (see Figure 4.9).

External Knowledge Required: These errors are difficult for any of the models to get correct. Many of these include numbers which imply positive or negative sentiment (350 meters from the beach is *Positive* while 3 kilometers from the beach is *Negative*). BLSE performs the best (63.5 F_1) while MT performs comparably

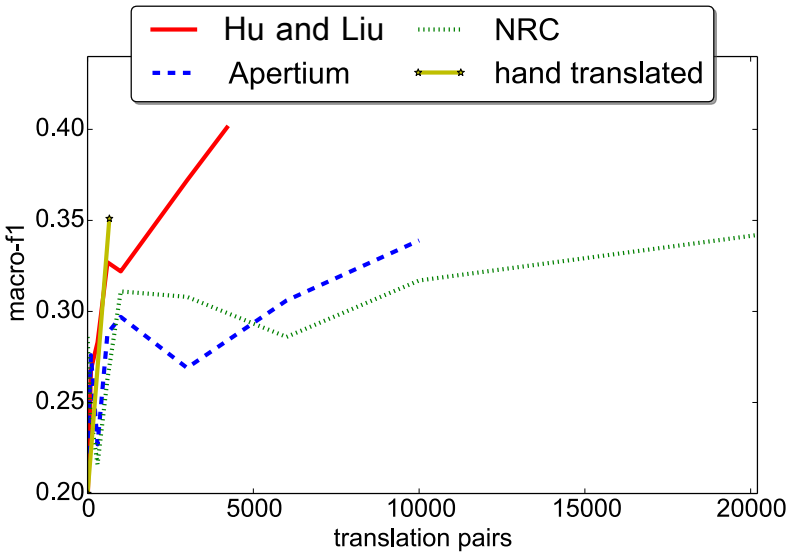


Figure 4.6: Macro F_1 for translation pairs in the Spanish 4-class setup. Training with the expanded hand translated lexicon and machine-translated Hu and Liu lexicon gives a macro F_1 that grows constantly with the number of translation pairs. Despite having several times more training data, the Apertium and NRC translation dictionaries do not perform as well.

well (62.5). BARISTA performs the worst (43.6).

Binary vs. 4-class: All of the models suffer when moving from the binary to 4-class setting; an average of 26.8 in macro F_1 for MT, 31.4 for VECMAP, 22.2 for BARISTA, and for 36.6 BLSE. The two vector projection methods (VECMAP and BLSE) suffer the most, suggesting that they are currently more apt for the binary setting.

Effect of Bilingual Lexicon

We analyze how the number of translation pairs affects our model. We train on the 4-class Spanish setup using the best hyper-parameters from the previous experiment.

Research into projection techniques for bilingual word embeddings (Mikolov et al., 2013b; Lazaridou et al., 2015; Artetxe et al., 2016) often uses a lexicon of the most frequent 8–10 thousand words in English and their translations as training data. We test this approach by taking the 10,000 word-to-word translations from the

Apertium English-to-Spanish dictionary[20]. We also use the Google Translate API to translate the NRC hashtag sentiment lexicon (Mohammad et al., 2013) and keep the 22,984 word-to-word translations. We perform the same experiment as above and vary the amount of training data from 0, 100, 300, 600, 1000, 3000, 6000, 10,000 up to 20,000 training pairs. Finally, we compile a small hand translated dictionary of 200 pairs, which we then expand using target language morphological information, finally giving us 657 translation pairs[21]. The macro F_1 score for the Hu and Liu dictionary climbs constantly with the increasing translation pairs. Both the Apertium and NRC dictionaries perform worse than the translated lexicon by Hu and Liu, while the expanded hand translated dictionary is competitive, as shown in Figure 4.6.

While for some tasks, *e. g.*, bilingual lexicon induction, using the most frequent words as translation pairs is an effective approach, for sentiment analysis, this does not seem to help. Using a translated sentiment lexicon, even if it is small, gives better results.

Analysis of M'

The main motivation for using two projection matrices M and M' is to allow the original embeddings to remain stable, while the projection matrices have the flexibility to align translations and separate these into distinct sentiment subspaces. To justify this design decision empirically, we perform an experiment to evaluate the actual need for the target language projection matrix M': We create a simplified version of our model without M', using M to project from the source to target and then P to classify sentiment.

The results of this model are shown in Figure 4.7. The modified model does learn to predict in the source language, but not in the target language. This confirms that M' is necessary to transfer sentiment in our model.

4.3.6 Qualitative Analyses of Joint Bilingual Sentiment Space

In order to understand how well our model transfers sentiment information to the target language, we perform two qualitative analyses. First, we collect two sets of 100 positive sentiment words and one set of 100 negative sentiment words. An effective cross-lingual sentiment classifier using embeddings should learn that two

[20] `http://www.meta-share.org`

[21] The translation took approximately one hour. We can extrapolate that hand translating a sentiment lexicon the size of the Hu and Liu lexicon would take no more than 5 hours.

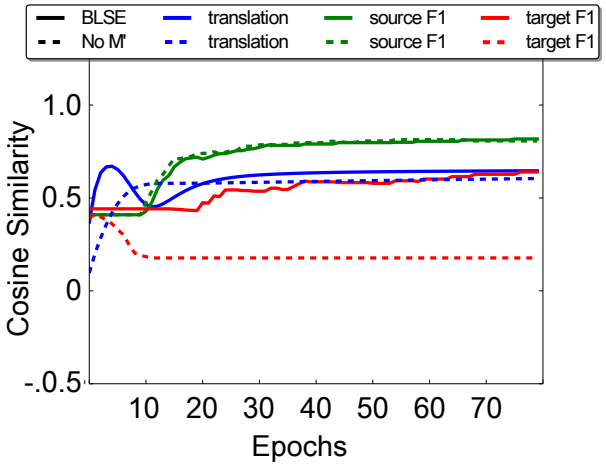


Figure 4.7: BLSE model (solid lines) compared to a variant without target language projection matrix M' (dashed lines). "Translation" lines show the average cosine similarity between translation pairs. The remaining lines show F_1 scores for the source and target language with both variants of BLSE. The modified model cannot learn to predict sentiment in the target language (red lines). This illustrates the need for the second projection matrix M'.

positive words should be closer in the shared bilingual space than a positive word and a negative word. We test if BLSE is able to do this by training our model and after every epoch observing the mean cosine similarity between the sentiment synonyms and sentiment antonyms after projecting to the joint space.

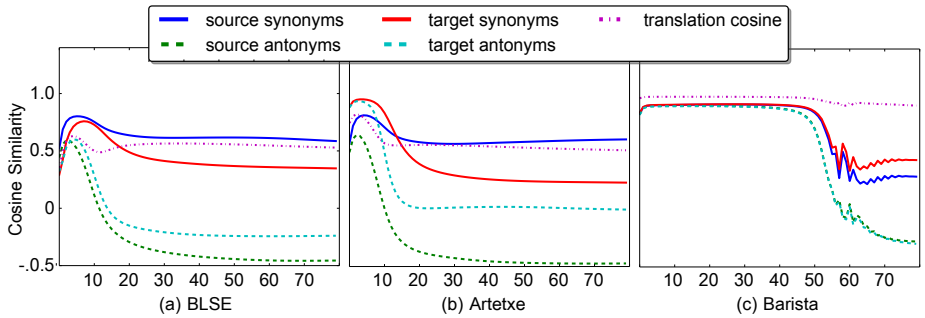


Figure 4.8: Average cosine similarity between a subsample of translation pairs of same polarity ("sentiment synonyms") and of opposing polarity ("sentiment antonyms") in both target and source languages in each model. The x-axis shows training epochs. We see that BLSE is able to learn that sentiment synonyms should be close to one another in vector space and sentiment antonyms should not.

We compare BLSE with VECMAP and BARISTA by replacing the Linear SVM

classifiers with the same multi-layer classifier used in BLSE and observing the distances in the hidden layer. Figure 4.8 shows this similarity in both source and target language, along with the mean cosine similarity between a held-out set of translation pairs and the macro F_1 scores on the development set for both source and target languages for BLSE, BARISTA, and VECMAP. From this plot, it is clear that BLSE is able to learn that sentiment synonyms should be close to one another in vector space and antonyms should have a negative cosine similarity. While the other models also learn this to some degree, jointly optimizing both sentiment and projection gives better results.

Secondly, we would like to know how well the projected vectors compare to the original space. Our hypothesis is that some relatedness and similarity information is lost during projection. Therefore, we visualize six categories of words in t-SNE, which projects high dimensional representations to lower dimensional spaces while preserving the relationships as best as possible (Van der Maaten and Hinton, 2008): positive sentiment words, negative sentiment words, functional words, verbs, animals, and transport.

The t-SNE plots in Figure 4.9 show that the positive and negative sentiment words are rather clearly separated after projection in BLSE. This indicates that we are able to incorporate sentiment information into our target language without any labeled data in the target language. However, the downside of this is that functional words and transportation words are highly correlated with positive sentiment.

4.3.7 Discussion

The experiments in this section have proven that it is possible to perform cross-lingual sentiment analysis without machine translation, and that jointly learning to project and predict sentiment is advantageous. This supports the growing trend of jointly training for multiple objectives (Tang et al., 2014; Klinger and Cimiano, 2015; Ferreira et al., 2016).

This approach has also been exploited within the framework of multi-task learning, where a model learns to perform multiple similar tasks in order to improve on a final task (Collobert et al., 2011). The main difference between the joint method proposed here and multi-task learning is that vector space projection and sentiment classification are not similar enough tasks to help each other. In fact, these two objectives compete against one another, as a perfect projection would not contain enough information for sentiment classification, and vice versa.

In the next section we will address how to move from sentence-level to aspect-level sentiment analysis within this framework.

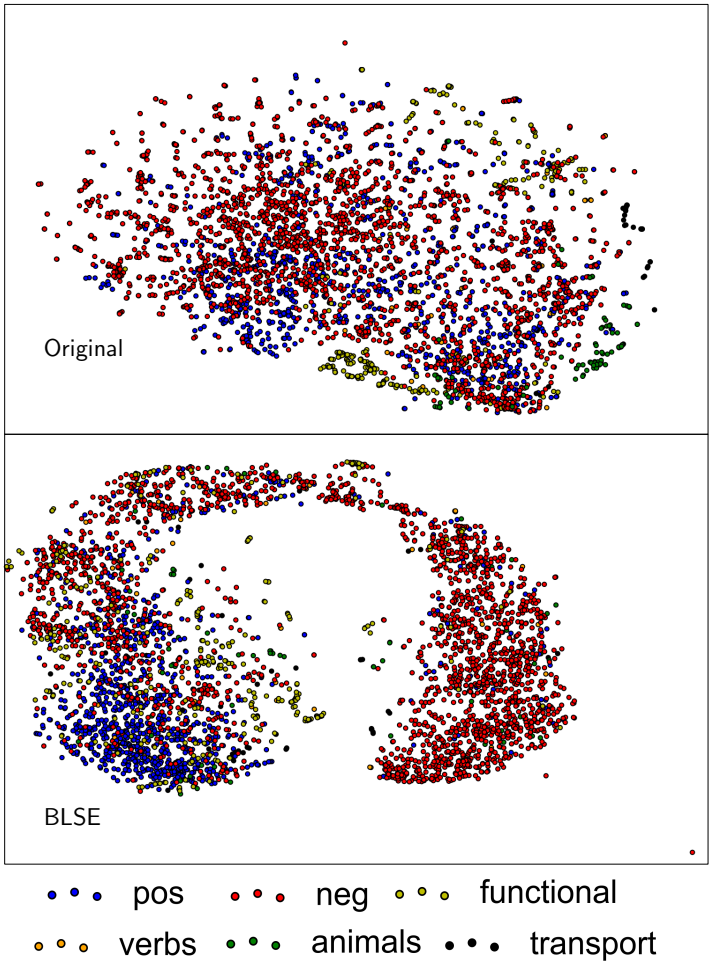


Figure 4.9: t-SNE-based visualization of the Spanish vector space before and after projection with BLSE. There is a clear separation of positive and negative words after projection, despite the fact that we have used no labeled data in Spanish.

4.4 Beyond Sentences: Projection-based Aspect-level Sentiment Analysis

In the previous Section, we proposed a method to learn bilingual sentiment embeddings and demonstrated their effectiveness for sentence-level cross-lingual sentiment analysis. In this section, we move beyond sentences and extend our method to aspect-level.

As mentioned in Chapter 1, aspect-level sentiment analysis (ABSA) aims at predicting the polarity expressed towards a particular entity or sub-aspect of that

entity. This is a more realistic view of sentiment, as polarities are directed towards targets, not spread uniformly across sentences or documents. Take the following example:

> The café near my house has great coffee but I never go there because the service is terrible.

We mark the aspect, an evaluative positive phrase and an evaluative negative phrase. In this sentence, it is not stated what the sentiment towards the aspect "café" is, while the sentiment of the aspect "coffee" is positive and that of "service" is negative. In order to correctly classify the sentiment of each aspect, it is necessary to (1) detect the aspects, (2) detect polarity expressions, and (3) resolve the relations between these. In this section we formulate the task of aspect-level sentiment analysis as classification, given the aspects from an oracle. In order to successfully determine the polarity of an aspect, approaches require a large number of annotated examples, which are only available in a few major languages.

The question we attempt to address in this section is *how to infer the polarities of aspects in a target language that does not have any annotated data to permit supervised learning while avoiding the need for machine translation or large parallel corpora?* In the following Spanish sentence, for example, how can we determine that the sentiment of "servicio" is negative, while that of "comida" is positive if we do not have annotated data in Spanish?

> El servicio en el restaurante fue pésimo. Por lo menos la comida estaba rica.

For languages without annotated sentiment resources, cross-lingual sentiment analysis (CLSA) approaches offer a way to transfer the information from a source language corpus to the target language. As mentioned several times before, MT has traditionally been the main approach for transferring information across language barriers. But this is particularly problematic for aspect-level sentiment analysis, as changes in word order or loss of words created during translation can directly affect the performance of a classifier.

In this section, we formulate aspect-level sentiment analysis following the **targeted** sentiment analysis setup from Section 3.1.4. This only differs from aspect-level sentiment analysis in that we assume we already know the entities and aspects to be classified. This is a simplification we make in order to identify the best projection and classification strategies, without having to deal with the automatic detection of aspects, either as part of a pipeline or jointly. Specifically, we compare approaches which require (1) minimal bilingual data and instead make use of (2) high-quality monolingual word embeddings in the source and target language. More specifically, the main contributions are

- extending previous cross-lingual approaches to enable aspect-level cross-lingual sentiment analysis with minimal parallel data requirements,
- comparing different model architectures for cross-lingual aspect-level sentiment analysis.
- performing a detailed error analysis, and detailing the advantages and disadvantages of each method.

