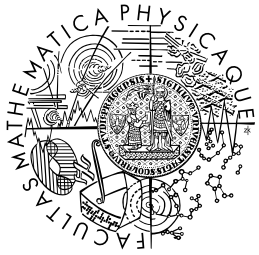


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Vladan Glončák

Using Syntactic Features for Opinion Target Identification

Institute of Formal and Applied Linguistics

Supervisor of the master thesis:	Jan Hajič Ph.D.
Study programme:	Computer Science
Study branch:	Computational Linguistics

Prague 2019

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... signature of the author

I would like to thank the supervisor of the thesis Mrg. Jan Hajič, jr. Ph.D. and the consultant Mgr. Kateřina Lesch Ph.D. for their support, advice and a patience throughout the process of writing this thesis.

I would also like to thank to RNDr. Milan Straka Ph.D. for advice with TensorFlow and his great lectures on Neural Networks.

I would also like to thank to my family and all my friends, namely Ľubomír Ohman, who lent me his computer when mine bailed on me during the work on this thesis, and Patrick Zehm, Ian Rastogi, Anastasia Serebryannikova for not bailing on me.

Title: Using Syntactic Features for Opinion Target Identification

Author: Vladan Glončák

Institute: Institute of Formal and Applied Linguistics

Supervisor: Jan Hajič Ph.D., Institute of Formal and Applied Linguistics

Consultant: Mgr. Kateřina Lesch Ph.D.

Abstract: Opinion Target Extraction (OTE) is a well-established subtask of sentiment analysis. While detecting sentiment polarity is useful in itself, the ability to extract the targets of the opinions allows for more thorough decision making. For example, an owner of a restaurant needs to know whether the guests are complaining about the food, or the ambience, or any other aspect of their establishment, etc.

Despite the lexical information being crucial for the task, syntactic structures have potential in being used to correctly decide among multiple candidate entities. Rules based on such structures have been used previously for the task. The objective of this thesis is to investigate, whether syntactic information influences the behavior of the state-of-the-art models such as recurrent neural networks for the OTE task. We did not find any substantial evidence to suggest that adding the syntactic information influences the behavior of the models.

Keywords: Opinion Target Identification, Sentiment Analysis, Syntax, Universal Dependencies, Machine Learning

Contents

Introduction **3**

1 Sentiment Analysis **6**
1.1 Objectives . . . 6
1.2 Methods . . . 6
1.2.1 Word Embeddings . . . 7

2 Syntactic Features **8**
2.1 Universal Dependencies . . . 8
2.1.1 Syntactic Features . . . 9
2.1.2 Morphological Features . . . 10
2.1.3 UDPipe . . . 11
2.2 Subjectivity Lexicon . . . 12
2.3 Feature Overview . . . 14

3 Data **16**
3.1 Data from Semeval 2016 Task 5 . . . 16
3.2 Customer Reviews in Czech . . . 18
3.3 Other Datasets . . . 21

4 Related Work **22**
4.1 Opinion Target Extraction . . . 22
4.1.1 Lexical-Based Methods . . . 22
4.1.2 Machine Learning Methods . . . 23
4.1.3 Syntactic Features . . . 23
4.2 English Dataset of Restaurant Reviews . . . 24
4.3 Czech Dataset of Customer Reviews . . . 25

5 Methods **26**
5.1 Data Preprocessing . . . 26
5.2 Baseline . . . 27
5.3 CRF . . . 27
5.4 Neural Networks . . . 28
5.5 Evaluation Metrics . . . 29
5.5.1 Analysis of the Results . . . 30

6 Experiments **31**
6.1 Baseline Model . . . 31
6.2 CRF Models . . . 31
6.3 Models with LSTM . . . 31

7 Results **33**
7.1 Baseline model . . . 33
7.2 English Dataset of Restaurant Reviews . . . 33
7.2.1 CRF models . . . 33
7.2.2 LSTM . . . 35

7.2.3 Overview of Models 36
7.3 Performance on the SemEval Test Data 36
7.4 Czech Dataset of Customer Reviews 39
7.4.1 CRF models 39
7.4.2 Overview of Models 39

Conclusion **44**

Bibliography **46**

List of Figures **52**

List of Tables **53**

List of Abbreviations **55**

A Attachments **56**
A.1 Source Code 56
A.2 Confusion Tables for UPOS 57
A.3 Confusion Tables for DEPREL 58
A.4 Confusion Tables for XPOS 59
A.5 Confusion Tables for FEATS 60
A.6 Confusion Tables for HEAD UPOS 61
A.7 Confusion Tables for HEAD DEPREL 62
A.8 Confusion Tables for SENT 63
A.9 Confusion Tables for HEAD SENT 64
A.10 Confusion Tables for SIBLING SENT 65

Introduction

Expressing and understanding emotions is one of the basic human cognitive abilities. Emotions are neither rational nor objective, yet they still drive our lives. Aside from non-verbal means of expressing emotions, such as facial expression, tone, intonation etc., verbal communication is one of the main tool to express emotional meaning. There is no clear definition of what emotional meaning means, and this notion varies from discipline to discipline. In general, emotional meaning can be anything that is not "*an objective description of any event, situation or mental state*" [Veselovská, 2017, p. 2].

It is clear that understanding subjective information in the textual data can be used for many practical applications, such as predicting market trends, criminal investigation, etc. The abundance of data in Information Age urges for automation of the data processing and analysis. Unfortunately, a large portion of the data is unstructured—not organized in a way that is easy to process. Due to the lack of predefined structure, ambiguities arise.

Aspect-Based Sentiment Analysis

Ambiguities from unstructured textual data are the focus of computational linguistics. In particular, *sentiment analysis* is concerned with extracting subjective information, such as emotion or opinion of the author, which is why it is also referred to as *opinion mining*. In general, the two main categories of tasks of sentiment analysis are *polarity detection* and *subjectivity detection*. Subjectivity detection aims to decide whether the information is factual or non-factual. On the other hand *polarity detection* classified the information, most commonly as positive or negative. This can be done at multiple levels, such as phrases, sentences, documents, etc.

In this thesis, we are concerned with identifying the target (source) of the opinion, i.e. the objects or topics towards which the opinion is expressed. This identification can be also done on various levels. For a high-level object of evaluation, aspects of the target can be also identified, which is referred to as *aspect based sentiment analysis* (ABSA).

The issue with ABSA is that the number of things that can be aspect is immense, and therefore some manual categories have to be designed. We are concerned with extraction of the targets at sentence level, also known as opinion target extraction (OTE). OTE does not require any manual engineering of categories—the task is to locate an explicit mention of a target in the data.

OTE can be useful to aggregate large amount of unstructured data, or produce summaries automatically. This is particularly useful, in the information age, since the amount of text that would otherwise had to be read by people is massive.

Methods in OTE

This can be accomplished using a dictionary of words and phrases that are often being evaluated in the data, such as restaurant. However, this approach is susceptible to producing both false positives, and false negatives (a target will not be identified). For example:

(1) "*I've eaten at many different Indian restaurants.*"[1]

is a sentence from a restaurant review that, per se, does not express any opinion. This would produce a false positive.

On the other hand, the size of the dictionary is limited, and the targets often come from open word classes. If the target of the evaluation is a name of a place or a dish, it will likely not be recognized, i.e. a false negative. For example:

(2) "***Rao** is a good restaurant, but it's nothing special.*"

"Rao" is the target of the evaluation, but unless it is a very popular place, it will not be in any dictionary.

Therefore, we hope that modern machine learning methods can be used to tackle those problems. Machine learning methods have been used for this task by Hu and Liu [2004], Popescu and Etzioni [2007], Liu et al. [2015], Tamchyna et al. [2015] and many others. Based on the previous work in the field, we have selected two popular methods for the experiments in this thesis—conditonal random fields (CRF) and recurrent neural networks, in particular long short-term memory (LSTM) units.

Syntactic Features

Part of what makes this problem particularly hard is that a large context may have to be considered. On a high-level, the pragmatic aspect needs to be considered. Sometimes cultural knowledge is necessary to understand what is being evaluated, particularly if the author uses irony. On the low-level, the context can be viewed as the surrounding words.

Relations between words in phrases and sentences are studied within syntax. In this thesis, we aim to employ those relations to improve the performance of the aforementioned models. **The objective of this thesis is not to find a model configuration with the best possible performance, but explore how adding features influences the performance of the models.** The features that we propose are mostly based on the syntactic relations of the words, therefore we refer to the as syntactic, although they may not be entirely based on syntax.

Recently, NLP methods have been shifting from heavily language dependent methods to more universal approaches. Universal Dependencies is a linguistic framework that is a successful example of such approach. It is designed with automatic processing in mind, which makes it a suitable tool for the objective of this thesis.

Straka and Straková [2017] developed UDPipe, a trainable tool for automatic dependency parsing. Words of the sentence are organized into a tree structure that captures how the words depend on each other. There are different kinds of this dependency relation. For example, we can say that subject and object both depend on some verb in a sentence, but the type of relation is different, hence the two different terms "subject" and "object".

We use this tool to augment the raw textual data with some additional information. Strictly speaking, we are not adding any information—it is already

[1] All the examples in quotation marks come from the dataset described in Section 3.1, targets are in bold.

in the textual data. A few thousand sentences used for a particular OTE task may not be enough to understand the complexity of a language as a whole. We hope that this can be used to boost performance, especially for small datasets, similarly to how basis expansion can improve the performance of a model.

A sentence containing subjective information typically contains a sentiment expression in addition to the target of the evaluation. Consider the following sentences:

(3) "*The* ***waiter*** *was attentive.*"

(4) The waiter was young.

One states a subjective evaluation, the other merely states a fact.