The rest of the section is structured as follows: in Section 4.4.1, we outline the projection methods we will compare and in Section 4.4.2, we describe the approaches to move from sentence- to aspect-level CLSA. In Section 4.4.3, we detail the data and experiments. Finally, we perform a detailed error analysis in Section 4.4.5 and conclude in Section 4.4.6.

4.4.1 Methodology

In this section, we combine four different (three projection-based) cross-lingual methods with four approaches to move from sentence to aspect level. These methods are detailed in Chapter 3, but we repeat some of the information for ease of reading. We compare splitting the phrase into contexts, as proposed by Zhang et al. (2015); Tang et al. (2016), to a baseline and two simplified versions of the model (explained in Section 4.4.2) in order to understand the differences and individual model properties better.

Cross-lingual Projection Methods

Previous approaches to ABSA and cross-lingual sentiment analysis are not easily applicable to aspect-level cross-lingual sentiment analysis, as they require high-quality annotated resources to train, *i. e.*, machine translation, or large parallel corpora. Therefore, we compare three projection-based bilingual embedding methods (Bilingual Sentiment Embeddings (BLSE), Bilingual Word Embedding Mappings (VECMAP), and Multilingual Unsupervised and Supervised Embeddings (MUSE)), as well as BARISTA, which creates bilingual embeddings using a pseudo-bilingual corpus and has similar data requirements as the projection methods.

For all methods except BARISTA, we assume that we have two precomputed vector spaces $S = \mathbb{R}^{v \times d}$ and $T = \mathbb{R}^{v' \times d'}$ for our source and target languages, where v (v') is the length of the source vocabulary (target vocabulary) and d (d') is the dimensionality of the embeddings. We use the embeddings from Section 2.2.1.

4.4.2 Context Splitting for Cross-Lingual Aspect-level Sentiment Analysis

We assume that the list of target aspects as they occur in the text is given. These can be extracted previously either by using domain knowledge (Liu et al., 2005), by using a named entity recognizer (Zhang et al., 2015) or by using a number of aspect extraction techniques (Zhou et al., 2012). Given these aspects, the task is reduced to classification. For this classification task given the aspect, we compare all projection methods explained in the previous Section 4.4.1 respectively.

Our approach, SPLIT, is similar to the methods proposed by Tang et al. (2016) and Zhang et al. (2016) for LSTMS and gated recurrent networks respectively in a sentence classification setting. For a sentence with an aspect a, we split the sentence at a in order to get a left and right context, $\text{con}_\ell(a)$ and $\text{con}_r(a)$ respectively.

Initial experiments using LSTMS showed that, in cross-lingual setups, they overfit too much to word order and source-language specific information to perform well on our tasks. Therefore, instead of using an LSTM to create a representation of each context, we average each left context $\text{con}_\ell(a_i)$, right context $\text{con}_r(a_i)$, and target aspect a_i separately. Although averaging is a simplified approach to create a compositional representation of a phrase, it has been shown to work well for sentiment (Iyyer et al., 2015; Barnes et al., 2017). After creating a single averaged vector for the left context, right context, and target aspect, we concatenate them and use these as features for the sentiment classifier.

Given that information related to the aspect is normally local, we also experiment with giving more weight to the terms that are close to the aspect. We do this while creating the averaged embedding $\bar{\mathbf{e}}_i$. We give the words in the contexts a weight w depending on their distance from the aspect a.

The weighted embedding vector $\bar{\mathbf{e}}_i$ for a context or aspect $(\mathbf{e_1}, \ldots, \mathbf{e}_n)$ with $\mathbf{e}_i$ being the embedding vectors is then

$$\bar{\mathbf{e}}_i = \frac{1}{n} \sum_{i=1}^{n} w_i \cdot \mathbf{e}(i)_i \,, \tag{4.4}$$

where w_i is the distance weighting for token i.

Simplified Models: Context only and Aspect only

We hypothesized in the introduction of this section that cross-lingual approaches are particularly error-prone when evaluative phrases and words are wrongly predicted. In such settings, it might be beneficial for a model to put emphasis on the

		Binary		Multiclass				
		+	−	++	+	0	−	−−
OpeNER	EN	1658	661	472	1132		556	105
	ES	2404	446	813	1591		387	59
MultiBooked	CA	1453	883	645	808		741	142
	EU	1461	314	686	775		273	41
SemEval	EN	2268	953		2268	145	953	
	ES	2675	948		2675	168	948	
USAGE	EN	2985	1456		2985	34	1456	
	DE	3115	870		3115	99	870	

Table 4.12: Aspect-level statistics for the datasets. A blank area indicates that the dataset has no annotations for that label.

target word itself and learn a prior distribution of sentiment for each target independent of the context. To analyze this, we compare our model to two simplified versions.

The first is ASPECT-ONLY, which means that we use the model in the same way as before but ignore the context completely. This serves as a tool to understand how much model performance originates from the aspect itself.

In the same spirit, we use a CONTEXT-ONLY model, which ignores the target by constraining the parameters of all aspect phrase embeddings to be the same. This approach might be beneficial over our initial model if the prior distribution between aspects was similar and the context actually carries the relevant information.

Baseline: Sentence Assumption

As the baseline for each projection method, we assume all aspects in each sentence respectively to be of the same polarity (SENT). This is generally an erroneous assumption, but can give good results if all of the aspects in a sentence have the same polarity.

4.4.3 Experiments

Bilingual Lexicon

For all projection-based methods, as well as BARISTA, we require a bilingual lexicon to learn the projection. We take the sentiment lexicon from Hu and Liu (2004) , which we then translate to each target language using the Google API, which gave the best results in Section 4.3.

Datasets

We use the following corpora to set up the experiments in which we train on a source language corpus C_S and test on a target language corpus C_T. Statistics for all of the corpora are shown in Table 4.12. We include a binary classification setup, where neutral has been removed and strong positive and strong negative have been mapped to positive and negative, as well as a multiclass setup, where the original labels are used.

OpeNER Corpora: The OpeNER corpora (Agerri et al., 2013) are composed of hotel reviews, annotated for aspect-based sentiment. Each aspect is annotated with a sentiment label (Strong Positive, Positive, Negative, Strong Negative). We perform experiments with the English and Spanish versions.

MultiBooked Corpora: The MultiBooked corpora (Barnes et al., 2018a) are also hotel reviews annotated in the same way as the OpeNER corpora, but in Basque and Catalan. These corpora allow us to observe how well each approach performs on low-resource languages.

SemEval 2016 Task 5: We take the English and Spanish restaurant review corpora made available by the organizers of the SemEval event (Pontiki et al., 2016). These corpora are annotated for three levels of sentiment (positive, neutral, negative).

USAGE Corpora: The USAGE corpora (Klinger and Cimiano, 2014) are Amazon reviews taken from a number of different items, and are available in English and German. Each aspect is annotated for three levels of sentiment (positive, neutral, negative). As the corpus has two sets of annotations available, we take the annotations from annotator 1 as the gold standard.

Baselines and Empirical Upper Bounds

For further analysis, we show a simple majority baseline as well as an MT upper bound. We assume our models to perform between these two, as we do not have access to the millions of parallel sentences required to perform high-quality MT and particularly aim at proposing a method which is less resource-hungry.

Baseline: Majority class (Maj.) This is a naive baseline that always chooses the majority class. This simple baseline gives us a way of determining how well we are performing.

Upperbound: Machine Translation (MT) To test the effectiveness of machine translation, we translate all of the test sets of the target language sentiment corpora to English using the Google Translate API[22]. This approach is not considered a baseline, as we assume not to have access to high-quality machine translation for low-resource languages of interest.

4.4.4 Results

Table 4.13 shows the macro F_1 scores for all cross-lingual approaches (BLSE, VECMAP, MUSE, BARISTA, MT) and all aspect-level approaches (SENT, SPLIT, CONTEXT-ONLY, and ASPECT-ONLY). The final column is the average over all corpora. The final row in each setup shows the macro F_1 for a classifier that always chooses the majority class.

We experimented with the best weighting scheme for the weighted average in SPLIT, but found that a uniform weighting scheme performed best. The results from this weighting scheme are reported in all experiments.

BLSE outperforms other projection methods on the binary setup, 63.0 macro averaged F_1 across corpora versus 59.0, 57.9, and 51.4 for VECMAP, MUSE, and BARISTA, respectively. On the multiclass setup, however, MUSE (32.2 F_1) is the best, followed by VECMAP (31.0), BARISTA (28.1) and BLSE (23.7). VECMAP is never the best nor the worst approach. In general, BARISTA performs poorly on the binary setup, but slightly better on the multiclass, although the overall performance is still weak.

The SPLIT approach to ABSA improves over the SENT baseline on 33 of 50 experiments, especially on binary (21/25), while on multiclass it is less helpful

[22] `https://translate.google.com`

			EN-ES OpeNER	EN-CA MultiBooked	EN-EU MultiBooked	EN-ES SemEval	EN-DE USAGE	Average
Binary	SENT	BLSE	64.4	47.3	45.5	61.1	63.8	56.4
		VECMAP	52.2	41.8	39.1	42.3	31.2	51.3
		MUSE	47.6	40.1	45.8	45.3	47.5	45.3
		BARISTA	47.3	39.1	45.8	42.3	33.4	41.6
		MT	**70.8**	**81.5**	**76.2**	70.9	58.8	71.6
	SPLIT	BLSE	*66.8*	*69.8*	*66.3*	*62.2*	50.0	*63.0*
		VECMAP	65.8	64.4	65.1	60.0	39.9	59.0
		MUSE	58.3	64.3	50.2	59.8	*57.0*	57.9
		BARISTA	61.9	59.0	56.1	44.5	35.3	51.4
		MT	67.3	77.8	74.8	73.2	69.4	72.5
	CONTEXT-ONLY	BLSE	47.3	39.1	45.8	42.3	55.9	46.1
		VECMAP	47.3	39.1	45.8	42.3	45.8	44.1
		MUSE	55.5	67.5	52.1	61.6	45.4	56.4
		BARISTA	47.3	60.2	51.9	42.3	45.5	49.4
		MT	66.5	78.1	72.4	**74.2**	**73.1**	**72.9**
	ASPECT-ONLY	BLSE	53.1	43.7	42.7	42.3	41.5	44.7
		VECMAP	54.4	51.1	35.4	45.5	45.2	46.3
		MUSE	56.2	55.4	52.3	46.0	47.5	51.5
		BARISTA	48.9	53.0	48.5	42.3	44.8	47.5
		MT	46.7	40.1	45.8	47.5	56.0	47.2
	Maj.		47.3	39.1	45.8	42.3	43.0	43.5
Multiclass	SENT	BLSE	25.2	23.3	16.6	36.0	40.5	28.3
		VECMAP	28.1	19.9	26.3	28.2	28.3	26.2
		MUSE	22.4	23.2	23.5	27.4	24.1	24.1
		BARISTA	29.3	35.8	27.0	27.4	29.9	29.9
		MT	**41.4**	**46.5**	**44.3**	33.1	28.9	**38.8**
	SPLIT	BLSE	18.5	14.3	15.7	**40.6**	29.5	23.7
		VECMAP	29.2	30.9	28.0	38.9	27.9	31.0
		MUSE	*32.9*	*33.5*	*27.3*	27.4	**39.7**	*32.2*
		BARISTA	27.9	35.1	27.3	27.4	33.4	28.1
		MT	24.7	29.2	27.0	33.8	33.2	29.6
	CONTEXT-ONLY	BLSE	18.5	12.6	15.7	27.4	38.4	22.5
		VECMAP	18.5	12.6	15.7	27.4	28.3	20.5
		MUSE	22.7	39.0	27.4	27.4	30.0	29.3
		BARISTA	32.9	31.6	27.2	27.4	32.1	30.2
		MT	27.5	31.4	27.2	30.6	34.4	30.2
	ASPECT-ONLY	BLSE	19.1	17.3	16.7	27.4	25.3	21.2
		VECMAP	25.8	23.1	19.0	32.1	25.3	25.1
		MUSE	23.2	21.6	17.1	29.5	31.1	24.5
		BARISTA	21.8	21.5	16.8	27.4	33.9	24.3
		MT	26.9	23.3	23.9	30.5	33.6	27.6
	Maj.		18.5	12.6	15.7	27.4	28.3	20.5

Table 4.13: Macro F_1 results for all corpora and techniques. We denote the best performing projection-based method per column with a *blue box* and the best overall method per column with a **green box**.

	correct	incorrect
BLSE	2.1	2.5
VECMAP	2.5	2.1
MUSE	2.1	2.2
BARISTA	1.7	2.2
MT	2.1	2.2

Table 4.14: Average length of tokens of correctly and incorrectly classified aspects on the OpeNER Spanish binary corpus.

(13/25). Both SENT and SPLIT normally outperform CONTEXT-ONLY or ASPECT-ONLY approaches. This confirms the intuition that it is important to take both context and aspect information for classification.

SENT with MT performs well on the OpeNER and MultiBooked datasets, but suffers on the SemEval and USAGE datasets. This is explained by the percentage of sentences that contain contrasting polarities in each dataset: between 8 and 12% for the OpeNER and MultiBooked datasets, compared to 29% for SemEval or 50% for USAGE. In sentences with multiple targets that have contrasting polarities, the SENT baseline performs poorly.

The CONTEXT-ONLY approach always performs better than ASPECT-ONLY, confirming that context is more important than the prior probability of an aspect being positive or negative.

The main outlier is MT multiclass, where the sentence-level baseline is nearly 10 percentage points better than the sentence splitting approach (38.8 F_1 versus 29.6). This is mainly a result of the performance on the OpeNER and MultiBooked corpora, where the sentence-level baseline is 20 performance points better than the SPLIT approach.

Finally, the general level of performance of projection-based aspect-level cross-lingual sentiment classification systems shows that they still lag 10+ percentage points behind MT on binary (compare MT (72.9 F_1) with BLSE (63.0)), and 6+ percentage points on multiclass (MT (38.8) versus MUSE (32.2)).

4.4.5 Error Analysis

We perform a manual analysis of the aspects misclassified by all systems on the OpeNER Spanish binary corpus (see Table 4.14), and found that the average length of misclassified aspects is slightly higher than that of correctly classified aspects, except for VECMAP. This indicates that averaging may have a detrimental effect

as the size of the aspect increases.

With the MT upperbounds, there is a non-negligible amount of noise introduced by aspects which have been incorrectly translated (0.05% OpeNER ES, 6% MultiBooked EU, 2% CA, 2.5% SemEval, 1% USAGE). We hypothesize that this is why MT with CONTEXT-ONLY performs better than MT with SPLIT. This motivates further research with projection-based methods, as they do not suffer from translation errors.

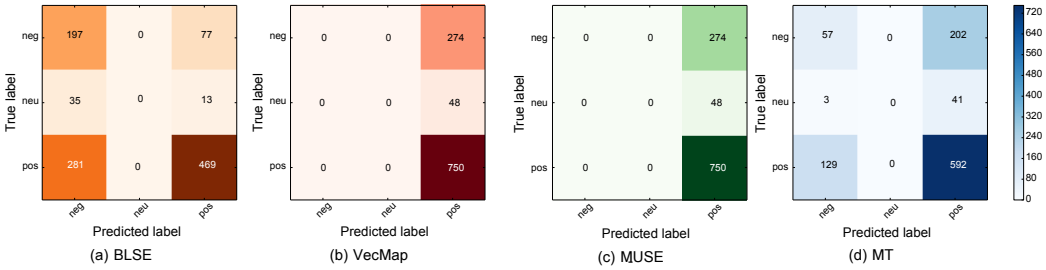


Figure 4.10: Confusion matrices for all SPLIT models on the SemEval task.

The confusion matrices of the models on the SemEval task, shown in Figure 4.10, show that on the multilabel task, models are not able to learn the neutral class. This derives from the large class imbalance found in the data (see Table 4.12). Similarly, models do not learn the Strong Negative class on the OpeNER and MultiBooked datasets. In the future, it may be beneficial to upsample these minority classes in order to improve performance.

4.4.6 Discussion

In this section we showed that is possible to learn a binary sentiment classifier that performs well for targeted sentiment analysis with little parallel data, even outperforming MT on multi-class targeted sentiment analysis on several datasets. However, MT still outperforms the projection methods for binary classification.

The low performance of all models on the multiclass setup reflects the difficulty of the task. Aspects which have neutral sentiment are often found within sentences that overall express positive or negative sentiment. With our approach, we do not require any syntactic information, but this may be necessary to improve results in the future. Cross-lingual dependency information transferred with MT has shown promise for cross-lingual sentiment analysis (Almeida et al., 2015). However, if this is to be used for under-resourced languages, it would be necessary to create high-quality bilingual dependency parsers, which are currently not available.

While we did not experiment with ensemble techniques in this section, experiments in Section 4.3 suggest that ensemble methods using MT and BLSE could improve performance. But this falls outside the goals of this thesis, as MT assumes the existence of parallel sentences. For under-resourced languages, it is more interesting that the projection based techniques perform nearly as well MT with much less parallel data.

4.5 Case Study: Real World Deployment

In the previous sections we developed methods to perform targeted cross-lingual sentiment analysis with little parallel data. Yet most of the experiments were performed on a two to three languages and only a few domains. It would be informative to know how well these methods work when tested on real world data, as many times we would like to know what opinion people from different countries and cultures have regarding the same entity.

Knowing how a machine learning model will perform on different target languages is another important facet of research, as different source-target combinations will likely lead to a change in results. There have been many previous works that have observed target-language specific differences in dependency parsing (Agić et al., 2016), machine translation (Johnson et al., 2017), and language modeling (Cotterell et al., 2018; Gerz et al., 2018). We are not aware of any work in sentiment analysis that explores the relationship between target language and performance in such depth.

Additionally, the effect of domain differences when performing cross-lingual tasks has not been studied in depth. Hangya et al. (2018) propose domain adaptation methods for cross-lingual sentiment classification and bilingual dictionary induction. They show specifically that creating domain-specific cross-lingual embeddings improves the classification for English-Spanish. However, the source-language training data used to train the sentiment classifier is taken from the same domain as the target-language test data. Therefore, it's not clear what the effect of using source-language training data from different domains would be. In this section, we explore this relationship in more depth.

In this section we detail a case study in which we deploy three targeted cross-lingual sentiment models on tweets in ten Western European languages. We use English as the source language in all experiments, and test on each of the ten target languages. We attempt to answer the following research questions:

- How much does the amount of monolingual data available to create monolingual embeddings effect the final results?
- How do features of the target language, *i. e.* similarity to source language or morphological complexity, affect the performance?
- How does domain mismatches between source-language training and target-language test data affect the performance?

In this section, we demonstrate that 1) the amount of monolingual data does not directly affect classification results, 2) language similarity between the source and

		EU	CA	GL	IT	FR	NL	DE	NO	SV	DA
Wiki	# sents. (M)	3.1	9.6	2.5	23.7	39.1	19.4	53.7	6.8	35.9	3.6
	# tokens (M)	47.9	143.7	51.0	519.6	771.8	327.3	902,1	110.5	457.3	64.4
Emb	# vocab. (k)	246.0	400.9	178.6	729.4	967.7	877.9	2,102.7	443.3	1,346.7	294.6
	dim.	300	300	300	300	300	300	300	300	300	300
Dict	# pairs	4,616	5,271	6,297	5,683	5,383	5,700	6,391	5,177	5,344	5,007

Table 4.15: Statistics of Wikipedia corpora, embeddings, and projection dictionaries.

target languages based on word and character n-gram distributions can predict the performance of BLSE on new datasets, and 3) domain mismatch often has a larger effect on BLSE than MT-based cross-lingual models.

4.5.1 Methodology

We collect Wikipedia dumps for ten languages; namely, Basque, Catalan, Galician, German, Italian, Dutch, French, Norwegian, Swedish and Danish. We then preprocess them using the Wikiextractor script[23], and sentence and word tokenize them with either Ixa pipes (Agerri et al., 2014) (Basque, Galician, Italian, Dutch, and French), Freeling (Padró et al., 2010) (Catalan), or NLTK (Loper and Bird, 2002) (Norwegian, Swedish, Danish).

For each language we create Skip-gram embeddings with the word2vec toolkit following the pipeline and parameters described in Section 2.2.1. This process gives us 300 dimensional vectors trained on similar data for all languages. We can assume that any large differences in the embedding spaces derive from the size of the data and the characteristics of the language itself.

We create projection dictionaries by translating the Hu and Liu dictionary Hu and Liu (2004) to each of the target languages and keeping only translations that are single word to single word, as described in Section 4.3.3.

The statistics of all Wikipedia corpora, embeddings, and projection dictionaries are shown in Table 4.15.

[23] http://attardi.github.io/wikiextractor/

The Sagrada Familia church
Parc Güell
La Boqueria market
Tibidabo
Santiago de Compostela
The Guggenheim Museum Bilbao
Txindoki
Anboto
The Eiffel Tower
The Louvre
Big Ben
The London Eye
Buckingham Palace
Akershus castle Oslo
The Oslo viking ship museum
The Gamla Stan Stockholm

Table 4.16: Touristic targets used as tweet search criteria.

4.5.2 Data Collection

In order to evaluate the effectiveness of targeted cross-lingual sentiment models on a large number of languages, we collect and annotate small datasets from twitter for each of the target languages, as well as a larger dataset to train the models in English. While it would be possible to only concentrate our efforts on languages with existing datasets in order to enable evaluation, this could give a distorted view of how well these models generalize.

Tourism is a topic where people often like to express their opinions on social media. With this in mind, we collect tweets directed at a number of tourist attractions in European cities using the Twitter API. We require these tourist attractions to be unambiguous, *i. e.* Barcelona would be unfit as it can be associated either with the city or with the football team. The list of tourist attractions used is found in Table 4.16.

A preliminary attempt to use only the mention of these tourist attractions led to tweets that were almost always neutral towards the target. Therefore, in an attempt to find a more varied sample, we download tweets that contain mentions of these tourist attractions as well as one of several emoticons or keywords[24]. This distant

[24]The emoticons and keywords we used were ":)", ":(", "good", "bad", and the translations of these last two words into each target language.

	EN	EU	CA	GL	IT	FR	NL	DE	NO	SV	DA
+	388	40	88	27	63	51	30	72	47	40	34
0	645	93	165	57	103	140	48	125	80	93	77
−	251	9	47	15	56	66	8	48	10	20	11
Cohen's κ	0.62	0.60	0.61	-	-	-	-	-	-	-	-

Table 4.17: Statistics of Tweet corpora collected for the deployment study, as well as inter-annotator agreement for English, Basque, and Catalan calculated with Cohen's κ.

supervision technique has been used to create sentiment lexicons (Mohammad et al., 2016), semi-supervised training data (Felbo et al., 2017), and features for a classifier (Turney and Littman, 2003).

We then remove any tweets that are less than 7 words long or which contain more than 3 hashtags or mentions. This increases the probability that a tweet contains enough information to correctly classify it based solely on the text in the tweet.

At sentence- or tweet-level it would be possible just to do a small revision, but at target-level the keywords and emoticons used to collect the tweets do not necessarily refer to the target in question. Therefore, we manually annotate all tweets for its polarity toward the target to insure the quality of the data. Any tweets that have unclear polarity towards the target are assigned a neutral label. This produces the three class setup that is commonly used in the SemEval tasks (Nakov et al., 2013, 2016). Finally, for a subset of tweets in English, Catalan, and Basque two annotators classify each tweet. We report the pairwise inter-annotator agreement using Cohen's κ. The final statistics for the tweets in each language are found in Table 4.17.

After annotation, the neutral class is the largest in all languages, followed by positive, and finally negative. These distributions are similar to those found in other twitter crawled datasets (Nakov et al., 2013, 2016). We calculate pairwise agreement on a subset of languages using Cohen's κ. The scores reflect a good level of agreement (0.60 - 0.62), indicating the reliability of these annotations.

4.5.3 Experiments

Since we predetermine the sentiment target for each tweet, we can perform targeted experiments without further annotation. We use the SPLIT models described in Section 4.4.2. Our model is the targeted BLSE models described in Section 4.4.

	Training Data	Model	EU	CA	GL	IT	FR	NL	DE	NO	SV	DA	AVERAGE
Binary		maj. class	46.0	39.5	38.8	34.6	36.1	44.1	37.5	43.4	45.2	40.0	40.5
	Twitter	BLSE	53.7	54.5	52.0	**63.4**	49.2	44.1	53.4	56.4	**65.3**	**68.3**	**56.0**
		VECMAP	56.4	48.1	33.3	38.2	48.6	55.2	51.0	59.0	60.5	43.4	49.4
		MT	41.3	41.4	**56.5**	39.7	54.5	43.3	55.1	52.2	49.8	55.6	48.9
		Ensemble	40.5	42.5	41.8	44.2	54.5	44.1	53.0	53.9	52.2	46.7	47.4
	USAGE	BLSE	36.4	44.5	46.8	59.4	50.4	52.2	44.6	57.7	65.2	44.3	50.1
		VECMAP	32.9	45.9	35.2	49.8	42.3	49.0	47.3	**59.2**	33.3	44.3	43.9
		MT	49.1	54.3	53.5	58.1	49.8	21.1	55.5	41.4	49.0	45.1	47.7
		Ensemble	48.2	55.5	42.7	57.1	50.4	28.9	53.3	48.4	44.5	48.0	47.7
	SemEval	BLSE	31.6	55.3	37.9	47.9	56.4	**70.3**	**58.3**	43.4	44.5	47.9	49.3
		VECMAP	**59.8**	**59.0**	45.6	55.3	**60.0**	55.9	39.7	43.4	48.2	40.0	50.7
		MT	57.0	58.7	40.5	58.2	49.0	61.6	57.6	40.3	53.8	50.8	52.8
		Ensemble	46.0	47.2	36.9	44.4	37.3	62.8	54.9	41.1	59.3	42.7	47.3
	Training Data	Model	EU	CA	GL	IT	FR	NL	DE	NO	SV	DA	AVERAGE
Multi-class		maj. class	26.6	23.7	24.3	21.1	23.5	23.9	22.5	26.1	24.6	25.2	24.1
	Twitter	BLSE	32.6	35.9	30.1	26.7	28.0	28.7	36.9	41.4	40.9	24.3	32.6
		VECMAP	26.5	30.2	**39.6**	26.7	**37.2**	34.6	**39.8**	31.7	33.4	**41.0**	34.1
		MT	37.3	34.1	33.9	**35.6**	35.6	**35.9**	32.5	**43.2**	38.6	39.6	**36.6**
		Ensemble	**41.5**	30.5	36.5	26.9	36.3	31.9	30.9	37.9	42.8	36.3	35.1
	USAGE	BLSE	29.3	**36.3**	35.2	34.0	27.7	27.8	36.9	24.3	41.0	40.6	33.3
		VECMAP	27.6	30.6	37.3	24.7	37.2	31.6	38.5	40.4	31.3	33.1	33.2
		MT	37.3	34.2	33.9	35.6	35.6	35.9	32.5	39.6	**43.2**	38.6	36.6
		Ensemble	41.5	30.5	36.5	26.9	36.3	31.9	30.9	37.9	42.8	36.3	35.1
	SemEval	BLSE	29.3	**36.3**	35.2	34.0	27.7	27.8	36.9	24.3	41.0	40.6	33.3
		VECMAP	27.6	30.6	37.3	24.7	37.2	31.6	38.5	40.4	31.3	33.1	33.2
		MT	37.3	34.2	33.9	35.6	35.6	35.9	32.5	39.6	**43.2**	38.6	36.6
		Ensemble	41.5	30.5	36.5	26.9	36.3	31.9	30.9	37.9	42.8	36.3	35.1

Table 4.18: Macro F_1 of targeted cross-lingual models on twitter data in 10 target languages.

Additionally, we compare to the targeted VECMAP and MT models. Finally, we set a majority baseline by assigning the most common label (neutral) to all predictions, as well as an Ensemble classifier that uses the predictions from BLSE and MT before taking the largest predicted class for classification (see Section 4.3 for details). All models are trained for 300 epochs with a learning rate of 0.001 and α of 0.3.

We train the three models on the English data compiled during this study, as well as on the USAGE, and SemEval English data (the details can be found in Table 4.12) and test the models on the target-language test set.

4.5.4 Results

Table 4.18 shows the macro F_1 scores for all cross-lingual targeted sentiment approaches (BLSE, VECMAP, MT) trained on English data and tested on the target-language using the SPLIT method proposed in 4.4.2. The final column is the average over all languages.

On the binary setup, BLSE outperforms all other cross-lingual methods including

MT, with 56.0 macro averaged F_1 across languages versus 49.4, and 48.9 for VECMAP, and MT respectively. BLSE performs particularly well on Catalan (54.5), Italian (63.4), Swedish (65.3), and Danish (68.3). VECMAP performs poorly on Galician (33.3), Italian (38.2), and Danish (43.4), but outperforms all other methods on Basque (56.4), Dutch (55.2) and Norwegian (59.0). MT performs the worst overall, although it does perform best for Galician (56.5). Unlike experiments in Section 4.3, the Ensemble approach does not perform better than the individual classifiers.