Therefore, we used lexicons of subjective terms to help the models identify phrases that carry some subjective meaning. Such lexicons have previously been employed by Veselovská and Tamchyna [2014] and Tamchyna et al. [2015] for the same task. In particular, Veselovská and Tamchyna [2014] used hand-crafted rules based on syntactic relations which required the information whether a word carries emotion or an opinion. In combination with the features based on the dependency relations, the models may be able to learn similar rules to correct some errors.

Outline

This thesis is structured as follows. Chapter 1 provides a brief overview of the field of Sentiment Analysis and the task of this thesis. Chapter 2 describes the motivation for the features that we propose for the experiments. The data that we are using for the experiment are described in Chapter 3, followed by Chapter 4 that lists the work previously published for the task and the data. Chapter 5 describes the how we processed the data and evaluated the results. In Chapter 6 we provide a description of the conducted experiments, results of which are presented in Chapter 7.

1. Sentiment Analysis

Sentiment analysis (SA) is a field that is concerned with extracting subjective information from discourse. In this thesis we consider SA to be a subfield of natural language processing (NLP)—a subfield of computer science concerned with interaction of computers and human languages.

With the growing popularity of social networks and review websites, SA recently became an area of focus due to its the practical application and the amount of available data. SA can be helpful in automatic processing of large amount of reviews, predicting market trends or even forensic linguistics.

1.1 Objectives

In general, SA systems can be used to extract any information that is not an objective description of the state of the world. Two major subtasks of SA are *subjectivity classification*, i.e. determining whether the utterance contains an opinion at all, and *polarity classification*, i.e. determining the polarity of the opinion. Another established tasks within SA are emotion detection and intent analysis. Emotion detection is simply detection of emotions, such as happiness or anger, from the piece of discourse. Intent analysis is about extracting intentions, such as intent to complain or sell, from a piece of text.

Polarity is the type of opinion that is being expressed. Usually it is classified either as positive or negative. Sometimes, more granular division is used, for example accounting for neutral opinions, or a system based on a number of stars out of 5, etc.

In addition to polarity, SA systems are often concerned with identifying *opinion holder*, *opinion target*, or both. Opinion holder is the person or entity that expresses the opinion. Opinion target is the entity that towards which the opinion is directed. The entity can be inferred by the utterance and its context. For example:

(5) "*The only thing the **waiters** don't do for you is wipe your chin when you leave.*"

The review again evaluates the staff, but now we can extract the expression which refers to the target—"waiters". This is called *opinion target extraction* (OTE) and it is the focus of this thesis.

On the other hand, the target may not be explicitly mentioned in the text.

(6) "*And even with it's (sic!) Pub atmosphere they were great to my kids too!*"

The review evaluates the staff of a pub, but the staff is not explicitly mentioned. It is true that "they" represents the staff, but this is not specific enough if the goal is to search across millions of reviews.

1.2 Methods

In general, SA systems can be divided into 3 categories:

1. rule-based (e.g. Veselovská and Tamchyna [2014]),

2. automatic (e.g. Tamchyna and Veselovská [2016]) or

3. hybrid (e.g. Tamchyna et al. [2015]).

Rule based methods are usually based on a fixed set of rules. More specifically, they can based entirely on a lexicon. A lexicon is a list of subjective terms (usually) manually extracted from a large corpora, e.g. MPQA [Deng and Wiebe, 2015]. This lexicon is then used to determine the polarity of an utterance by some simple rule. Those systems have only limited access to the context which is crucial, therefore these systems are not suitable for applications that require high accuracy, as discussed by Veselovská and Hajič Jr. [2013].

In this thesis we focus on (fully) automatic systems. Unlike the rule-based systems, automatic systems rely on machine learning methods. In order to be used by machine learning algorithms, the raw data have to be pre-processed. This usually means tokenization and a subsequent numerical representation of the tokens, which also called feature extraction. We present the features that we are investigating in Chapter 2.

1.2.1 Word Embeddings

The simplest approach is to assign number the elements in some order and simply use those numbers. This approach has some obvious flaws, e.g. that the range of the feature is the same as the size of the vocabulary, which is typically large. Moreover, this does not preserve any linguistic relationships, words 2 and 3 may be in a similar relation as 42 and 16954, but this information is "lost forever". Therefore, more convoluted techniques were developed, such as bag-of-words and bag-of-n-grams that take into account frequency of words or n-grams respectively.

More recently, vector representation of words (tokens) as vectors have been widely used. The vectors are called *word vectors* or *word embeddings* and Mikolov et al. [2013c] showed that such vectors can be used to capture many linguistic regularities, e.g. the vector between singular and a plural form are similar across word pairs. Currently, the most popular methods are based on neural networks, such as continuous bag-of-words proposed by Mikolov et al. [2013a], skip-gram model by Mikolov et al. [2013b] and GloVe proposed by Pennington et al. [2014].

The advantage is that the embeddings can be trained once on a massive corpora and then used for other NLP tasks including SA. This is particularly useful, because the vectors are able to capture linguistic phenomena which may not be present in small datasets used for a specific task. This is a motivation for the features that we propose in Chaper 2, where we hope to make use of linguistic theories instead of just a lot of data. We use pre-trained word vectors were made available online.[1]

[1]For example at `https://nlp.stanford.edu/projects/glove/`

2. Syntactic Features

The purpose of this thesis is to explore how adding features based on how the words form phrases and sentences together. This is usually studied within syntax, therefore, we refer to the features as syntactic, although, strictly speaking, they may be considered to be part of morphology or semantics.

Most of the features that we propose are based on the notion of dependency grammar. Dependency grammars are based on the idea that linguistic units are connected by asymmetric binary links, i.e. the dependency relation. There are number of theories using this notion, e.g. functional generative description Functional generative description (FGD) by Sgall et al. [1969] or meaning-text theory (MTT) by Meľcuk et al. [1988].

2.1 Universal Dependencies

We decided to use Universal Dependencies[1] (UD) as the underlying dependency grammar, mainly due to its growing popularity with many freely accessible tools. The ultimate goal of UD is language parallelism, hence the name *universal*. UD is a framework is designed to provide a treebank annotation that is consistent across languages. It provides a universal inventory of categories to facilitate consistent annotation, but it also supports language specific features. In addition, UD is build with focus on suitability for computer processing with high accuracy, as well as comprehensibility.

UD is based on a lexicalist view, i.e. the relations hold between words. Because of that UD describes detailed guidelines for word segmentation and tokenization. In addition, it provides a specification of a morpho-syntactic representation consisting of a lemma, part-of-speech (see Section 2.1.2) and a set of features describing a set of grammatical properties such as tense, number, etc.

The syntactic annotation that UD scheme provides is based on typed dependency relations. The dependency relations form a tree (see Figure 2.1), where one word (usually a verb) is the head of the sentence. To ease computational processing, there is a notional node "root" (with ID 0).

Currently, UD uses 37 universal syntactic relations, which is based on the set of relations originally proposed by De Marneffe et al. [2014]. Primarily, the relations hold between content words, for example object is connected to the verb by *obj* relation. UD avoids mediating relations between content words by functional words, because those vary across languages. Function words are typically attached to content words by a direct dependency, for example "the" in the phrase "the cat" is connected to "cat" by *det* relation. Furthermore, copulas are considered to be auxiliary and are attached to a non-verbial predicate.

In addition to those *basic* dependecy relations, UD schema supports *enhanced* dependencies that do not form a tree, but a general graph, which is not necessarily a supergraph of the basic tree. Direct dependecies can be used for relation extraction, i.e. to determine the relation between two entities, e.g. two people. However, sometimes the path between the entities is too long in the UD tree.

[1] Complete anotation guidelines can be found at `https://universaldependencies.org`

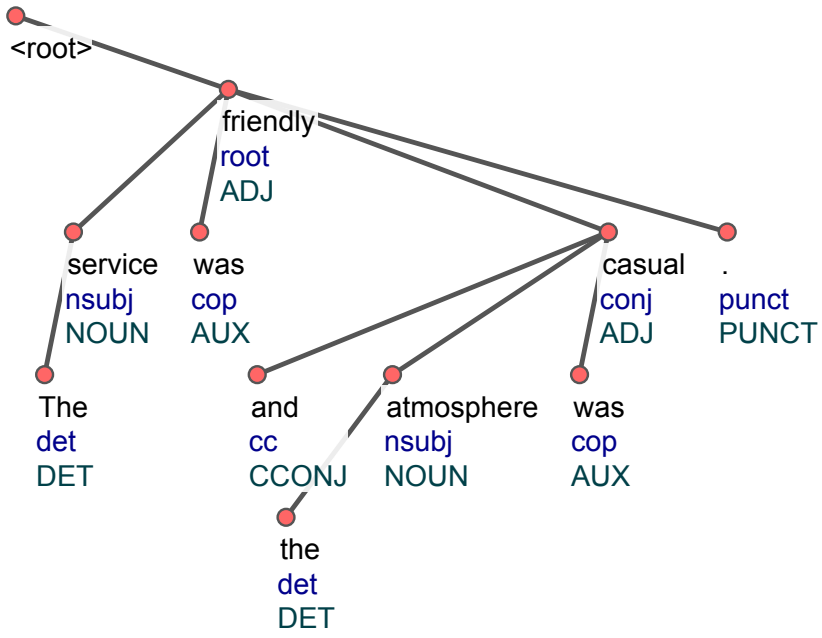


Figure 2.1: A dependency tree of a sentence from the data described Section 3.1. The dependency relation is displayed in lower case, the POS tag is in upper cases. The nominal subjects (*nsubj*) are two targets of this sentence.

Enhanced dependencies make some implicit relations between words more explicit. Unfortunately, since this part of annotation is not mandatory, many UD treebanks do not provide enhanced dependencies.

2.1.1 Syntactic Features

UD scheme provides syntactic annotation based on 37 different dependency relations (DEPREL, see Table 2.1) between words. Those relations define a tree structure (dependency tree), where an element with dependent elements is called the head, and the dependents are also called subordinate elements. Jakob and Gurevych [2010] used direct dependency to extract a binary feature indicating whether a word is dependent on an opinion expression. Veselovská [2017, p. 110] proposed hand-crafted syntactic rules to determine the opinion target. Those rules are language-independent to some extent, which is in line with the inter-language parallelism philosophy.