On the multiclass setup, however, MT (36.6 F_1) is the best, followed by VECMAP (34.1), and BLSE (32.6). Compared to the experiments on hotel reviews, the average differences between models is small (2.5 percentage points between MT and VECMAP, and 1.5 between VECMAP and BLSE). Again, all methods outperform the majority baseline.

On both the binary and multiclass setups, the best overall results are obtained by testing and training on data from the same domain (56.0 F_1 for BLSE and 36.6 F_1 for MT). Training MT and VECMAP on the SemEval data performs better on the binary setup than training on SemEval, however.

Finally, compared to the experiments performed on hotel and product reviews in Section 4.4, the noisy data from twitter is more difficult to classify. Despite the rather strong majority baselines (an average of 40.5 macro F_1 on binary and 24.1 on multiclass), no model achieves more than an average of 56 Macro F_1 on the binary task and 36 Macro F_1 on the multi-class task. Another marked difference is that MT is the worst model on the binary setup.

4.5.5 Error Analysis

An initial error analysis shows that all models suffer greatly on the negative class. We assumed that this was due to the data imbalance in the source-language training data, and performed additional experiments with sub-sampled and over-sampled balanced source-language data. These two approaches, however, gave poor results. This seems to suggest that negative polarity towards a target is more difficult to determine within these frameworks. A significant amount of the tweets that have negative polarity towards a target also express positive or neutral sentiment towards other targets. The averaging approach to create the context vectors does not currently allow any of the models to exclude this information, leading to poor performance on these instances.

Availability of Monolingual Unlabeled Data

It is possible that the variance in availability of unlabeled monolingual data for each language negatively affects the models. If the original word embedding spaces are not of high quality, this could make it difficult for the projection-based models to create useful features, as well as affecting the features used for classification in the MT approach. We plot the performance of the models as a function of available monolingual data in Figure 4.11.

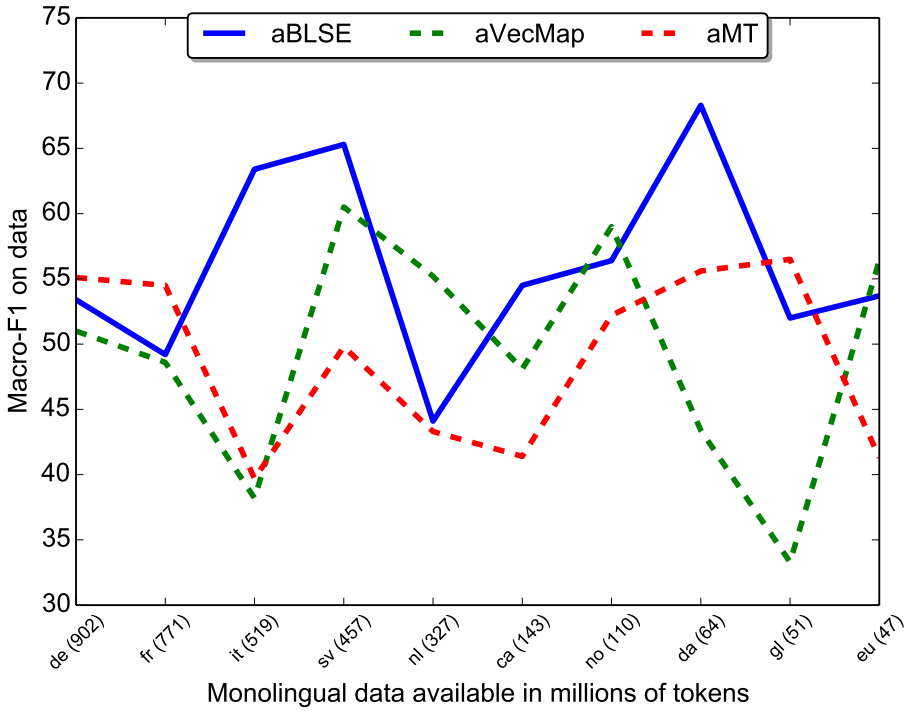


Figure 4.11: Performance (Macro F_1) on the binary task as a function of amount of monolingual data available in each language.

A statistical analysis of the amount of unlabeled data available and the performance of BLSE, VECMAP, and MT (Pearson's r = -0.14, 0.08, 0.17, respectively) reveals no statistically significant correlation between them.

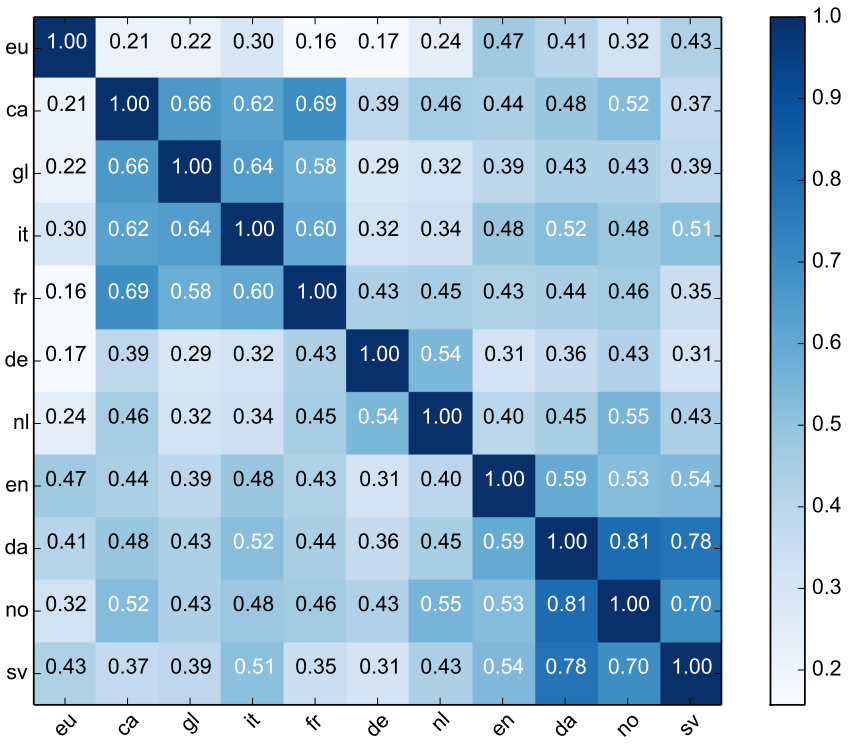


Figure 4.12: Cosine similarity of 3-gram POS-tag and 3-gram character frequency.

Language Similarity

We would like to know whether the similarity of the source and target languages can affect the classification performance of our cross-lingual models. Since all of the features we use for classification are derived from distributional representations, we will consider two aspects which directly influence the embedding representations: i) universal POS-tag n-grams which model the contexts used during training, and ii) character n-grams, which model differences in morphology. POS-tags have shown promise for determining genre (Fang and Cao, 2010), improving statistical machine translation (Lioma and Ounis, 2005), and the combination of POS-tag and character n-grams have proven useful features for identifying the native language of second language writers in English (Kulmizev et al., 2017). This indicates that these are useful features for characterizing a language. In this section we calculate the pairwise similarity between all languages and then check whether this correlates with performance.

We first POS-tag the test sentences obtained from twitter using the universal part of speech tags (Petrov et al., 2012) and calculate the normalized frequency distribution

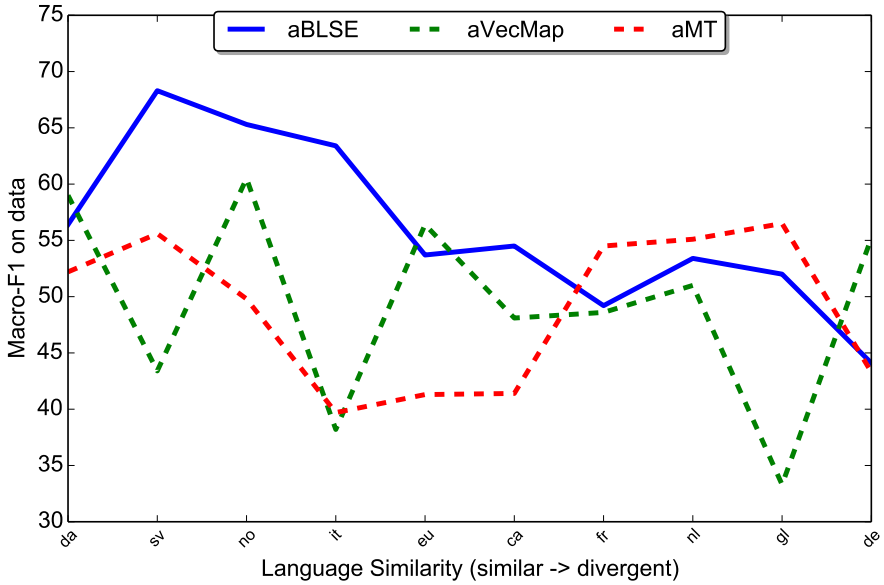


Figure 4.13: Performance (Macro F_1) on the binary task as a function of cosine similarity between POS-tag and character trigram distributions in the source language (EN) and the target languages.

of POS-tag trigrams and character trigrams for the two languages. We then measure the cosine similarity of the distributions between all language combinations. The pairwise similarities in Figure 4.12 confirm to expected similarities, and language families are clearly grouped (Romance, Germanic, Scandinavian, with Basque as a clear outlier). This confirms the use of our similarity metric for our purposes.

We additionally plot model performance as a function of language similarity in Figure 4.13. To measure the correlation between language similarity and performance, we calculate Pearson's r. We find that for BLSE there is a strong correlation between language similarity and performance, $r = 0.76$ and significance $p < 0.01$. VECMAP and MT do not show these correlations ($r = 0.24$ and $r = 0.14$, respectively). For MT this may be due to robust machine translation available in less similar languages according to our metric, such as German-English, which helps mediate this correlation. For VECMAP, however, it is less clear why it does not follow the same trend as BLSE.

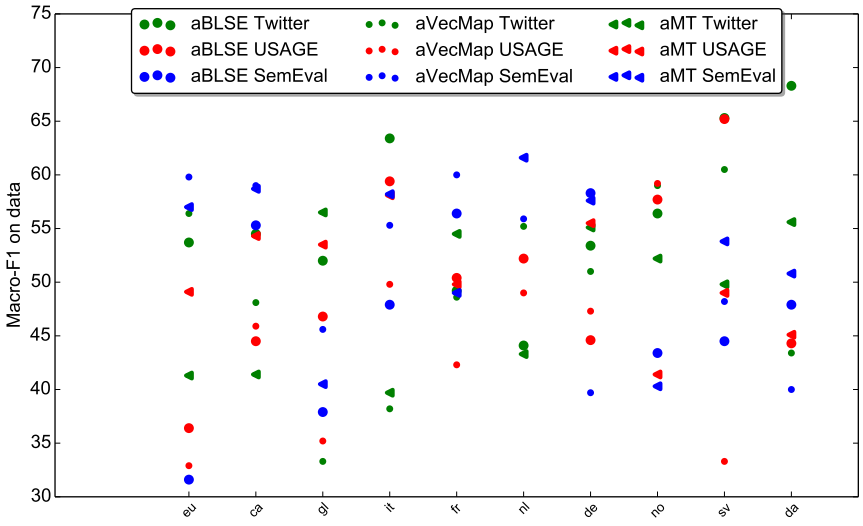


Figure 4.14: Performance (Macro F_1) on the binary task when each model is trained on data from different domains.

Domain Similarity

In this section, we determine the effect of source-language domain on the cross-lingual sentiment classification task. Specifically, we use English language training data from three different domains (twitter, restaurant reviews, and product reviews) to train the cross-lingual classifiers, and then test on the target-language twitter data. In monolingual sentiment analysis, one would expect to see a drop when moving to more distant domains, but in cross-lingual sentiment analysis, it's not clear that this is the case.

	twitter	SemEval	USAGE
twitter	1.000	0.749	0.749
SemEval	0.749	1.000	0.819
USAGE	0.749	0.819	1.000

Table 4.19: Domain similarity of English training data measured as Jennson-Shannon divergence between the most common 10,000 unigrams.

In order to analyze the effect of domain similarity further, we test the similar-

BLSE	VECMAP	MT
0.32 (0.08)	0.11 (0.55)	-0.07 (0.7)

Table 4.20: Pearson's r and p values for correlations between domain and performance of each model. There is a statistically insignificant effect of domain on BLSE, and no effect on VECMAP or MT.

ity of the domains of the source-language training data using Jensen-Shannon Divergence, which is a smoothed, symmetric version of the Kullback-Leibler Divergence, $D_{KL}(A||B) = \sum_i^N a_i \log \frac{a_i}{b_i}$. Kullback-Leibler Divergence measures the difference between the probability distributions A and B, but is undefined for any event $a_i \in A$ with zero probability, which is common in term distributions. Jensen-Shannon Divergence is then

$$D_{JS}(A, B) = \frac{1}{2}\Big[D_{KL}(A||B) + D_{KL}(B||A)\Big].$$

Our similarity features are probability distributions over terms $t \in \mathbb{R}^{|V|}$, where t_i is the probability of the i-th word in the vocabulary V.

For each domain, we create frequency distributions of the most frequent 10,000 unigrams that all domains have in common and measure the divergence with D_{JS}.

The results in Table 4.19 show us that both the SemEval and USAGE datasets are relatively distinct from the twitter data described in Section 4.5.2, while they are more similar to each other. The fact that both MT and VECMAP have an overall performance improvement on the binary setup when training on the SemEval data but not when trained on USAGE data seems, therefore, to be due to the larger size of the SemEval data, rather than to domain effects (plotted in Figure 4.14).

We calculate Pearson's r on the correlation between domain and model performance, shown in Table 4.20. The results show a negligible correlation for BLSE (0.32), with no significant correlation for VECMAP or MT. This suggests that the models are relatively robust to domain noise, or rather that there is so much other noise found in the approaches that domain is less relevant.

Translation Errors

Given the noisy nature of the twitter data and the lower performance of the MT method, we perform two analysis of possible errors: 1) mistranslated targets and

	EU	CA	GL	IT	FR	NL	DE	NO	SV	DA
targets	9.4%	7.7%	4.2%	9.0%	2.3%	0.0%	0.0%	6.6%	3.6%	9.8%
contexts	20.9%	19.7%	17.7%	10.8%	10.9%	14.0%	13.5%	8.8%	7.2%	15.7%

Table 4.21: Percentage of mistranslated targets and contexts per language.

$-$	si home!	yes man!	$+$
$-$	ja está bé!	it's fine!	$+$
$-$	etait en traveaux	it works	$+$
$-$	c'era troppa coda	there was tail	0
$+$	prou bé!	that's enough !	$-$
$+$	ça promet	it promises	0
$+$	darf nicht fehlen	may not be missing	0

Table 4.22: Common mistranslations of short phrases that indicate polarity.

2) translations that remove or change the sentiment with respect to the target word.

Given that we know the targets we wish to identify in each tweet, it is easy to find translation errors. We compare the translations to the list of possible target names in English, and if it is not within the list, it is considered an error. Table 4.21 shows the results. The most commonly mistranslated targets are "Big Ben" $\rightarrow$ "Big Leg" in the Scandinavian languages, and "Sagrada Familia" $\rightarrow$ "Holy Family" in the Romance languages. For German and Dutch, there were no mistranslations of targets.

We also perform a manual check of the contexts, only considering translation mistakes when they change the polarity expressed towards the target, shown in Table 4.21. The most common mistakes lead to a loss of polarity and a larger neutral class prediction. However, there are a number of examples which flip the polarity (see Example 6). There is a small correlation between context errors and MT performance (Pearson's r = -0.53 (0.1)), but not for mistranslations of targets (Pearson's r = -0.17 (0.6)).

(6) No em direu que despertar-se al Parc Güell no és una molt bona forma de despertar-se !
Not me tell that wake-up at Parc Güell not is a very good way to wake-up !
'I will not tell you to wake up at Parc Güell it's not a very good way to wake up!'

There are also some common informal phrases that carry a large amount of polarity and are consistently mistranslated. We include a short list of the most common mistakes in Table 4.22.

4.5.6 Discussion

In this section, we have performed a case study by deploying the models from Sections 4.3 and 4.4 to real world twitter data, which we collect and annotate for targeted sentiment analysis. We then looked in detail at phenomena that affect the performance of models. We found that for binary targeted sentiment analysis BLSE performs better than machine translation on noisy data from social media, although it is sensitive to differences between source and target languages. We have shown that there is little correlation between performance on the cross-lingual sentiment task and the amount of unlabeled monolingual data used to create the original embeddings spaces. Finally, we have performed an analysis of the different kinds of errors that MT introduces.

Unlike the experiments in Section 4.3, the ensemble classifier employed here was not able to improve the results. This is likely because the small size of the datasets does not allow for the classifier to learn which features are useful in certain contexts.

One common problem that appears when performing targeted sentiment analysis on noisy data from twitter is that many of the targets of interest are ambiguous, which leads to false positives. Even with relatively unambiguous targets like "Big Ben", there are a number of entities that can be referenced; Ben Rothlisberger (an American football player), an English language school in Barcelona, and many others. In order to deploy a full sentiment analysis system on twitter data, it will be necessary to disambiguate these mentions before classifying the tweets, either as a preprocessing step or jointly.

In sentiment analysis, it is not yet common to test a model on multiple languages, despite the fact that current state-of-the-art models are often theoretically language agnostic. This section shows that good performance in one language does not guarantee that a model will transfer well to other languages, even given similar resources. We hope that future work in sentiment analysis will make better use of the available test datasets.

The results in this section also lead us to a new set of research questions. Namely, if BLSE performs relatively well on cross-lingual tasks and requires only precomputed embedding spaces, will it be applicable to domain adaptation? We explore these questions in the next section.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde (2018). "Projecting Embeddings for Domain Adaptation: Joint Modeling of Sentiment Analysis in Diverse Domains." In: *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics*, pp. 818-830. http://aclweb.org/anthology/C18-1070.

4.6 Projecting Embeddings for Domain Adaptation

Besides the lack of training data in under-resourced languages, one of the main limitations of current approaches to sentiment analysis is that they are sensitive to differences in domain. This leads to classifiers that, after training, perform poorly on new domains (Pang and Lee, 2008; Deriu et al., 2017). Domain adaptation techniques provide a solution to reduce the discrepancy and enable models to perform well across multiple domains (Blitzer et al., 2007). The two main approaches to domain adaptation for sentiment analysis are *pivot-based methods* (Blitzer et al., 2007; Pan et al., 2010; Yu and Jiang, 2016), which augment the feature space with domain-independent features learned on unsupervised data, and *autoencoder* approaches (Glorot et al., 2011; Chen et al., 2012), which seek to create a good general mapping from sentence to a latent hidden space. While pivot-based domain adaptation methods are well-motivated, they are often outperformed by autoencoder methods. However, both approaches to domain adaptation effectively lead to a loss of information, as they must reduce the effect of discriminant features which are domain-dependent.

Unlike previous sections, in this section we concentrate not on cross-lingual sentiment analysis, but rather we propose a domain adaptation approach based on lessons learned from cross-lingual sentiment analysis. This approach maintains the domain-dependent features, while adapting them to the target domain. Following state-of-the-art approaches to create bilingual word embeddings (Mikolov et al., 2013a; Artetxe et al., 2016, 2017), we learn to project a mapping from a source domain vector space to the target domain space, while jointly training a sentiment classifier for the source domain.

We show that our proposed model (1) performs comparably to state-of-the-art models when domains are similar and (2) outperforms state-of-the-art models significantly on divergent domains. We report novel state-of-the-art results on 11 domain pairs. We also contribute a detailed error analysis and compare the effect of different projection lexicons. Our code is available at `http://github.com/jbarnesspain/domain_blse`.

The content of this section derives directly from the paper accepted at COLING 2018, mentioned in Section 1.4 (Barnes et al., 2018b).

4.6.1 Related Work and Motivation

Domain adaptation is an omnipresent challenge in natural language processing. It has been applied for many tasks, such as part-of-speech tagging (Blitzer et al.,

2006; Daume III, 2007), parsing (Blitzer et al., 2006; Finkel and Manning, 2009; McClosky et al., 2010), or named entity recognition (Daume III, 2007; Guo et al., 2009; Yu and Jiang, 2015). In the following, we limit ourselves to adaptation techniques which have been applied to sentiment analysis.

Pivot-based Approaches

Blitzer et al. (2006) propose *structural correspondence learning* (SCL), which introduces the concept of *pivots*. These are features that behave in the same way for discriminative learning for both domains, *e. g.*, *good* or *terrible* for sentiment analysis. The intuition is that non-pivot domain-dependent features, *e. g.*, *well-written* for the book domain or *reliable* for electronics, which are highly correlated to a pivot should be treated the same by a sentiment classifier.

Blitzer et al. (2007) extend their SCL approach to sentiment analysis and also create one of the benchmark datasets for domain adaptation in sentiment analysis. They crawl between 4000 and 7000 product reviews for each domain, and create balanced datasets of 1000 positive and 1000 negative reviews for four product types (books, DVD, electronics, and kitchen appliances). The remaining reviews serve as unlabeled training data for the SCL approach. For each pivot, they train a binary classifier to predict the existence of the pivot from non-pivot features. They then use these classifiers to create a domain-independent representation of the data. The concatenation of the original representation and the SCL representation are used to train a classifier.

Pan et al. (2010) also exploit the relationship between pivots and non-pivots to span the domain gap, but use a graph-based approach to cluster non-pivot features and augment the original feature space. Yu and Jiang (2016) learn sentence embeddings that are useful across domains through multi-task learning. They jointly train a convolutional recurrent neural network model to predict the sentiment of source domain sentences while at the same time predicting the presence of pivots. Finally, Ziser and Reichart (2017) propose neural structural correspondence learning (NSCL), which marries SCL and autoencoder techniques by using a neural network to create a hidden representation of a text, and then using this representation to predict the existence of pivots.

NSCL is currently state of the art, but requires a careful choice of pivot features and extensive hyper-parameter searches to achieve the best results.

Autoencoder Approaches

Glorot et al. (2011) adopt a deep learning approach for domain adaptation. They create lower-dimensional representations for their data through the use of *stacked denoising autoencoders* (SDA), which are trained to reconstruct the original sentence from a corrupted version. They then train a linear SVM on the original feature space augmented with the hidden representations obtained from the autoencoder.

Chen et al. (2012) extend this work by proposing *Marginalized Denoising Autoencoders* (MSDA), which are more scalable thanks to a series of linear transformations which are performed in closed-form, with the non-linearity being applied afterwards. This leads to a significant gain in speed, as well as the ability to include more features from the original representations. Autoencoder models perform better than earlier SCL models (excluding NSCL), but have the disadvantages of being less interpretable, requiring long training times, and only utilizing a small amount of the original feature space.

Domain Specific Word Representations

A third approach is to create word representations that provide useful features for multiple domains. He et al. (2011) propose a joint sentiment-topic model which uses pivots to change the topic-word Dirichlet priors. Bollegala et al. (2015) create domain-specific embeddings for pivots and non-pivots with the constraint that the pivot representations are similar across domains.

The work that is most similar to ours is that of Bollegala et al. (2014). Their method learns to predict differences in word distributions across domains by learning to project lower-dimensional SVD representations of documents across domains. Unlike our work, however, they learn the projection step separately from the classification. They also only learn to project the features that the two domains have in common, which implies discarding information useful for classification. These approaches, however, perform worse than MSDA and NSCL.

4.6.2 Projecting Representations

In this section, we cast domain adaptation for sentiment analysis as a version of this cross-lingual adaptation in which the source and target domains have a large shared vocabulary. However, as is the case in domain adaptation, words do not necessarily have the same semantics across domains. Therefore, we will use the

aforementioned BLSE projection model to learn a word-level projection from one domain to another, while jointly learning to classify the source domain.

Target-domain Classification

For inference, we classify sentences from a target-domain corpus C_{target}. As in the training procedure, for each sentence, we take the word embeddings from the target embeddings T and average them to $\mathbf{a}_i \in \mathbb{R}^d$. We then project this vector to the joint space $\hat{\mathbf{z}}_i = \mathbf{a}_i \cdot M'$. Finally, we pass $\hat{\mathbf{z}}_i$ through a softmax layer P to get our prediction $\hat{y}_i = \text{softmax}(\hat{\mathbf{z}}_i \cdot P)$.

4.6.3 Experimental Setup

We compare our method with two adaptive baselines and one non-adaptive version. In the following, we describe the six evaluation corpora and the baselines.

Datasets

Amazon Corpora In order to evaluate our proposed method, we use the corpus collected by Blitzer et al. (2007) , which consists of Amazon product reviews from four domains: books (B), DVD (D), electronics (E), and kitchen (K). Each subcorpus contains a balanced labeled subset, with 1000 positive and 1000 negative reviews, as well as a much larger set of unlabeled reviews. We use the standard split of 1600 reviews from each domain as training data and the remaining 400 reviews as validation data. For testing, we use all of the 2000 reviews from the target domain (Ziser and Reichart, 2017).

We take the unlabeled data from each domain to create the domain embeddings for our method, as well as to train the domain independent representations for the NSCL and MSDA methods. In order to create embeddings for the Amazon corpora, we concatenate all of the unlabeled data from all domains. The statistics for this corpus are given in Table 4.23.

SemEval Corpora Sentiment analysis of Twitter data is common nowadays, with several popular shared tasks organized on the topic (Nakov et al., 2013, 2016). In order to evaluate how well domain adaptation techniques perform on large domain gaps, we also use the message polarity classification corpora provided by the organizers of SemEval 2013 and 2016 (Nakov et al., 2013, 2016). We will

		B	D	E	K	S13	S16
Unlabeled		973,194	122,438	21,009	17,856	49M	49M
Train	+	800	800	800	800	2,225	2,468
	−	800	800	800	800	831	664
Dev	+	200	200	200	200	328	682
	−	200	200	200	200	163	310
Test	+	1000*	1000*	1000*	1000*	946	5,619
	−	1000*	1000*	1000*	1000*	316	2,386
Total		2,000	2,000	2,000	2,000	4,809	12,129

Table 4.23: Statistics for the Amazon corpora (Books, DVD, Electronics, Kitchen), as well as the SemEval 2013 and 2016 message classification tasks (S13 and S16 respectively). * For the Amazon corpora, we test on the entire target domain corpora.

refer to these as S13 and S16, respectively. These contain tweets which have been annotated for positive, negative, and neutral sentiment. We remove neutral tweets, giving us a binary setup which allows compatibility with the Amazon corpora. The statistics for these corpora are given in Table 4.23.

Embeddings

For BLSE, we create mono-domain embeddings using the Word2Vec toolkit[25]. We train skip-gram embeddings with 300 dimensions, subsampling of 10^{-4}, window of 5, negative sampling of 15 on the concatenation of the unlabeled Amazon corpora. We also create Twitter-specific embeddings by training on nearly 8 million tokens taken from tweets collected using various hashtags. The parameters were the same as those used to create the Amazon embeddings. We use a vector initialized randomly between -0.25 and 0.25 for out-of-vocabulary words to approximate the variance of the pretrained vectors.

Baselines and Model

Domain transfer for sentiment analysis has been widely studied on the Amazon sentiment domain corpus. However, we believe that progress previous approaches

[25] https://code.google.com/archive/p/word2vec/

have made on this particular corpus may not hold when tested on more divergent domains. Therefore, we compare two state-of-the-art approaches on the Amazon corpus with our method, as well as a standard non-adaptive baseline.

NOAD is a non-adaptive approach which uses a bag-of-words representation from each review as features for a linear SVM.

MSDA is the original implementation of marginalized Stacked Denoising Autoencoders (Chen et al., 2012), one of the state-of-the-art domain adaptation methods on the Amazon sentiment domain corpus. The approach learns a latent hidden representation of the data, which is then concatenated to the original feature space. For our experiments, we use the 30000 most common uni- and bi-grams as features and take the top 5000 features as pivots (Chen et al., 2012). We tune the corruption level (0.5, 0.6, 0.7, 0.8, 0.9) and the C-parameter for the SVM classifier on the source domain validation data, but leave the number of layers at 5.

NSCL is an approach that marries both the pivot-based methods and autoencoders. Specifically, we use the original implementation[26] of the Autoencoder SCL with Similarity Regularization, which we refer to as NSCL. This approach substitutes the reconstruction weights of the autoencoder with a matrix of the pre-trained word embeddings of pivots. This allows the model to generalize beyond boolean features. We set the hyper-parameters for training the autoencoders with stochastic gradient descent to those from the original paper[27] and tune the number of pivots (100, 200, 300, 400, 500), dimensionality of the hidden layer (100, 300, 500), and C-parameter for Logistic Regression on the source domain validation data (400 reviews).

BLSE is our approach based on cross-lingual vector projection. We use the domain-specific word embeddings to initialize our model and following the embedding literature, we take the most common 20,000 words in the concatenated corpora as a projection dictionary (see Section 4.6.5). We tune the hyper-parameters training epochs, alpha (0.1–0.9), and batch sizes (20–500) on the source domain validation data.

4.6.4 Results

Tables 4.24 and 4.25 present the results of our experiments. We report accuracy scores for the balanced Amazon corpora. For the unbalanced SemEval corpora,

[26] `https://github.com/yftah89/Neural-SCL-Domain-Adaptation`

[27] We set the learning rate to 0.1, momentum to 0.9 and weight-decay regularization to 10^{-5}.