Zhuang et al. [2006] used direct dependencies to mine feature-opinion pairs from textual data. Kessler and Nicolov [2009] also used direct dependency to identify which opinion (sentiment) expressions are semantically related to the targets. They found out that frequent direct dependency path can be accurate connections between target and the opinion expression.

Therefore, we decided to use the **DEPREL** of every element (token) to its immediate head. However, distance in the tree may be longer. Motivated by the handcrafted rules of Veselovská [2017, p. 110] we also decided to provide the DEPREL of the immediate head to its own head, two steps up the tree. If the element is the head of the sentence (its head is the notional root), we use a special value "#ROOT#". We call this feature **HEAD DEPREL**.

	Nominals	Clauses	Modifier words	Function words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case

Coordination	conj	cc	
Multi-word expression	fixed	flat	compound
Loose	list	parataxis	
Special	orphan	goeswith	
Other	punct	root	dep

Table 2.1: Dependency relations used in UD schema.

2.1.2 Morphological Features

Part-of-speech Tag

Part o speech is a category of words that with similar grammatical properties, which means that words within one category behave similarly in terms of syntactic structure of the sentence. Therefore, providing a machine learning model with the part-of-speech could provide some additional information for infrequent words.

Universal Dependencies currently support 17 universal part-of-speech tags (see Table 2.2), abbreviated as UPOS. It is possible that some of those UPOS tags are not used for some languages, but for any language at most the set of 17 UPOS tags can be used. Language-specific part-of-speech tags (XPOS) are available in the UD schema.

We propose to use the **UPOS** tag a feature, as in the work of Jakob and Gurevych [2010]. The POS tags can provide some means for disambiguation. For example, "pllace" can be a noun or a verb.

(7) This **place** is amazing.

(8) It's difficult to place your order.

In Example 7, the noun "place" is the target of the evaluation, while in Example 8 the verb "place" is clearly not. In addition, we use the UPOS of the head element (**HEAD UPOS**), which may help to identify the role of the word in a phrase. We propose using **XPOS** and **FEATS** as features for every token as well.

	UPOS tag	Part of Speech
Open word class	ADJ	adjective
	ADV	adverb
	INTJ	interjection
	NOUN	noun
	PROPN	propper noun
	VERB	verb
Closed word class	ADP	adposition
	AUX	auxiliary
	CCONJ	coordinating conjunction
	DET	determiner
	NUM	numeral
	PART	particle
	PRON	pronoun
	SCONJ	subordinating conjunction
Other	PUNCT	punctuation
	SYM	symbol
	X	other

Table 2.2: Universal part of speech tags used in the Universal Dependency framework.

Lemma

Identifying the part-of-speech is one common task of morphological analysis. Another common task is providing a lemma of a word in some context, e.g. sentence or a phrase. Lemma is a canonical representation of a lexeme—a unit of lexical meaning. For example, words *cut*, *cuts* and *cutting* are all a part of the same lexeme, represented by a lemma "cut".

UDPipe provides a lemma for every token of analyzed piece of text. Supposing that a model has a word form and the POS tag, it should be able to "learn" the associated lemma, therefore adding the lemma does not seem to provide any extra information. However, the word form or the lemma of the head element could be useful. Since we are already providing the POS of the head, we decided to use the lemma of the head element as a feature.

2.1.3 UDPipe

In order to be used in real-world applications, the features extraction must be automated. Straka and Straková [2017] developed a trainable tool UDPipe. It is meant to be used for tokenization, tagging, lemmatization and dependency parsing. It is not dependent on any specific language and therefore can be trained on any data annotated using the UD schema.

Many fine-tuned pre-trained models are available online under CC BY-NC-SA licence.[2] UDPipe outputs analysis in the CoNLL-U format, which consist of the

[2] `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2998`

following

1. ID—word index starting at 1 for each new sentence,
2. FORM—surface form of the token,
3. LEMMA—lemma of the FORM,
4. UPOS—universal part-of-speech tag, see Table 2.2
5. XPOS—language-specific part-of-speech tag,
6. FEATS—a list of morphological features,
7. HEAD—ID of the head of the current word, 0 for the root,
8. DEPREL—universal dependency relation to the HEAD (or "root"),
9. DEPS—enhanced dependency graph, and
10. MISC—any other annotation

for every token.

We propose using FORM, UPOS, XPOS, DEPREL, FEATS as "raw" features, together with additional features based on the (dependency) head, described in the previous sections. Since the enhanced relations are not mandatory for UD treebanks, DEPS (and MISC) are often not supported by the available models trained on the treebanks; therefore, we refrained from using them.

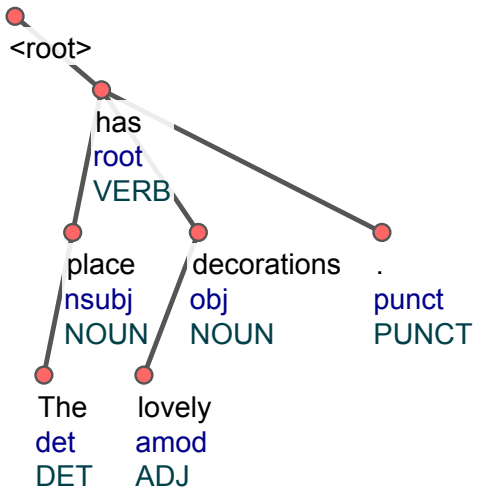


Figure 2.2: Decorations are the object of this sentence. Moreover, a sentimentally charged adjective is its dependent, which are clues that "decorations" is the target.

2.2 Subjectivity Lexicon

A sentence expressing an evaluating opinion typically has not only the target of the evaluation, but also some opinion (or sentiment) expression. These may be words typically associated with some emotion. For example:

(9) Amazing **food**.

Lexicons of such emotionally charged terms are available for many languages. However, this alone is not enough, because the word may or may not carry emotion depending on the context. For example:

(10) I like the **food**.

(11) It looks like food.

Syntax and morphology are important when we try to determine whether an opinion expression expresses an opinion towards an opinion target (see Figure 2.2). While in Example 10 the word "like" is used to express a positive emotion towards the food, while in Example 11 it is merely a function word. Note that the word has different POS in the two examples, which means that providing UPOS a feature may help mitigate those problems.

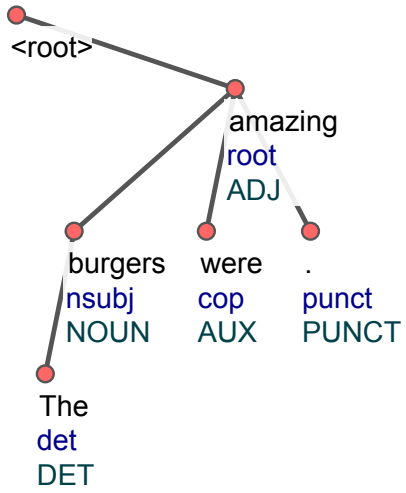


Figure 2.3: The head is an adjective that carries a sentiment, while the target is a subordinate element of the sentimentally charged adjective.

Jakob and Gurevych [2010] used a binary feature indicating whether a word has "a direct dependency relation to an opinion expression". However, they also note that using such feature requires some automatic identification of the opinion expression. For this purpose, corpora of subjective terms were collected for many languages. They were already for OTE used by Veselovská and Tamchyna [2014]. They assumed that words that are present in the subjectivity lexicon carry sentiment. This is obviously a simplifying assumption, however, one that is necessary if a accurate analysis of the sentence is not available.

Subsequently, Tamchyna et al. [2015] used a feature indicating whether a word is in a subjectivity lexicon for their OTE model. We decided to use the polarity of the word and its head. This can be useful because of the way how UD handles copula. Rather than being the head, the copula is a subordinate element and the adjective, which often carries the sentiment, is the head (see Figure 2.3) In a more complicated dependency tree, the opinionated expression may be in a relative clause, which is often attached to the same head as the subject (often the opinion target) as in Figure 2.4 Therefore, in addition to the two aforementioned features, we also propose to use a binary feature indicating whether any of the other elements subordinate to the same head ("siblings") carry sentiment.

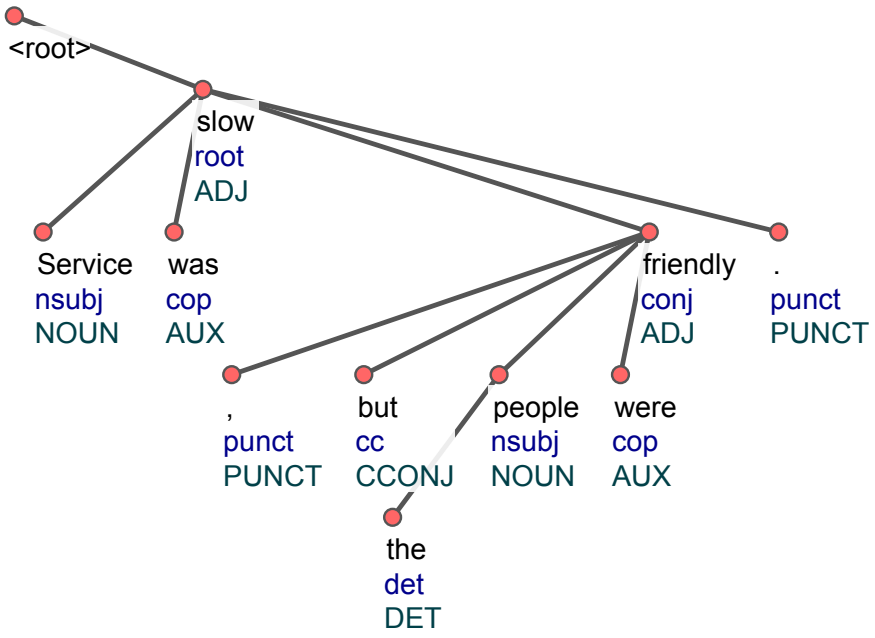


Figure 2.4: "Friendly" is a word associated with positive emotions. In this sentence, it is a sibling to an opinion target "service".

2.3 Feature Overview

In the previous sections, we described motivation for features based on morphological and syntactic analysis. We use UDPipe (Section 2.1.3) for to obtain such analysis automatically. Table 2.3

We propose using UPOS, DEPREL, FEATS and XPOS for every token. All of these are provided by UDPipe. In addition to that, we propose using UPOS, DEPREL and LEMMA of the (dependency) head of each token as a feature. For the head of the sentence, which has the notional root as its head, we use a special value "#ROOT#" for those features.

Feature	Short Description
word form	the surface form of the token
UPOS	universal POS of the token
DEPREL	dependency relation of the token to its head
XPOS	language specific POS of the token
FEATS	morphological features of the token
HEAD UPOS	UPOS of the token's head
HEAD DEPREL	dependency relation of the token's head to its head
HEAD LEMMA	the lemma of the head
SENT	sentiment of the token from a subjectivity lexicon
HEAD SENT	sentiment of the head
SIBLING SENT	binary feature indicating whether a sibling has sentiment

Table 2.3: An overview of proposed features.

To provide information about which words carry sentiment, we propose three

features based on a subjectivity lexicon. The feature called SENT, is simply the polarity of the word (or its lemma if the word form is not in the lexicon). HEAD SENT is simply the value of SENT of the dependency head, similarly to the features HEAD UPOS and HEAD DEPREL.

SIBLING SENT is a binary feature indicating whether any of the elements subordinate to the same head carry sentiment. This may be useful for more complex sentences with relative clauses, such as the one in Figure 2.4 etc.

3. Data

Customers tend to research products or services prior to purchasing them. Unsurprisingly, in the information age the most valuable resources are available on the Internet. Probably the most informative are experiences of other users, which are usually expressed as unstructured textual data. This is essential not only for the customers, but also for the providers, who receive feedback in this manner.

Each review contains subjective information about some entity (i.e. a product or a service). The goal of ABSA systems is to extract the entity and the polarity of the opinion towards the entity, which allows for automatic categorization of the feedback, identifying issues related to the products, etc. In particular, we focus on a dataset of restaurant reviews in English.

3.1 Data from Semeval 2016 Task 5

For the experiments, we use the data set used in the Task 5 of SemEval 2016. The full task description was given by Pontiki et al. [2016]. The data is an extended and corrected version of the dataset from Task 12 of SemEval 2015 [Pontiki et al., 2015], which was based on the dataset of Ganu et al. [2009]. Task 5 consisted of 3 sub-tasks, corresponding to two levels of ABSA (sentence and text levels) and a task with an unknown domain. The data is freely available (for non-commercial purposes) on SemEval's webpage[1].

We focus on the sentence level sub-task, which was divided into 3 slots: identification of aspect category, opinion target extraction (OTE) and polarity identification. First, the data were annotated by annotator A, "an experienced linguist". Then the data were investigated and corrected by annotator B, another "expert linguist". Unfortunately, Pontiki et al. [2016] does not specifies the inter-annotator agreement, which can serve as an upper limit on the performance of the models. The two human annotators were asked to identify the following:[2]

Aspect category: given by a pair of entities (e.g. food, service) and an attribute of the entity (e.g. quality, price) from a predetermined set of entities and aspects,

Opinion polarity: described as either positive, neutral or negative for each entity and aspect pair;

Opinion target expression: the explicit mention of the entity in the sentence.

If the target expression is stated implicitly, ergo inferred in the sentence, then the target expression is represented by a "NULL" value. For example:

(12) *"MMMMMMMMMmmmmmm so delicious"*

This review expresses a positive opinion towards the quality of the food, but the entity (food) is not explicitly mentioned in the sentence.

[1] `http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools`

[2] Detailed guidelines are available at `http://alt.qcri.org/semeval2016/task5/data/uploads/absa2016_annotationguidelines.pdf`

Sentences that contain opinions that cannot be described by the annotation schema are considered to be out of scope.

The training dataset consists of 2 000 sentences and the test dataset contains 676 sentences. Each sentence is represented as a single xml element. The surface form of the sentence is represented as a sub-element of the sentence element named text.

```
<sentence id="TR#1:3">
    <text>
        I highly recommend this beautiful place.
    </text>
    <Opinions>
        <Opinion target="place"
                 category="AMBIENCE#GENERAL"
                 polarity="positive"
                 from="34" to="39"/>
        <Opinion target="place"
                 category="RESTAURANT#GENERAL"
                 polarity="positive"
                 from="34" to="39"/>
    </Opinions>
</sentence>
```

Opinion targets are expressed by "opinion" elements, where its attribute "target" is the opinion target's expression, "category" is the entity and aspect pair; "polarity" is the polarity of the opinion towards the target. Finally, "to" and "from" attributes represent character offsets which uniquely identify the opinion target's expression. Only the first occurrence of the opinion target's expression is covered by the "from" and "to" offsets. However, the same opinion target expression can be assigned multiple categories. In this case the "opinion" element contains multiple "opinion" elements which differ only in their category attribute.

Using the UDPipe tokenizer and parser (see Section 2.1.3) we pre-process the data (see the script `semeval_data_preprocessing.py`, Appendix A.1). The average length of the training sequences is approximately 14 tokens where 1 is the minimum and 66 is the maximum length. The training data contains 24 536 tokens in total, 3 220 being unique. The average sequence length in the test set is approximately 14 tokens where 1 is the minimum and 78 is the maximum length. The training data contains 8 453 tokens in total, out of which 1 684 are unique (see Figure 3.1 and Figure 3.2).

The training dataset contains only 2 000 sentences. The scarcity of the data also affects the vocabulary size that can be extracted from the training data, which amplifies the impact of spelling mistakes.

Furthermore, 292 out of the training dataset are considered to be "out of scope", i.e. do not contain any annotation, because they do not fit the annotation guidelines of the task. This is detrimental when attempting to extract explicit mentions of targets, as it narrows the domain even further, thereby allowing only the targets fitting a particular schema. For example, the sentence

(13) *"There are many Thai places in the city but so far **Toons** is #1."*

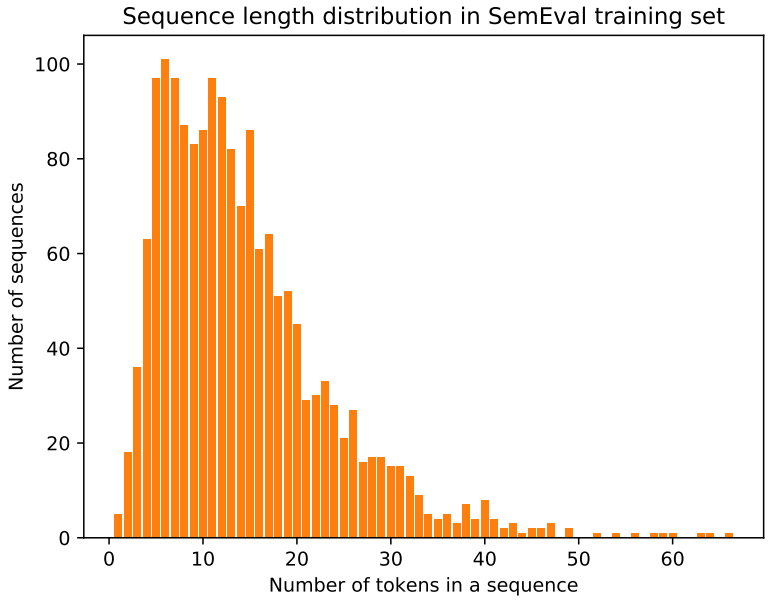


Figure 3.1: The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the training dataset of the restaurant reviews.

contains an explicit mention of the place (*Toons*) that is being evaluated. However, it is a comparative opinion, which does not fit the annotation guidelines, and therefore the target is not annotated.

We exclude the "out-of-scope" sentences because this shifts the task from locating the target of the expressed opinion to learning a particular annotation schema. Jebbara and Cimiano [2017], who worked on the same dataset and also used IOB tagging format, also selected this approach.

The training data contains 2 507 opinion targets and the test set contains 859 targets. However, those contain the same expressions under different entity aspect pairs and also the NULL targets that are not explicitly expressed in the text. The number of unique opinion target expressions, which we are trying to identify, is 1 744 and 616 for training and test data respectively.

In addition, if the targets are implicitly referred through pronouns, the pronouns themselves are not marked as the targets. Moreover, each target expression is captured only once in the data, which does not allow for a training of models that are intended for extraction of all mentions of an entity. This does not affect the objective of this thesis, but it makes the dataset unsuitable for more complex tasks.

3.2 Customer Reviews in Czech

To compare the results cross-linguistically we use a dataset of Czech reviews. This dataset contains product reviews and their fragments from a Czech e-shop with electronic devices. The data, together with the manual for annotation, are freely available at `http://hdl.handle.net/11234/1-1507`.