	D→B	E→B	K→B	B→D	E→D	K→D	B→E	D→E	K→E	B→K	D→K	E→K
BLSE	**82.2**	71.3	69.0	81.0	**76.8**	**76.5**	71.8	70.3	70.8	73.8	72.3	78.3
NSCL	77.3	71.2	**73.0**	**81.1**	74.5	76.3	**76.8**	**78.1**	**84.0**	**80.1**	**80.3**	**84.6**
MSDA	76.1	**71.9**	70.0	78.3	71.0	71.4	74.6	75.0	82.4	78.8	77.4	84.5
NOAD	73.6	67.9	67.7	76.0	69.2	70.2	70.0	70.9	81.6	74.0	73.2	82.4

Table 4.24: Sentiment classification accuracy for the Blitzer et al. (2007) task (Books (B), DVD (D), Kitchen (K), Electronics (E)).

	B→S13	D→S13	E→S13	K→S13	B→S16	D→S16	E→S16	K→S16
BLSE	**65.8**	**67.1**	**65.6**	**63.9**	**65.2**	**66.1**	**67.0**	**62.8**
NSCL	62.8	60.6	59.2	50.7	61.5	61.9	60.7	57.6
MSDA	52.2	45.3	48.8	53.2	53.1	43.1	48.2	55.6
NOAD	61.6	61.5	60.9	51.8	59.6	63.2	59.3	54.2

Table 4.25: Sentiment classification macro F_1 for the SemEval 2013 and 2016 tasks in binary setup, (Books (B), DVD (D), Kitchen (K), Electronics (E), SemEval 2013 (S13), SemEval 2016 (S16)).

we present macro F_1 scores. We introduce the notation X→Y, where X is the train corpus and Y is the test corpus, to indicate the domain pairs.

On the Amazon corpora, NSCL outperforms the other approaches (3.6 accuracy points on average compared to BLSE, 2.5 compared to MSDA, and 5.1 compared to NOAD). BLSE only performs better than NSCL on three setups (DVD to books, electronics to DVD, and kitchen to DVD) and MSDA on four setups (DVD to books, books to DVD, electronics to DVD, and kitchen to DVD). BLSE performs much better on the books and DVD test sets than the electronics and kitchen test sets. This can be explained by the fact that the corpora used to train the Amazon embeddings contain many more unlabeled reviews for books and DVDs (973,194 / 122,438 respectively) than electronics and kitchen (21,009 / 17,856). Consequently, the vector representations for sentiment words that only appear in the books and DVD subcorpora are of higher quality than those that only appear in the electronics and kitchen subcorpora (see Table 4.27). Since BLSE relies entirely on the embeddings as input, the lower quality of what should be discriminative features affects the classification.

For the SemEval corpora (see Figures **??** and **??**), BLSE significantly outperforms all other models (8.2, 15.5, and 6.4 F_1 better on average compared to NSCL, MSDA, and NOAD, respectively). NSCL is better than MSDA on 7 of the 8 setups, but better than the NOAD baseline on only 4. MSDA performs particularly poorly here and only outperforms the baseline on one setup. We suspect that this may

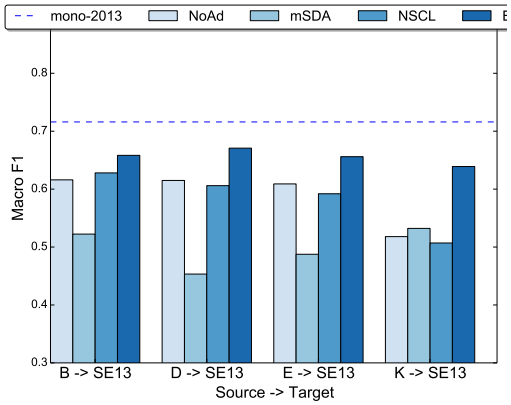


Figure 4.15: F_1 of approaches trained on the source dataset and tested on the 2013 SemEval corpus.

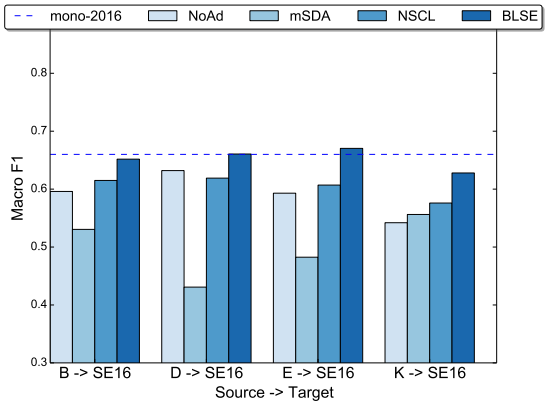


Figure 4.16: F_1 of approaches trained on the source dataset and tested on the 2016 SemEval corpus.

be caused by the substantial differences in the source and target corpora and the way this effects the representation given to the classifier, which we explore in more detail in Section 4.6.4.

Domain Divergence and Feature Sparsity

From the initial results, it seems that the BLSE model performs better on more divergent domains when compared to other state-of-the-art models. In order to analyze this further, we test the similarity of our domains using the Jensen-Shannon Divergence, which is a smoothed, symmetric version of the Kullback-Leibler Divergence, $D_{KL}(A||B) = \sum_i^N a_i \log \frac{a_i}{b_i}$. Kullback-Leibler Divergence measures the difference between the probability distributions A and B, but is undefined for any event $a_i \in A$ with zero probability, which is common in term distributions. Jensen-Shannon Divergence is then

$$D_{JS}(A, B) = \frac{1}{2}\Big[D_{KL}(A||B) + D_{KL}(B||A)\Big] .$$

Our similarity features are probability distributions over terms $t \in \mathbb{R}^{|V|}$, where t_i is the probability of the i-th word in the vocabulary V.

For each domain, we create frequency distributions of the most frequent 10,000 unigrams that all domains have in common and measure the divergence with D_{JS}. The results in Table 4.26 make it clear that the SemEval datasets are more distant from the Amazon datasets than the Amazon datasets are from each other. This is

	book	DVD	elect.	kitchen	SemEval 2013	SemEval 2016
book	1.000	**0.940**	0.870	0.864	0.775	0.802
DVD		1.000	0.873	0.866	0.790	0.814
elect.			1.000	**0.908**	0.748	0.769
kitchen				1.000	0.741	0.761
SemEval 2013					1.000	**0.921**
SemEval 2016						1.000

Table 4.26: Jensen-Shannon divergence between term distribution representations of datasets. The **bold** numbers represent the most similar domains and underlined numbers represent the most divergent.

	Books		Electronics	
word	admires	conceit	indispensable	cumbersome
neighbors	professes	conceits	career.this	choppiness
	unselfish	macgruffen	non-western	setups
	parminder	pretentiously	mindwalk	forgiveable
	well-liked	contrivance	all-too-rare	unweildy

Table 4.27: Words and their nearest neighbors for important domain-dependent sentiment words. The nearest neighbors for the two example words from the book domain are more coherent than those of the electronics domain.

especially true for the distance between the SemEval datasets from the Kitchen dataset (D_{JS} = 0.741 and 0.761, respectively). This suggests that NSCL and MSDA give the best results when the difference between domains is relatively small, whereas BLSE performs better on more divergent datasets.

On the SemEval datasets, BLSE also benefits from using dense representations, rather than the sparse unigram and bigram features of NSCL and MSDA. This is particularly important when you have less domain overlap and smaller texts (the average number of features for the Amazon corpora is 76, compared to 17 for SemEval). BLSE is always able to find useful features, even if the tweet is quite short, whereas a bag-of-words representation can be so sparse that it is not helpful.

Error Analysis

We perform a label-based error analysis of the models on the SemEval 2013 and 2016 datasets by checking the error rate for the positive and negative classes, which

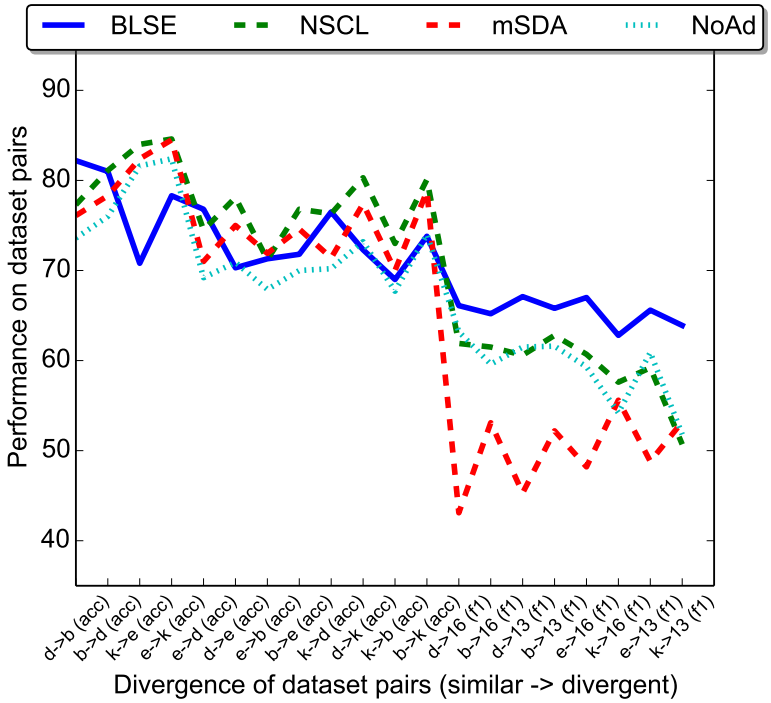


Figure 4.17: Plot of performance of each model as a function of domain similarity. The x axis plots the rank of similarity from most similar (left) to least similar (right). BLSE maintains its performance as the similarity decreases.

we define as

$$\text{Error Rate} = \frac{e_c}{n_c}, \tag{4.5}$$

where the number of errors e_c in class c is divided by the total number of examples n_c in the class. The results are found in Table 4.28. In general, BLSE has better overall performance than NSCL or MSDA. In fact, MSDA performs poorly on the minority negative class, with error rates reaching 98 percent. NSCL almost always favors a single class, with error rates as high as 60.4 on negative and 70.5 on positive.

4.6.5 Choice of projection lexicon

Given that the choice of projection lexicon is one of the key parameters in the BLSE model, we experiment with three approaches to creating a projection lexicon

		SemEval 2013		SemEval 2016	
		Pos	Neg	Pos	Neg
Books	BLSE	26.9	35.4	31.8	33.0
	NSCL	34.2	43.0	28.9	43.5
	MSDA	1.4	92.7	1.4	89.5
DVD	BLSE	22.8	39.2	28.0	36.5
	NSCL	18.1	60.4	18.7	52.1
	MSDA	0.2	97.8	0.2	98.2
Elec.	BLSE	19.1	48.7	27.7	34.6
	NSCL	35.2	41.5	38.9	33.7
	MSDA	1.1	93.7	0.8	91.6
Kitchen	BLSE	21.6	49.1	23.3	50.1
	NSCL	63.6	19.3	70.5	13.2
	MSDA	2.4	90.5	2.4	85.8

Table 4.28: Error rates for positive and negative classes for BLSE, NSCL, and MSDA trained on the Amazon corpora and tested on the SemEval corpora.

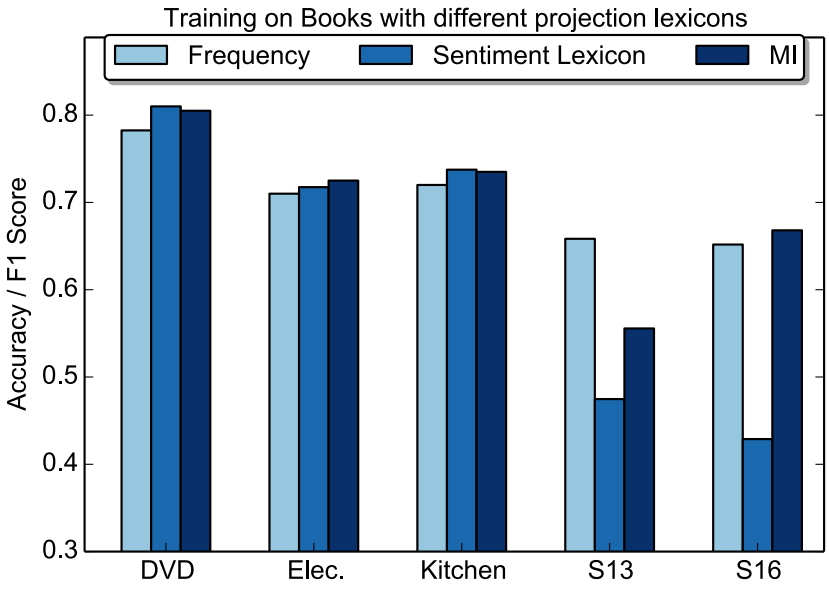


Figure 4.18: The effect of different projection lexicons for the BLSE method when training on the books domain.

and observe their effect on the books to SemEval 2013 setup.

The most frequent source words are a common source of projection lexicon in the multilingual embedding literature (Faruqui and Dyer, 2014; Lazaridou et al., 2015). For our experiment, we take the 20,000 most frequent tokens from the Brown corpus (Francis and Kučera, 1979). The hypothesis behind using a general corpus is that learning the best projection across domains will give the best results.

Sentiment Lexicons often contain domain-independent words that convey sentiment. In our model, using a sentiment lexicon as a translation dictionary is equivalent to the use of pivots in other frameworks, as these are usually domain independent words with are good predictors of sentiment. For our experiment, we take a subset of the sentiment lexicon from Hu and Liu (2004) which is found in the Amazon and SemEval corpora. The final version has 1130 words.

Mutual Information Selected Pivots have been shown to be a good predictor of sentiment across domains (Blitzer et al., 2006; Pan et al., 2010; Ziser and Reichart, 2017). We experiment with using words with the highest mutual information scores as a projection lexicon. For each source and target domain pair, we take unigrams and bigrams with high mutual information scores that appear at least 10 times in both domains. The number of pivots differs with each domain pair. The lowest number is 100 (DVD to SemEval 2013) and the highest 955 (books to DVD), with an average of 470 per domain pair.

Figure 4.18 shows that the frequency-based lexicon gives better results on the more divergent datasets, while the sentiment lexicon performs slightly better on the similar datasets, but poorly on the divergent datasets. The mutual information induced pivot lexicons provide good results on all but the SemEval 2013 dataset. This is likely because the lexicon is too small to give a good mapping.

4.6.6 Discussion

We have presented an approach to domain adaptation which learns to project mono-domain embeddings to a bi-domain space and use this bi-domain representation to predict sentiment. We have experimented with 20 domain pairs and shown that for highly divergent domains, our model shows substantial improvement over state-of-the-art methods. Our model constitutes a novel state of the art on 11 of the 20 domain pairs.

One of the main advantages of this approach is that the learned classifier can be used to classify sentiment in either of the two domains without further tuning. In the future, we would like to extend this model to learn multiple domain mappings

at a time, effectively permitting zero-shot domain adaptation at a large scale. This would enable a single model to predict sentiment for a number of domains.

Another promising avenue for improvement is to create lexicons that map concepts from the source domain to the target domain, *i. e.*, "read" in the books domain to "watch" in the movies domain. It would be interesting to see if it is possible to use vector algebra (Mikolov et al., 2013a) to find similar concepts in different domains, *e. g.*, *read - books + DVD = watch*. It would also be beneficial to map multiword units across domains, *e. g.*, "not particularly exciting" in DVD to "not very reliable" in electronics. This could be particularly helpful for moving beyond a binary view of sentiment at document-level, where domain adaptation would be of particular use, given that the cost of annotation is higher for multi-class, sentence-, or aspect-level classification.

A current disadvantage of our model might be that it uses skip-gram embeddings trained on more than one domain. Therefore, it would be of interest to investigate if methods which create domain specific embeddings (He et al., 2011; Bollegala et al., 2014, 2015) are able to give better results within our framework.

Chapter 5

CONCLUSION

5.1 Conclusion

In this thesis, we introduced a low-resource approach to cross-lingual sentiment analysis, as well as a new aspect-level sentiment analysis dataset for two under-resourced languages; namely, Catalan and Basque. This dataset also allows for research into the effects of variables such as word order or morphology on cross-lingual sentiment analysis, as both corpora are drawn from the same domain and annotated with the same guidelines.

In Chapter 4, we outlined the main experiments conducted during the thesis. Section 4.1 compared eight machine learning approaches to monolingual sentiment analysis on six benchmark datasets. The results confirmed the general consensus that BILSTMS are strong baselines across many NLP tasks. Section 4.2 compared three cross-lingual models that rely on distributional representations to two machine translation approaches for cross-lingual sentiment analysis. We described the limitations of distributional representations and hypothesized that they did not contain enough sentiment information to provide useful features for cross-lingual sentiment analysis.

Section 4.3 introduced a joint model that learns to project monolingual embeddings to a bilingual space which is simultaneously optimized to predict sentiment. Section 4.4 proposed several methods to perform targeted cross-lingual sentiment analysis by building upon state-of-the-art bilingual embedding methods, including the BLSE method introduced in the previous section. We concentrated only on target classification, leaving the integration of target and subjective phrase identification for future work. The fact that BLSE performed best on binary classification, while

MUSE performed best on multiclass classification leaves the possibility of looking at combining the strongest aspects of both methods.

Section 4.5 details a case study of deploying our model on ten target languages for targeted sentiment analysis for tourism. We created a small test dataset for the ten target languages. BLSE performed best on the binary task, outperforming both VECMAP and MT by a large margin. We showed that the overall performance of our models is limited to a degree by the similarity between source and target languages, while the amount of unlabeled monolingual data available in the target language has little effect. Finally we performed a detailed analysis of the errors introduced by machine translation.

In Section 4.6 we reformulated domain adaptation within the embedding projection framework. Treating each domain as if it were a separate language, we learn to project the embeddings to a bi-domain space, which is jointly optimized for sentiment. BLSE outperformed state-of-the-art models on distant domains, demonstrating that our approach generalizes well to other tasks.

Returning to the original aims and research questions posed in the introduction, there are a number of affirmations we can now make.

Do monolingual vector spaces contain enough distributional information for a sentiment classifier to learn to both project them to a common space and learn to classify sentiment? Monolingual vector spaces do have enough information to perform cross-lingual sentiment analysis by first projecting to a bilingual space and then learning to classify. The experiments using VECMAP and MUSE suggest that these techniques work relatively well, but do not contain enough sentiment information. Jointly learning the projection and classification, *i. e.* BLSE, often performs much better. This is especially true on binary classification tasks.

How much parallel data is necessary to perform the transfer? The amount of projection data needed is quite small, between 500-5000 translation pairs. This makes our approach to cross-lingual sentiment analysis fast to develop. Additionally, we have found that translating a small sentiment lexicon is more useful than using larger general purpose dictionaries.

How much source language annotated data is necessary to learn to classify the target language? From these experiments, even small amounts of labeled

data, *e. g.* 1000 labeled sentiment phrases, were enough to transfer the sentiment information to a target language.

What amount of loss of accuracy does this joint model suffer when compared to monolingual models? All cross-lingual models perform worse than monolingual models. Machine translation generally performs better than bilingual embedding methods, but obviously requires an order of magnitude more parallel data. On clean data, our models perform similarly to machine translation on the binary task, but significantly worse on multiclass classification tasks. On noisy twitter data, however, projection based models outperform machine translation.

Is it possible to improve machine-translation based CLSA methods using this approach? Yes, we have shown that ensemble methods that make use of machine translation and projection-based methods improve the state-of-the-art. The use of multi-view cross-lingual representations is a promising avenue for future research.

Given a bilingual sentiment representation, is it better to assume that all aspects in a phrase have the same polarity, or try to predict each separately? It is almost always better to split the sentence into contexts, as proposed in Section 4.4.2. This is especially true for datasets that often have multiple aspects in a single sentence, such as the SemEval and USAGE datasets. MT does not seem to follow this trend, but this is likely because of the number of errors and mismatches to the aspects caused by the use of MT.

Can we predict the sentiment of aspects in a target language without using machine translation? Yes. In fact, for binary sentiment analysis, bilingual word embedding approaches can outperform machine translation at aspect-level. This is likely due to the fact that machine translation introduced more changes to the original data than bilingual word embeddings, namely lexical changes, reordering, and loss of information. These have a bigger impact on finer-grained sentiment analysis, as there is less redundancy in the signal.

5.2 Future Work

The current performance of the projection-based techniques still lags behind state-of-the-art MT approaches on most tasks, indicating that there is still much work to be done. While general bilingual embedding techniques do not seem to incorporate enough sentiment information, they are able to retain the semantics of their word vectors to a large degree even after projection. We hypothesize that the ability to retain the original semantics of the monolingual spaces leads to MUSE performing better than MT on multiclass aspect-level sentiment analysis. The joint approach introduced in this thesis suffers from the degradation of the original semantics space, while optimizing the sentiment information. Adding more regularization to our model in order to maintain the original structures to a higher degree could potentially help in this.

In the future, including multiword expressions within the projection of our framework should help in part with the loss of performance on multiclass sentiment analysis, as the model would learn to project phrases with similar semantics closely together.

One problem that arises when using bilingual embeddings instead of machine translation is that differences in word order are no longer handled. Machine translation models, on the other hand, always include a reordering element. Nonetheless, there is often a mismatch between the real source language word order and the translated word order. In this thesis, we avoided the problem by using a bag of embeddings representation, but as we showed in experiments in Section 4.1, this approach does not perform as well as approaches that take word order into account, such as LSTMS or CNNS. We leave the incorporation of these classifiers into our framework for future work.

The recent introduction of Unsupervised Machine Translation (Artetxe et al., 2018; Lample et al., 2018b) may also introduce new avenues to explore CLSA for under-resourced languages, avoiding in large part the need for large amounts of parallel data. The need for long training times and the word ordering problems, however, will remain. This means that the approaches proposed here will still be of interest.

Experimenting with ensembles of MT and BLSE showed that our model captures information that MT does not, and that the combination of the two leads to state-of-the-art results. In the future, it may be interesting to perform more research on ways to combine MT and cross-lingual distributed models.

Multi-view cross-lingual representations pRastogi2015,Ammar2016 also show promise on many tasks. It may be possible to create a single classifier for groups of

language families, so that languages with more resources within the family are able to help the under-resourced ones. This approach has enabled zero-shot translation (Johnson et al., 2017), so it is not impossible to imagine that multilingual zero-shot modeling would work well for a comparatively easy task such as sentiment analysis.

Finally, we showed that this approach can be used for a number of languages in a multilingual deployment scenario. This could drastically reduce the time and money spent on translating documents to determine sentiment in crisis scenarios.

Bibliography

Agerri, R., Bermudez, J., and Rigau, G. (2014). Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3823–3828.

Agerri, R., Cuadros, M., Gaines, S., and Rigau, G. (2013). OpeNER: Open polarity enhanced named entity recognition. In *Sociedad Española para el Procesamiento del Lenguaje Natural*, volume 51, pages 215–218.

Agić, Ž., Johannsen, A., Plank, B., Martínez Alonso, H., Schluter, N., and Søgaard, A. (2016). Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Akhtar, M. S., Sawant, P., Sen, S., Ekbal, A., and Bhattacharyya, P. (2018). Solving data sparsity for aspect based sentiment analysis using cross-linguality and multi-linguality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 572–582.

Alkorta, J., Gojenola, K., Iruskieta, M., and Perez, A. (2015). Using relational discourse structure information in basque sentiment analysis. In *Proceedings of the 5th Workshop on RST and Discourse Studies at SEPLN(2015)*, pages 1–10.

Allen, R. B. (1987). Several studies on natural language and back-propagation. In *Proceedings of IEEE First International Conference on Neural Networks*, pages 335–341.

Almeida, M. S. C., Pinto, C., Figueira, H., Mendes, P., and Martins, A. F. T. (2015). Aligning opinions: Cross-lingual opinion mining with dependencies. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 408–418.

Arruda, G. D. d., Roman, N. T., and Monteiro, A. M. (2015). An annotated corpus

for sentiment analysis in political news. In *Proceedings of the 2015 Symposium in Information and Human Language Technology*, pages 101–110.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2289–2294.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2018). Unsupervised neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Aue, A. and Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Bakliwal, A., Foster, J., van der Puil, J., O'Brien, R., Tounsi, L., and Hughes, M. (2013). Sentiment analysis of political tweets: Towards an accurate classifier. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 49–58.

Balahur, A. and Turchi, M. (2014). Comparative experiments usingsupervised learning and machine translation for multilingual sentiment analysis. *Computer Speech & Language*, 28(1):56–75.

Balamurali, A. R., Khapra, M. M., and Bhattacharyya, P. (2013). Lost in translation: Viability of machine translation for cross language sentiment analysis. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 2*, CICLing'13, pages 38–49.

Banea, C., Mihalcea, R., and Wiebe, J. (2010). Multilingual subjectivity: Are more languages better? In *Proceedings of COLING 2010, the 23th International Conference on Computational Linguistics: Technical Papers*, pages 28–36.

Banea, C., Mihalcea, R., and Wiebe, J. (2013). Porting multilingual subjectivity resources across languages. *IEEE Transactions on Affective Computing*, 99(PrePrints).

Banea, C., Mihalcea, R., Wiebe, J., and Hassan, S. (2008). Multilingual subjectivity analysis using machine translation. In *Proceedings of the 2008 Conference on*

Empirical Methods in Natural Language Processing (EMNLP 2008), pages 127–135.

Barnes, J., Klinger, R., and Schulte im Walde, S. (2017). Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–12.

Barnes, J., Klinger, R., and Schulte im Walde, S. (2018a). Bilingual sentiment embeddings: Joint projection of sentiment across languages. In *Proceedings of the 56th Annual Meeting of the Association for Comutational Linguistics (Volume 1: Long Papers)*, pages 2483–2493.

Barnes, J., Klinger, R., and Schulte im Walde, S. (2018b). Projecting embeddings for domain adaption: Joint modeling of sentiment analysis in diverse domains. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics: Technical Papers*, pages 818–830.

Barnes, J., Lambert, P., and Badia, T. (2016). Exploring distributional representations and machine translation for aspect-based cross-lingual sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1613–1623.

Barnes, J., Lambert, P., and Badia, T. (2018c). Multibooked: A corpus of basque and catalan hotel reviews annotated for aspect-level sentiment classification. In *Proceedings of 11th Language Resources and Evaluation Conference (LREC'18)*, pages 656–660.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, abs/1206.5533.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). Aneural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Blitzer, J., Dredze, M., and Pereira, F. (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.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 120–128.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bollegala, D., Maehara, T., and Kawarabayashi, K.-i. (2015). Unsupervised cross-domain word representation learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 730–740.

Bollegala, D., Weir, D., and Carroll, J. (2014). Learning to predict distributions of words across domains. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 613–623.

Bosco, C., Lai, M., Patti, V., Pardo, F. M. R., and Rosso, P. (2016). Tweeting in the debate about catalan elections. In *Proceedings of the 2016 LREC workshop on Emotion and Sentiment Analysis Workshop (ESA)*, pages 67–70.

Bosma, W., Vossen, P., Soroa, A., Rigau, G., Tesconi, M., Marchetti, A., Monachini, M., and Aliprandi, C. (2009). KAF: A generic semantic annotation format. In *Proceedings of the Generative Lexicon (GL2009) Workshop on Semantic Annotation*.

Buys, J. and Botha, J. A. (2016). Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '16, pages 1954–1964.

Cauchy, A.-L. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Mathematique Academie des Sciences, Paris*, 25:536–538.

Chandar, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 1853–1861. Curran Associates, Inc.

Chen, M., Xu, Z., Weinberger, K. Q., and Sha, F. (2012). Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International*

Coference on International Conference on Machine Learning, ICML'12, pages 1627–1634.

Chen, P., Sun, Z., Bing, L., and Yang, W. (2017). Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 452–461.

Chen, X., Athiwaratkun, B., Sun, Y., Weinberger, K. Q., and Cardie, C. (2016). Adversarial deep averaging networks for cross-lingual sentiment classification. *CoRR*, abs/1606.01614.

Cheng, J., Kartsaklis, D., and Grefenstette, E. (2014). Investigating the role of prior disambiguation in deep-learning compositional models of meaning. *Learning Semantics Workshop NIPS 2014*, 2(1):1–5.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Chollet, F. (2015). Keras. `https://github.com/fchollet/keras`.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurements*, 20:37–46.

Cohn, T. and Lapata, M. (2007). Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Committee, A. (2018). Acl 2018: Statistics on submissions and reviewing. `https://acl2018.org/2018/03/12/reviewing-statistics/`. Accessed: 2018-13-06.

Cotterell, R., Mielke, S. J., Eisner, J., and Roark, B. (2018). Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541.

Das, S. R. and Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388.

Daume III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 519–528.

Demirtas, E. and Pechenizkiy, M. (2013). Cross-lingual polarity detection with machine translation. In *Proceedings of the International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '13*, pages 9:1–9:8.

Deriu, J. M., Weilenmann, M., Von Gruenigen, D., and Cieliebak, M. (2017). Potential and limitations of cross-domain sentiment classification. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 17–24.