The dataset contains 1000 positive and 1000 negative short review segments. The segments roughly correspond to sentences, however, this is not guaranteed

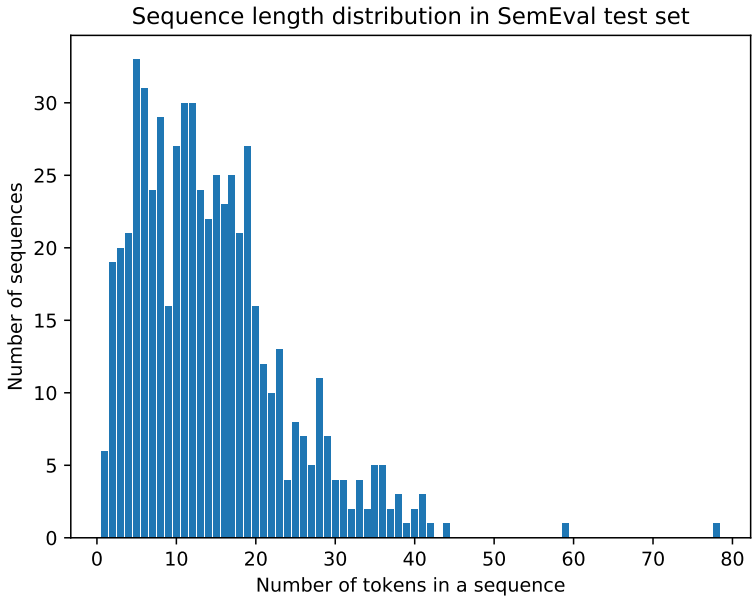


Figure 3.2: The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the test dataset of the restaurant reviews.

and often the segments are only nominal phrases. In addition it also contains 100 positive and 100 negative long reviews. However, we do not use those since the objective of this thesis is to conduct the experiments for sentence level OTE.

The data are structured into XML files. All reviews are accompanied with manually annotated targets of the evaluative phrases. If the target expression is explicitly mentioned in the review, it is enclosed in a `target` tag. The whole review or segment are enclosed in a `positive`$_s$`ummary` or a `negative`$_s$`ummary` tag, according to the polarity of the evaluative phrase.

A review in the dataset looks like this

```
<positive_summary id="1000000040">
    Nejlepší <target>podložka</target> pro práci
</positive_summary>
```

The data in the dataset are of varying length and complexity. Many of the reviews are brief descriptions of the aspects of the products, such as:

```
<positive_summary id="1000000099">
    <target>Kabel</target> je kabel.
</positive_summary>

<negative_summary id="1000000752">
    Žádné
</negative_summary>

<positive_summary id="1000000425">
    funguje :)
</positive_summary>
```

On the other hand, some review segments are quite complex and/or vague For example, the review segment

```
<positive_summary id="1000000471">
    Výborná věc pro vytváření domácí sítě.
    V nejhorším se dá použit i jako kladivo :)
</positive_summary>
```

does not contain any explicit mentions of the targets. It is almost impossible to determine that the review is in fact about crimping pliers from the segment alone.

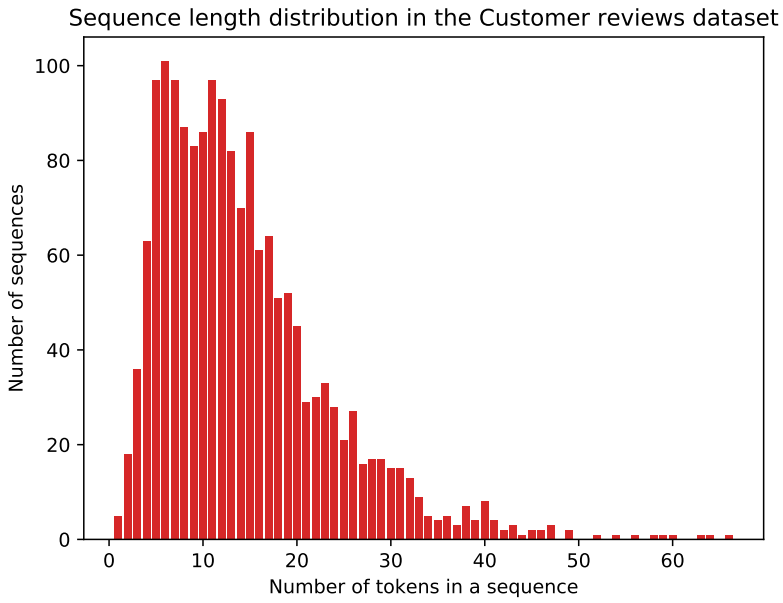


Figure 3.3: The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the Czech dataset of customer reviews.

Out of the 2 000 segments, only 870 contain at least one targets, and 2 contain only white space. The segments contain 1 148 targets in total. We use the script `lindat_data_preprocessing.py` (see Appendix A.1) to preprocess the data using UDPipe (see Section 2.1.3) we get a dataset where the average length of the sequence is approximately 7 tokens, minimum length is 1 and maximum is 122. The dataset contains 13 946 tokens in total, out of which 3 659 are unique. We split the dataset randomly into training and test data containing 1 598 and 400 sentences respectively (20%).

Furthermore, the data contain a lot of spelling mistakes, the segments are often a mixture of Czech and Slovak, sometimes even English. The reviews contain spelling errors, which may cause problems for tokenization and the morphosyntactic analysis. Some reviews do contain diacritical marks and some do not, which may further hinder the said analysis, however, it can be then easily turned into a consistent format.

The annotation guidelines for the Czech data are more vague than the ones for SemEval data (see Section 3.1) and therefore we cannot draw any conclusions on the performance of the models across different languages for this task. We can

only test whether similar trends can be observed after the data is enriched by the syntactic features.

3.3 Other Datasets

Many datasets for sentiment analysis are available online. Unfortunately, most of them are focusing on the polarity. Maas et al. [2011] created a popular dataset[3] of 50 000 IMDB reviews labelled with binary sentiment labels (positive or negative) and 50 000 unlabelled reviews. Pang and Lee [2005] created another dataset of movie reviews, which Socher et al. [2013] enhanced with parse trees and created Standford Sentiment Treebank with fully labeled parse trees.

Unlike for languages such as English or Spanish, up until the recent work of Veselovská [2017], there has been no systematic research of sentiment analysis in Czech. For Czech, Habernal et al. [2013] provided a dateset of social media posts with manually automated polarity and two other datasets where the polarity is based on the user ranking.

Nonetheless, the number of datasets with manually annotated opinion targets is scarce. In addition to the English dataset from Section 3.1, SemEval 2016 released datasets with restaurant reviews in Dutch, Russian, Spanish and Turkish. They also released an Arabic dataset for the domain of hotel reviews and a dateset that contains English reviews of electronic devices.

In addition to the Czech dataset from Section 3.2 there are a few other datasets that annotate opinion target expressions. Luo and Litman [2015] created a dataset of students responses to course evaluation where "summary phrases" are manually annotated in the text, and Luo et al. [2018] subsequently extended the annotation scheme for the same data. The dataset by Chathuranga et al. [2018] annotates the opinion target expression in student feedback directly with the IOB scheme.

Opinion target extraction is often viewed as a task in lexical semantics. For example, Wiegand et al. [2016] used German deWaC corpus to create a dataset of opinion compound that are possible targets. This is more specific than the general sentiment lexicons such as MPQA 3.0 subjectivity lexicon [Deng and Wiebe, 2015] or Czech Sublex 1.0 [Veselovská and Bojar, 2013].

[3] Available at `http://ai.stanford.edu/~amaas/data/sentiment/`

4. Related Work

In this chapter we discuss work published prior in the task of extracting the opinion target expressions. More specifically, we focus on experiments conducted on the datasets that we described in Chapter 3.

The objective of this thesis is not to select the best model, but explore how adding automatically generated morpho-syntactic analysis to the models influences their performance. Sequential models have proven to be successful for the tasks. Therefore we chose two specific discriminate models — conditional random fields (CRF) and neural networks with long short-term memory units (LSTM). We chose LSTM because it is widely used for a number of sequential tasks, and CRF because it is a non-neural sequential model, which provides a point of comparison for such small data.

4.1 Opinion Target Extraction

Opinion target identification aims to identify the entity towards which the opinion is expressed, not just the polarity of the opinion. This can be a high-level target (an object, e.g. "Murg Makhani" or "laptop") or an aspect of the object, e.g. spice in the food or a laptop component. We focus on previous work that aimed to extract the expression that represents the target.

OTE (a subtask of OTI) is a well-established task in NLP, similar to named entity recognition and other subtasks of information extraction. The target expression is either the object or its aspect that is evaluated in the opinionated text, therefore the task is sometimes also referred to as aspect term extraction (ATE). Each expression within a segment (phrase, sentence, paragraph, etc.) can have its own polarity, which may different than the sentiment of the segment. The target may not be explicitly expressed in the segment, which is especially true for pro-drop languages, such as Czech.

One of the main motivation for OTE is summarizing and grouping product reviews, therefore the first published works on OTE, e.g. by Hu and Liu [2004] or Popescu and Etzioni [2007], also call it "product feature extraction". In this thesis we are also concerned with features of restaurants, such as food and service quality, ambience, location, etc.

4.1.1 Lexical-Based Methods

The simplest method to extract the targets is to use a corpus of terms, such as the one by Wiegand et al. [2016]. However, such methods suffer greatly from false positives for frequently used terms, because they do not consider surrounding words. The presence of a term from a dictionary in itself is not enough—we would also expect to see some opinion indicating word such as "great" or "terrible". For example,

(14) This restaurant is absolutely amazing.

expresses an opinion (or rather an emotion) towards the restaurant, while the sentence

(15) The restaurant was open since 1965.

does not, hence it is not an opinion target expression. There are more convoluted way of constructing a lexical-based OTE extractor, e.g. Hu and Liu [2004] used Apriori algorithm to extract frequent noun phrases. Veselovská and Hajič Jr. [2013] provided a thorough error analysis of lexical-based classifiers for polarity detection, but some of the reasoning can be applied to OTE as well.

To mitigate the issues of lexical-based methods, Veselovská and Tamchyna [2014] used hand-crafted rules based on a syntactic analysis provided by a parser and a subjectivity lexicon. In addition to aspect term extraction, they also identified aspect term polarity, aspect category detection and aspect category polarity.

4.1.2 Machine Learning Methods

Machine learning methods were successfully employed for OTE. The tasks has been viewed as a labeling task, or a token classification problem. If task is viewed as a token classification problem, support vector machines were successfully used by Manek et al. [2017].