Dinu, G., Lazaridou, A., and Baroni, M. (2015). Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), workshop track*.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., and Xu, K. (2014). Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Duh, K., Fujino, A., and Nagata, M. (2011). Is machine translation ripe for cross-lingual sentiment classification? *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, 2:429–433.

Esuli, A. and Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM International*

Conference on Information and Knowledge Management, CIKM '05, pages 617–624.

Esuli, A. and Sebastiani, F. (2006). Senti-wordnet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th Language Resources and Evaluation Conference (LREC'06)*, pages 417–422.

Fang, A. C. and Cao, J. (2010). Enhanced genre classification through linguistically fine-grained pos tags. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 85–94.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

Faruqui, M. and Dyer, C. (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.

Faruqui, M. and Kumar, S. (2015). Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 1351–1356.

Feedforward Neural Networks (2018). Feedforward neural networks. https://brilliant.org/wiki/feedforward-neural-networks. Accessed: 2018-07-05.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 1615–1625.

Fellbaum, C. (1999). Wordnet. Wiley Online Library.

Ferreira, D. C., Martins, A. F. T., and Almeida, M. S. C. (2016). Jointly learning to embed and predict with multiple languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2019–2028.

Finkel, J. R. and Manning, C. D. (2009). Hierarchical bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference*

of the North American Chapter of the Association for Computational Linguistics, pages 602–610.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis (Special Volume of the Phological Society)*, 1952-1959:1–32.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Flekova, L. and Gurevych, I. (2016). Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041.

Fokkens, A., Soroa, A., Beloki, Z., Ockeloen, N., Rigau, G., Robert van Hage, W., and Vossen, P. (2014). Naf and gaf: Linking linguistic annotations. In *Proceedings 10th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*.

Francis, W. N. and Kučera, H. (1979). The Brown Corpus: A Standard Corpus of Present-Day Edited American English. Brown University Liguistics Department.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gerz, D., Vulić, I., Ponti, E. M., Reichart, R., and Korhonen, A. (2018). On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, page To Appear.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 513–520.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gouws, S., Bengio, Y., and Corrado, G. (2015). BilBOWA: Fast bilingual distributed representations without word alignments. *Proceedings of The 32nd International Conference on Machine Learning*, pages 748–756.

Gouws, S. and Søgaard, A. (2015). Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 1386–1390.

Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., and Su, Z. (2009). Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 281–289.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, 11.

Hangya, V., Braune, F., Fraser, A., and Schütze, H. (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 (Volume 1: Long Papers)*, pages 810–820.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hartung, M., Klinger, R., Schmidtke, F., and Vogel, L. (2017). Ranking right-wing extremist social media profiles by similarity to democratic and extremist groups. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 24–33.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181.

He, Y., Lin, C., and Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 123–131.

Hermann, K. M. and Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68.

Hochreiter, S., Bengio, Y., and Frasconi, P. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kolen, J. and Kremer, S., editors, *Field Guide to Dynamical Recurrent Networks*. IEEE Press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, pages 168–177.

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, volume 3, pages 2096–2104.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691.

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

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Kamps, J., Marx, M., Mooken, R. J., and de Rijke, M. (2004). Using wordnet to measure semantic orientations of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, LREC '04, pages 1115–1118.

Kartsaklis, D. (2014). Compositional operators in distributional semantics. *Springer Science Reviews*, 2(1):161–177.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.

Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Kim, E. and Klinger, R. (2018). Who feels what and why? annotation of a literature

corpus with semantic roles of emotions. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics: Technical Papers*.

Kim, S.-M. and Hovy, E. (2006). Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 483–490.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014)*.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012, the 24th International Conference on Computational Linguistics: Technical Papers*, pages 1459–1474.

Klinger, R. and Cimiano, P. (2013). Bi-directional inter-dependencies of subjective expressions and targets and their value for a joint model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 848–854.

Klinger, R. and Cimiano, P. (2014). The USAGE review corpus for fine grained multi lingual opinion analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2211–2218.

Klinger, R. and Cimiano, P. (2015). Instance selection improves cross-lingual model training for fine-grained sentiment analysis. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 153–163.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86.

Kulmizev, A., Blankers, B., Bjerva, J., Nissim, M., van Noord, G., Plank, B., and Wieling, M. (2017). The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

Lakew, S. M., Cettolo, M., and Federico, M. (2018). A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics: Technical Papers*, pages 641–652.

Lambert, P. (2015). Aspect-level cross-lingual sentiment classification with constrained smt. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 781–787.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018a). Word translation without parallel data. In *International Conference on Learning Representations*.

Lample, G., Denoyer, L., and Ranzato, M. (2018b). Unsupervised machine translation using monolingual corpora only. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Lazaridou, A., Dinu, G., and Baroni, M. (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 and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Lioma, C. and Ounis, I. (2005). Deploying part-of-speech patterns to enhance statistical phrase-based machine translation resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 163–166.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan and Claypool.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion Observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th international World Wide Web conference (WWW-2005)*.

Liu, F., Cohn, T., and Baldwin, T. (2018). Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 278–283.

Loper, E. and Bird, S. (2002). NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, pages 63–70.

Lu, B., Tan, C., Cardie, C., and K Tsou, B. (2011). Joint bilingual sentiment classification with unlabeled parallel corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 320–330.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 151–159.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150.

McClosky, D., Charniak, E., and Johnson, M. (2010). Automatic domain adaptation for parsing. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 28–36.

Meng, X., Wei, F., Liu, X., Zhou, M., Xu, G., and Wang, H. (2012). Cross-lingual mixture model for sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 572–581.

Merity, S., Keskar, N. S., and Socher, R. (2018). Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

Mihalcea, R., Banea, C., and Wiebe, J. (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.

Mikolov, T., Corrado, G., Chen, K., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *CoRR*.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Mnih, A. and Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 2012 International Conference on Machine Learning (ICML)*.

Mogadala, A. and Rettinger, A. (2016). Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–702.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, pages 321–327.

Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55:95–130.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In Lin, D. and Wu, D., editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 412–418.

Na, J.-C., Sui, H., Khoo, C., Chan, S., and Zhou, Y. (2004). Effectiveness of simple

linguistic processing in automatic sentiment classification of product reviews. In *Knowledge Organization and the Global Information Society: Proceedings of the Eighth International ISKO Conference*, pages 49–54.

Nahar, V., Unankard, S., Li, X., and Pang, C. (2012). Sentiment analysis for effective detection of cyber bullying. In *Proceedings of the 14th Asia-Pacific International Conference on Web Technologies and Applications*, APWeb'12, pages 767–774.

Nair, V. and Hinton, G. E. (2010a). Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814.

Nair, V. and Hinton, G. E. (2010b). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 2010 International Conference on Machine Learning (ICML)*.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320.

Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. (2016). Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459.

Noreen, E. (1989). *Computer intensive methods for testing hypotheses*. Wiley.

Olah, C. (2018). Long short-term memory networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 2018-07-05.

Oliphant, T. E. (2006). *A Guide to Numpy*. Trelgol Publishing. `http://www.numpy.org`.

Padó, S. (2006). User's guide to `sigf`: Significance testing by approximate randomisation. `https://nlpado.de/~sebastian/software/sigf.shtml`.

Padó, S. and Lapata, M. (2009). Cross-lingual annotation projection of semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Padró, L., Collado, M., Reese, S., Lloberes, M., and Castellón, I. (2010). Freeling 2.1: Five years of open-source language processing tools. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC'10)*.

Padró, L. and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

Pagolu, V. S., Reddy, K. N., Panda, G., and Majhi, B. (2016). Sentiment analysis of twitter data for predicting stock market movements. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 1345–1350.

Pan, J., Xue, G.-R., Yu, Y., and Wang, Y. (2011). Cross-lingual sentiment classification via bi-view non-negative matrix tri-factorization. In *15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 289–300.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 751–760.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical methods in natural language processing-Volume 10 (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Paszke, A., Gross, S., Chintala, S., and Chanan, G. (2016). Pytorch deeplearning framework. http://pytorch.org. Accessed: 2017-08-10.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418.

Polanyi, L. and Zaenen, A. (2006). Contextual valence shifters. In *Computing Attitude and Affect in Text: Theory and Applications*, pages 1–10.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Popat, K., Balamurali, A. R., Bhattacharyya, P., and Haffari, G. (2013). The haves and the have-nots: Leveraging unlabelled corpora for sentiment analysis. In *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, volume 1, pages 412–422.

Pourdamghani, N. and Knight, K. (2017). Deciphering related languages. In

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 2513–2518.

Prettenhofer, P. and Stein, B. (2011). Cross-lingual adaptation using structural correspondence learning. *ACM Transactions on Intelligent Systems and Technology*, 3(1):1–22.

Rasooli, M. S., Farra, N., Radeva, A., Yu, T., and McKeown, K. (2017). Cross-lingual sentiment transfer with limited resources. *Machine Translation*.

Reitan, J., Faret, J., Gambäck, B., and Bungum, L. (2015). Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.

Ruder, S. and Plank, B. (2017). Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 372–382.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Schuff, H., Barnes, J., Mohme, J., Padó, S., and Klinger, R. (2017). Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23.

Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *Transactions on Signal Processing*, 45(11):2673–2681.

Severyn, A. and Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1631–1642.

Søgaard, A., Agić, Ž., Martínez Alonso, H., Plank, B., Bohnet, B., and Johannsen, A. (2015). Inverted indexing for cross-lingual nlp. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1713–1722.

Søgaard, A., Johannsen, A., Plank, B., Hovy, D., and Martínez Alonso, H. (2014). What's in a p-value in nlp? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 1–10.

Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., and Zhou, M. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. on Knowl. and Data Eng.*, 28(2):496–509.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the Association for Computational Linguistics (ACL)*, pages 417–424.

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

Upadhyay, S., Faruqui, M., Dyer, C., and Roth, D. (2016). Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the ACL (volume 1: Long Papers)*, pages 1661–1670.

Uryupina, O., Plank, B., Severyn, A., Rotondi, A., and Moschitti, A. (2014). Sentube: A corpus for sentiment analysis on youtube social media. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Villena-Román, J., Lana-Serrano, S., Eugenio, M.-C., and González-Cristóbal, J. C. (2013). Tass - workshop on sentiment analysis at sepln. *Procesamiento del Lenguaje Natural*, 50(0):37–44.

Villena-Román, J., Martínez-Cámera, E., García Morera, J., and Jiménez Zafra, S. M. (2015). Tass 2014 - the challenge of aspect-based sentiment analysis. *Procesamiento del Lenguaje Natural*, 54(0):61–68.

Vulic, I. and Moens, M. (2016). Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, 55:953–994.

Vulić, I. and Moens, M.-F. (2014). Probabilistic models of cross-lingual semantic similarity in context based on latent cross-lingual concepts induced from comparable data. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 349–362.

Vulić, I. and Moens, M.-F. (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 and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725.

Wan, X. (2009a). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243.

Wan, X. (2009b). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual meeting of the ACL and the 4th international joint conference on natural language processing.*, pages 235–243.

Wang, B., Liakata, M., Zubiaga, A., and Procter, R. (2017). Tdparse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 483–493.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120.

Wang, S., Mazumder, S., Liu, B., Zhou, M., and Chang, Y. (2018). Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 957–967.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 625–631.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (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.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)*, pages 347–354.

Xiao, M. and Guo, Y. (2012). Multi-view adaboost for multilingual subjectivity analysis. In *Proceedings of COLING 2012, the 24th International Conference on Computational Linguistics: Technical Papers*, pages 2851–2866.

Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T.-Y. (2014). Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1219–1228.

Xue, W. and Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Yang, B. and Cardie, C. (2013). Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649.

Yang, K., Yu, N., Valerio, A., and Zhang, H. (2006). Widit in trec-2006 blog track. In *NIST Special Publication*.

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of COLING 2000, the 18th Conference on Computational linguistics*, pages 947–953.

Yu, J. and Jiang, J. (2015). A hassle-free unsupervised domain adaptation method using instance similarity features. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 168–173.

Yu, J. and Jiang, J. (2016). Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 236–246.

Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2017). Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 534–539.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550.

Zhang, M., Zhang, Y., and Vo, D. T. (2015). Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 612–621.

Zhang, M., Zhang, Y., and Vo, D.-T. (2016). Gated neural networks for targeted sentiment analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3087–3093.

Zhang, Y. and Wallace, B. (2017). A sensitivity analysis of (and practitioners guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263.

Zhou, G., Zhu, Z., He, T., and Hu, X. T. (2016). Cross-lingual sentiment classification with stacked autoencoders. *Knowledge and Information Systems*, 47(1):27–44.

Zhou, H., Chen, L., Shi, F., and Huang, D. (2015). Learning bilingual sentiment word embeddings for cross-language sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*

and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 430–440.

Zhou, X., Wan, X., and Xiao, J. (2012). Cross-language opinion target extraction in review texts. In *IEEE 12th International Conference on Data Mining*, pages 1200 – 1205.

Zhu, X., Guo, H., Mohammad, S., and Kiritchenko, S. (2014). An empirical study on the effect of negation words on sentiment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 304–313.

Ziser, Y. and Reichart, R. (2017). Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410.

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1393–1398.

Chapter 6

APPENDIX 1: STATISTICAL ANALYSIS OF STATE-OF-THE-ART MODELS

	BOW	AVE	RETROFIT	JOINT	LSTM	BILSTM	CNN
BOW		*SST-fine SenTube-A SenTube-T SemEval*	*SST-fine OpeNER SenTube-A SemEval*	*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*	*SST-fine SST-binary OpeNER SemEval*	*SST-fine SST-binary OpeNER SemEval*	*SST-binary OpeNER SenTube-A SenTube-T*
AVE	3		*SST-fine SST-binary SenTube-A SenTube-T*	*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*	*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*	*SST-fine SST-binary OpeNER SenTube-A SemEval*	*SST-fine SST-binary OpeNER SenTube-A SemEval*
RETROFIT	3	3		*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*	*SST-fine SST-binary OpeNER SenTube-A SemEval*	*SST-fine SST-binary OpeNER SenTube-A SemEval*	*SST-fine SST-binary SenTube-A SemEval*
JOINT	3	3	3		*SST-fine SST-binary OpeNER SenTube-A SenTube-T*	*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*	*SST-fine SST-binary OpeNER SenTube-A SenTube-T SemEval*
LSTM	4	5	4	3		*SemEval*	*SST-fine OpeNER SenTube-A SenTube-T SemEval*
BILSTM	4	5	5	4	1		*SST-fine OpeNER SenTube-A SenTube-T SemEval*
CNN	2	3	2	3	0	0	

Table 6.1: Results of the statistical analysis described in Section 4.1 for the best performing dimension of embeddings, where applicable. Datasets where there is a statistical difference (above diagonal) and number of datasets where a model on the Y axis is statistically better than a model on the X axis (below diagonal).