If the task is viewed to be a labeling task, sequential models are usually used. One of the most popular methods is a sequential model CRF, e.g. used by Chernyshevich [2014], Toh and Wang [2014], Tamchyna et al. [2015], Hamdan et al. [2015] and Jakob and Gurevych [2010]. Another commonly used methods is a sequential neural model with LSTM or GRU (gated recurrrent unit) units used by Liu et al. [2015], Jebbara and Cimiano [2017].

Moreover, Huang et al. [2015] combined the two into a single approach and Laddha and Mukherjee [2019] used it specifically for OTE. Wang et al. [2016] proposed Recursive Neural Conditional Random Fields (RNCRF), which consists of two components—a recurrent neural network (RNN) and a CRF layer. Where the RNN component is based on the dependency trees of the training sentences, which is intended to learn a high-level representation for each word in a sentence. The CRF component is supposed to capture the context around each word.

4.1.3 Syntactic Features

Jakob and Gurevych [2010] used a CRF model with the surface form of the tokens and the their POS tags together with two features based on a dependency parse. The first (binary) feature is to label all tokens that are dependent on an "opinion expression". The latter is a (binary) feature that labels all tokens in the nearest noun phrase to an "opinion expression". The use of these two features improved F-measure, while recall more than precision (see Table 4.1). However, those features rely on the "opinion expressions" that were manually annotated.

Ding et al. [2017] used a model similar to our model described in Section 5.4 to incorporate rules similar to the rules proposed by [Veselovská, 2017, p. 110]. They used the rules to produce auxiliary labels that are learned and are also predicted by a hidden layer in the network and then combined with the "ordinary" labels.

The sequential models clearly have access to the context of the potential candidates for an opinionated expression, so they should be less likely to produce the same kind of false positive as lexical-based methods, although they can still

Domain	**Precision**	**Recall**	**F-measure**
movies	0.79	0.48	0.60
movies + features	0.64	0.13	0.22
web-services	0.62	0.35	0.45
web-services + features	0.50	0.05	0.10
cars	0.60	0.39	0.47
cars + features	0.44	0.11	0.18
cameras	0.60	0.43	0.50
cameras + features	0.30	0.09	0.13

Table 4.1: Precision, recall and F-measure for a version of dataset with only the tokens and POS tags and a version enhanced by the two syntactic features [Jakob and Gurevych, 2010].

occur. This thesis aims to investigate whether features from the dependency parser can further mitigate this problem.

4.2 English Dataset of Restaurant Reviews

The dataset described in Section 3.1 served as a material for many experiments in sentiment analysis. The dataset was originally used for Task 5 of SemEval 2015, with 3 different slots (see Section 3.1). However, we are only concerned with Slot 2 of the task, which is the OTE.

SemEval 2016 used precission, recall and F-meassure as metrics to compare performance of the submitted models. The metrics were "calculated by comparing the list of the targets that a system returned (for a sentence) to the corresponding gold list"[1]. The NULL targets were discarded because "they do not correspond to explicit target mentions".

Table 4.2 shows the performance of the 3 best models submitted to SemEval 2016 on the english dataset of restaurant reviews. In addition it includes the official baseline, which is based on the dictionary extracted from the training data. Pontiki et al. [2016, p. 25, Table 3.] lists the F-measure[2] of all models on the test data from the dataset.

Model	**Precision**	**Recall**	**F-measure**	Citation
1. NLANG.	0.75	0.69	0.72	Toh and Su [2016]
2. AUEB-.	0.72	0.69	0.70	Xenos et al. [2016]
3. UWB	0.75	0.61	0.67	Hercig et al. [2016]
Baseline	0.51	0.39	0.44	Pontiki et al. [2016]

Table 4.2: Performance of the three best ranked models submitted to SemEval 2016 and the official baseline.

[1]From the "Evaluation-Validation-Submission-Baselines" available at `http://alt.qcri.org/SemEval2016/task5/index.php?id=data-and-tools`

[2]The values of precision and recall are cited from the paper in the column citation.

Aside from the SemEval 2016 submittions, there have been several experiments using the provided datasets. Chen et al. [2017] used a model consisting of a bidirectional long short-term memory (BiLSMT) layer stacked together with a conditional random field (CRF) layer and reported F-measure of 0.7244.

Jebbara and Cimiano [2017] used a RNN model based on GRU, similar to the model used in this thesis. They reported improving the F-measure from 0.6260 to 0.6586 by using character-level features to improve the performance.

In addition to the English dataset, experiments have been conducted for the datasets in other languages. For example, Al-Smadi et al. [2019] used syntactic and other features to improve the performance on the Arabic dataset of hotels reviews.

4.3 Czech Dataset of Customer Reviews

The Czech dataset of customer reviews contains annotation only for polarity and the target expressions in the reviews or review segments. Therefore, it is naturally suitable for OTE. In fact the dataset was introduced by Tamchyna et al. [2015] for this task specifically.

Tamchyna et al. [2015] used conditional random fields (CRF, see Section 5.3) using manually designed rules as a feature, together with morpho-syntactic features and subjectivity lexicon. They observed that adding those features improves recall, but lowers precision, however, the F-measure still improves for the (short) segments (see Table 4.3).

Features	**Precision**	**Recall**	**F-measure**
surface	0.85	0.37	0.51
+morpho-syntactic	0.76	0.54	0.63
+sublex	0.78	0.55	0.65
+rules	0.77	0.58	0.66

Table 4.3: Precision, recall and f-measure obtained using various feature sets for the segments of reviews [Tamchyna et al., 2015].

This dataset was also used by Glončák [2016] to reproduce the experiment of Tamchyna and Veselovská [2016] on the Czech data. However, this is experiment aimed to identify the aspect category, not OTE.

5. Methods

5.1 Data Preprocessing

For the experiments we used two datasets of reviews, one in English and Czech. We used the English data (on the restaurant domain) for the Subtask 2 of Taks 5 of SemEval 2016 [Pontiki et al., 2016]. The description of the dataset is provided in Section 3.1. The Czech dataset of customer reviews of electronics is described in Section 3.2

We used the available data to produce a sequence of OBI labels, that represent the target expression. To ensure consistency, we use UDPipe [Straka and Straková, 2017] tokenize the data only once, before all the experiments.

IOB scheme is frequently used for the task, e.g. by Jakob and Gurevych [2010] and Tamchyna et al. [2015]. "O" represents tokens that are outside of the target expression, tokens that are at the beginning of the target expression are labeled as "B" and all other tokens that are inside the expression are denoted by "I".

The benefit of using the "B" label is that it allows to separate targets that are adjacent to each other. However, this does happen in the English dataset only once for the sentence:

(16) "*We concluded with* ***tiramisu chocolate cake****, both were delicious.*"

In this sentence, "tiramisu" and "chocolate cake" are two separate entities. The UDPipe tokenizer also creates this problem for the sentence:

(17) "*Poor* ***customer service****/poor* ***pizza****.*"

This is because it fails to correctly split "*service/poor*" into two tokens. Arguably, those are an anomaly and therefore we also perform the experiments with a simpler OT scheme. We use OT scheme to denote whether a token is outside ("O") of a target, or a target ("T"). This is often referred to to as OI or IO scheme, but we decided to use "T" instead of inside to avoid any confusion. However, in the Czech dataset which consists mostly of just keywords this happens much more often and therefore we decided to strictly adhere to IOB for the Czech dataset.

After the sentences are split into tokens, each token is assigned an "O", "B" or "I" label based on all the start and end offsets for all target expressions in the respective sentence. The start offset is the number of characters from the beginning of the sentence that are not included in the target. All tokens that begin within the range determined by the two offsets is labeled as "B" if it is the first token in the opinion target expression or "I" if it is any consequent token in the expression.

Sentences that do not contain any target expressions (i.e. out of scope sentences) are ignored. This leaves us with 1708 training sentences and 587 test sentences.

In addition, we use UDPipe to extract syntactic and morphological features described in Chapter 2.

The surface word forms whose lemma occurs only once in the data are replaced by a special out-of-vocabulary (OOV) token. This allows the models to make generalizations about rare words.

The targets in the data do not overlap, which justifies the use of the IOB scheme, with only one exception. The whole sentence[1]

(18) "– *This is one of my top lunch spots, huge portions, fast service and amazing margaritas!!*"

is labeled as a "NULL" target. However, this does not fit the annotation guidelines, the "NULL" targets are not supposed to be explicitly represented in the sentence. Therefore, we ignore this target, as well as the rest of the "NULL" targets, because they are not explicitly expressed in the textual data.

5.2 Baseline

As the baseline for our experiments, we use a simple lexical-based model, similar to the ones described in Section 4.1.1. Our **Baseline-dictionary** only labels the exact phrases extracted from the training data. The phrases are selected so that more specific phrases (the longest possible phrase containing the tokens) are identified, i. e. "wine list" is preferred to just "wine". The shortcomings of this approach are thoroughly analyzed in the work of Veselovská and Hajič Jr. [2013].

5.3 CRF

Lafferty et al. [2001] proposed conditional random fields (CRF) for sequence modeling. It is a discriminative probabilistic graphical model. For observations $\mathbf{X}$ and random variables $\mathbf{Y}$, CRF is a graph where each vertex v is a representative for some Y_v. Conditioned on $\mathbf{X}$, $\mathbf{Y}_v$ obey the Markov property with respect to the graph, i.e.

$$p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, v \neq w) \approx p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, v \text{ adjacent to } w).$$

The conditional distribution $p(\mathbf{Y}|\mathbf{X})$ is modeled, therefore the model is called discriminative. The model is similar to the (generative) hidden Markov model.

The important detail is that the probability distribution is conditioned to all the observations. Therefore, if CRF are used to label a sentence, at each node, the decision can be based on any part of the sentence. This allows for the use of context, unlike the dictionary model from Section 5.2. However, this may require meticulous feature engineering.

CRF have been used for various NLP tasks in the past, such as POS tagging, named entity recognition, and speech recognition. For OTE specifically, CRF were utilized for example by Li et al. [2010], Jakob and Gurevych [2010] and Tamchyna et al. [2015].

First, we need to train the CRF, i.e. estimate the conditional probabilities $p(\mathbf{X}|\mathbf{Y})$. This can be done using iterative gradient descent algorithms, or Quasi-Newton methods such as the L-BFGS algorithm.

However, we are interested in finding the most likely sequence of labels, rather than determining probability of a given sequence. Fortunately, this can be done efficiently with the Vitterbi algorithm.

[1] Sentence id `en_MercedesRestaurant_478010602:1`

5.4 Neural Networks

Neural networks have been successfully used for various NLP tasks, including the subtasks of sentiment analysis. The winning submission to SemEval 2015 by Toh and Su [2015] used convolution networks for aspect category identification.

Long sentences often contain words that are in a direct dependency but far away from each other in the surface form of the sentence. Recurrent networks are able to "remember" information throughout a long sequence by feeding its output back to its input. However, repeating this process many times leads to the gradients used to train the network's weights become too small to be useful. Hochreiter and Schmidhuber [1997] proposed long short-term memory units to address the vanishing gradient issue.

LSTM mitigate this problem by using three trainable gates—input, output and forget gate. Gated recurrent units (GRU) proposed by Cho et al. [2014] are similar to LSTM with only the forget gate, which means that they have fewer parameters; thus they require fewer resources to train while often providing comparable performance.

In order to allow the network to use both, past and future information (relative to the current position in the sequence), bidirectional recurrent neural networks were proposed by Schuster and Paliwal [1997]. The bidirectional layer has two hidden states—one for reading the input forward and another one for backward reading. We decided to use such bidirectional version of LSTM, abbreviated as BiLSTM.

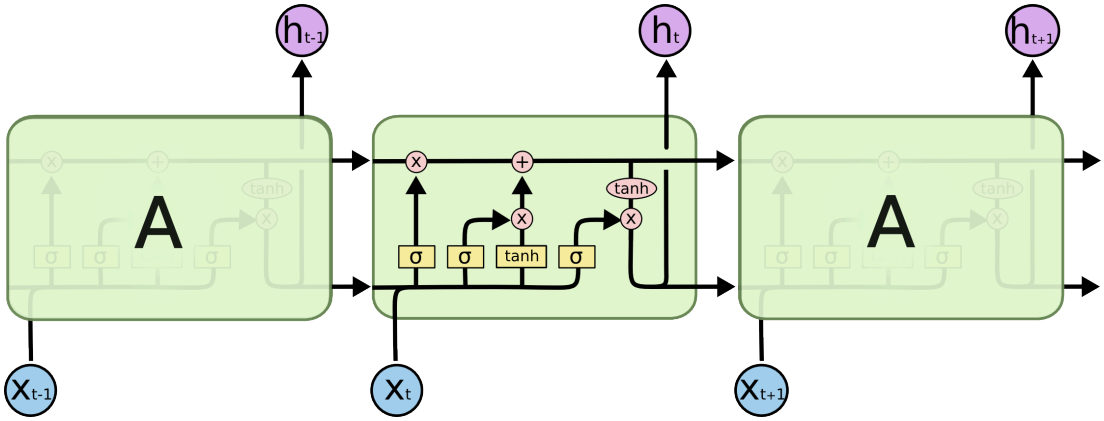


Figure 5.1: A scheme of a single LSTM unit. The gates are denoted by σ.[2]

LSTM were particularly successful in handwriting recognition and a lot of other tasks that require learning long-distance relationships. Tamchyna and Veselovská [2016] used a network with LSTM units to identify the aspect category because "*syntactic relationships and long-distance dependencies may play a significant role and that such phenomena may be better modeled with a recurrent network*". We select a similar model for the same reason. The model of Tamchyna and Veselovská [2016] was ranked the best out of all the models submitted to SemEval 2016 for the task for Russian and Turkish dataset.

The model that we use is based on the model used by Liu et al. [2015] for OTE. They found that the model with LSTM outperforms feature-rich models

[2]Taken from `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

based on CRF. Ding et al. [2017] proposed two extensions to the model of Liu et al. [2015] for OTE based on syntactic rules (see Section 4.1.3). To address the issue of over-fitting and training divergence, we used dropout between the embedding and the recurrent layer. Dropout, as proposed by Srivastava et al. [2014], is a technique in which only subset of nodes is considered in each training step. The subset is chosen randomly at each training step.

The model based on Liu et al. [2015], which we denote as **LSTM-1** consists of an embedding layer, which is used to transform the vocabulary into word vectors (see Section 1.2.1). The output of the embedding layer is then fed to a BiLSTM layer. Subsequently, the output of the recurrent layer is fed to a densely connected hidden layer which contains one neuron for each possible label. Then the output is passed to softmax function

$$\sigma(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

to select the most likely label.

Another model that we use is **LSTM-2**. It is similar to the LSTM-1 model, the difference is that it uses two recurrent layers. The output of the (first) hidden BiLSTM layer is fed into the second hidden BiLSTM layer. This is inspired by the model of Tamchyna and Veselovská [2016].

Training

Recurrent networks often suffer from exploding gradients problem, detailed by Bengio et al. [1994]. To address this issue, we use gradient clipping [Pascanu et al., 2013].

In addition to that, we use a decreasing learning rate scheduler. The scheduler has a fixed number of times the learning rate is decreased. The learning rate is decreased by a constant factor if at least some (fixed constant) improvement to the loss has not been achieved in some given number of epochs. This may help to get closer to a local minimum that the optimization may be oscillating around. However, if the value of the loss function is improving, the learning rate is not affected and therefore this technique is unlikely to hurt the performance of a model.

5.5 Evaluation Metrics

To compare the performance of our models we use precision, recall and F-measure for the three types of labels—"B", "I" and "O", where we focus mainly on the values for "B" and "I", as those represent the targets. This is evaluated on the token level.

It is also possible to evaluate precision, recall and F-measure on the target level, i.e. consider not only tokens, but the whole target expression as the instance. This is what the the SemEval's evaluation tool does.

For the data described in Section 3.1, we also used the provided tool to measure precision, recall and F-measure as they were measured in the original task of SemEval 2016. The values measured with this tool are always measured on the test data, and they are only meant to provide a point of comparison to the models in the original competition.

In addition to this, we define two "generous" metrics. A metric that we call *permissive recall* is the percentage of targets in which at least one token was labelled by "B" or "I". Similarly, we *permissive precision* to be a metric, in which is calculated by the formula

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}.$$

However, in permissive precision, any predicted sequence of labels that are not "O" is considered to be a true positive, if it has any overlap with a sequence of target labels in the gold data.

Having a dedicated test set is important for the purposes of a competition. However, as the data are already small, we decided to use both, training and testing dataset together to perform 5-fold cross-validation to obtain more robust values of the metrics defined above.

5.5.1 Analysis of the Results

The goal of this thesis is to establish whether the features based on syntactic relations, described in Chapter 2, can be used to improve the performance of the models.

To analyze the results, we compare pairs of predictions—one produced by a model that has access to the additional syntactic features and one that does not. We do this by producing the difference of the confusion matrices for the basic model and the model with syntactic features for every feature. We also compare individual examples of missclassification that occurred on the test data set.

In addition, we would like to establish whether the models produce different kinds of errors. Since we would like to compare pairs of observations (i.e. one produced by the model with syntactic features) it seems that we would like to use a paired test for this purpose.

However, there is yet no consensus on how to use statistical test for comparing classifiers, because multiple issues arise. Firstly, the number of observations that we have is quite small, which increases the probability of the error of the second type—not rejecting the null hypothesis that the models are producing predictions with the same distribution (of errors).

More importantly, most tests assume independence of the observations, in this case the predictions of a model or an evaluation metric computed from those. It is not clear when two models can be assumed to be independent. For a similar reason, cross-validation cannot be used (without producing some probability of error) to obtain more observations as their are not independent.

Dietterich [1998] recommends using McNemar's test for the models that can be trained only once, for example large neural models that take days to learn. McNemar's test [McNemar, 1947] is a test of marginal homogeneity, i.e. the null hypothesis states that the marginal probabilities are the same. This is trivially true if the models produce the same prediction. On the other hand, if two models produce significantly different results, we would expect to reject the null hypothesis. McNemar's test was defined for 2×2 contingency tables, which means that it is suitable only for binary classification, but the test has been extended to accommodate for multiple categories, notably by Cochran [1950].

6. Experiments

In this chapter, we provide an outline of the experiments we performed, following Chapter 7 that lists the results of the experiments. **Our focus is not necessarily to find the model that has the best performance, but to observe how the syntactic features that we proposed in Chapter 2 can influence the performance of the models.**

First we perform a number of experiments using cross-validation on the whole dataset of restaurant reviews from Section 3.1. Based on those results we select one model to evaluate against the test data provided for SemEval 2016 and analyze the behavior of the model depending on whether the syntactic features are used.

Finally, we provide the results of some of the models on the Czech dataset (Section 3.2) to provide a point of comparison across languages.

6.1 Baseline Model

We implement a trivial model to obtain a basic point of comparison. The model extract words and phrases from the training data. As we discussed in Section 3.1, only the first mention of the entity is labeled. Therefore, the model labels the first occurrence of any phrase from the vocabulary extracted from the training data.

6.2 CRF Models

We decided to use the L-BFGS method, because it has been used by Li et al. [2010] for the same task. After performing a few experiments with a randomly selected validation set, we chose to use 200 iterations for the CRF models, using 0 for L1 regularization and 0.01 for L2 regularization, and we allowed transitions that are not observed in data.

We use python-crfsuite, a python binding to CRFsuite software to train a CRF model on the data from Chapter 3. We decided to use the previous and the following tokens as the features, inspired by Tamchyna et al. [2015] who used 2 previous and 2 following tokens (in addition they also extracted all bigrams and trigrams).

In addition to the word form of a current token, we provide the CRF with the previous and the following word. This is our **CRF-basic** model. **CRF-syntax** is a model that in addition to the aforementioned features uses the syntactic features that we proposed in Chapter 2 for every token.

6.3 Models with LSTM

We chose 3 variations of the LSTM neural model:

1. **LSTM-1**—a model with a single hidden LSTM layer,
2. **LSTM-emb**—a LSTM-1 model with pre-trained word vectors,

3. **LSTM-2**—a model with two LSTM layers.

All of these models have a **basic** version that only uses the surface form of the tokens and a **syntax** version, using the features proposed in Chapter 2 in addition to the surface form. We use TensorFlow 1.12.0 [Abadi et al., 2015] to implement the models.

We use a GPU device[1] to perform the experiments with the neural models as they are computationally expensive but can be significantly sped up with (massive) parallelism provided by the GPU devices. To allow for more efficient processing, all the sequences should be of equal length. Based on the lengths of sentences (in tokens) reported in Chapter 3, we chose the length of the sequences to be 100. Sequences that are longer are trimmed from the end to 100 tokens.

We run several experiments evaluating the results on a small randomly selected validation set. From the observation of the performance on the validation set, we decided to use 64 LSTM units on the first layer for all the models, and 32 on the second layer for LSTM-2 model. Those numbers (powers of 2) are selected with the performance of GPU units in mind.

We chose the size of the embeddings of the features to be as follows

Feature	**Embedding Size**
word form	100
UPOS	2
DEPREL	8
XPOS	2
FEATS	8
HEAD UPOS	2
HEAD DEPREL	8
HEAD LEMMA	100
SENT	2
HEAD SENT	2
SIBLING SENT	1

Table 6.1: The sizes of the embeddings for the features of the neural models.

Similarly, we chose to train the models for 80 epochs. For the model with 2 layers, we use 200 training epochs. This is based on the performance on a validation set for both datasets. However, this model is computationally expensive, and therefore we refrain from using it. All the other parameters can be found in the attached configuration files.

[1]We used GeForce GTX 1050 for our experiments.

7. Results

In this chapter, we provide the values of performance measures of the models for groups of features. The models name stands for the model that has been trained only on the word forms. The abbreviation *+ syntax* stands for features based on the UD analysis, that are not based on the (dependency) head, namely UPOS, DEPREL, XPOS, and FEATS. The abbreviation *+ head* means that features based on the head, i.e. HEAD UPOS and HEAD DEPREL, have been used in addition to *syntax*. The abbreviation *+ sentiment* means that the features based on the subjectivity lexicon have been added to the previous two sets of features, namely SENT, HEAD SENT and SIBLING SENT.

7.1 Baseline model

The trivial model using phrase dictionary achieved permissive precision of 0.24 and permissive recall of 0.76.

	Precision	**Recall**	**F-measure**
B	0.17	0.68	0.28
I	0.65	0.13	0.22

Table 7.1: Cross-validated precision, recall and F-measure for "B" label for the CRF models on the English dataset.

7.2 English Dataset of Restaurant Reviews

7.2.1 CRF models

Table 7.2 shows the precision, recall and F-measure for the "B" label, Table 7.3 shows the same metrics for the "I" label. We omit the measures for the "O" as the information about miss-classification of this label is already included in the precision of the other two labels. Table 7.4 shows the values of the permissive metrics described in Section 5.5. Those metrics depict whether a target was identified at all, not necessarily with the right boundaries.

	B Precision	**B Recall**	**B F-measure**
CRF	0.74	0.64	0.69
CRF + syntax	0.73	0.63	0.67
CRF + head	0.74	0.64	0.69
CRF + sentiment	0.73	0.65	0.69
CRF + head's lemma	0.72	0.63	0.67

Table 7.2: Cross-validated precision, recall and F-measure for "B" label for the CRF models on the English dataset.

	I Precision	I Recall	I F-measure
CRF	0.65	0.41	0.50
CRF + syntax	0.61	0.43	0.50
CRF + head	0.60	0.43	0.50
CRF + sentiment	0.63	0.45	0.52
CRF + head's lemma	0.62	0.42	0.50

Table 7.3: Cross-validated precision, recall and F-measure for "I" label for the CRF models on the English dataset.

	Permissive Precision	Permissive Recall
CRF	0.80	0.70
CRF + syntax	0.79	0.69
CRF + head	0.80	0.71
CRF + sentiment	0.79	0.71
CRF + head's lemma	0.79	0.70

Table 7.4: Cross-validated permissive precision, recall for the CRF models on the English dataset.

Binary Labels

In addition to using the IOB scheme, we also try to use binary labels, as explained in Section 5.1. Table 7.6 shows the permissive metrics for the binary labels. The results are almost the same as the ones for IOB labels, displayed in Table 7.4. Table 7.5 shows the precision, recall and F-measure for the label "T" indicating a target. The values for "T" label are higher, however, "T" represents both "B" and "I". Therefore, based on the permissive metrics, which does not show significant difference in performance, in the rest of the experiments we stick to the IOB scheme.

	T Precision	T Recall	T F-measure
CRF	0.78	0.61	0.68
CRF + syntax	0.76	0.60	0.67
CRF + head	0.75	0.59	0.66
CRF + sentiment	0.76	0.63	0.68
CRF + head's lemma	0.75	0.54	0.63

Table 7.5: Cross-validated precision, recall and F-measure for "T" label for the CRF models on the English dataset.

Data Filtered on Lemma

The CRF model does not exhibit any significant difference on the English dataset for the added features. We replace the surface forms whose lemmas occur only once with a special out-of-vocabulary (OOV) token. In this scenario we are purposely hiding the information about some tokens from one model, while providing

	Permissive Precision	Permissive Recall
CRF	0.79	0.71
CRF + syntax	0.79	0.69
CRF + head	0.79	0.70
CRF + sentiment	0.79	0.72
CRF + head's lemma	0.78	0.64

Table 7.6: Cross-validated permissive precision, recall for the CRF models on the English dataset, using only binary labels.

at least some information to the other. This is disadvantages the model that has only access to the word forms (especially on such a small dataset) but it tells us whether the features can be used to infer the necessary information at all.

Now we can observe some difference in the precision, recall and F-measure for the "B" (Table 7.7) and "I" label (Table 7.8). The permissive precision and recall (Table 7.9) show a similar trend to the precision and recall for the labels—while precision tends to decrease, recall tends to increase with the added features.

	B Precision	B Recall	B F-measure
CRF + OOV	0.72	0.48	0.58
CRF + OOV + syntax	0.71	0.59	0.64
CRF + OOV + head	0.70	0.60	0.64
CRF + OOV + sentiment	0.72	0.62	0.66
CRF + OOV + head's lemma	0.70	0.61	0.65

Table 7.7: Cross-validated precision, recall and F-measure for "B" label for the CRF models on the filtered English dataset.

	I Precision	I Recall	I F-measure
CRF + OOV	0.61	0.26	0.36
CRF + OOV + syntax	0.60	0.43	0.50
CRF + OOV + head	0.57	0.43	0.49
CRF + OOV + sentiment	0.57	0.44	0.50
CRF + OOV + head's lemma	0.61	0.41	0.49

Table 7.8: Cross-validated precision, recall and F-measure for "I" label for the CRF models on the filtered English dataset.

7.2.2 LSTM

First we perform the same experiment as for the CRF model with the unfiltered data. The metrics for "B" and "I" are displayed in Table 7.10 and Table 7.11 respectively, the permissive metrics are in Table 7.12.

	Perm. Precision	Perm. Recall
CRF + OOV	0.80	0.54
CRF + OOV + syntax	0.76	0.64
CRF + OOV + head	0.75	0.65
CRF + OOV + sentiment	0.77	0.68
CRF + OOV + head's lemma	0.77	0.68

Table 7.9: Cross-validated permissive precision, recall for the CRF models on the filtered English dataset.

	B Precision	B Recall	B F-measure
LSTM-1	0.67	0.69	0.68
LSTM-1 + syntax	0.67	0.69	0.68
LSTM-1 + head	0.68	0.69	0.69
LSTM-1 + sentiment	0.67	0.69	0.68
LSTM-1 + head's lemma	0.65	0.69	0.67

Table 7.10: Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the English dataset.

Filtered Data

Our first neural model LSTM-1 with a single layer shows similar trends to the ones observed for the CRF model on the filtered data. Table 7.13 and Table 7.14 show the values of precision, recall and F-measure for "B" and "I" labels respectively, Table 7.15 shows the values permissive precision and recall for the model on the filtered data.

As Table 7.14 shows, the recall for the "I" label increased significantly. However, this is not surprising, since the model does not have acces to the surface form of lemmas of words that occur only once, i.e. adding the lemma of the head is a significant advantage.

7.2.3 Overview of Models

Based on the results mentioned above, we selected the set of features to be used to be the *+sentiment* feature set, i.e. all the features except for the lemma of the head. Although some results may suggest that the lemma of the head may improve performance, it is computationally expensive, especially for a neural model with pre-trained word vectors. In addition to the reported results, we tried to change the parameters of the model, such as the number of LSTM units etc. but we did not manage to achieve any significantly different results.

7.3 Performance on the SemEval Test Data

We train the LSTM-emb model with and without the syntactic features on the training set provided for SemEval 2016 (see Section 3.1). We meassure the performance of the two model on the provided test set.

	I Precision	I Recall	I F-measure
LSTM-1	0.55	0.43	0.48
LSTM-1 + syntax	0.55	0.41	0.47
LSTM-1 + head	0.54	0.44	0.49
LSTM-1 + sentiment	0.55	0.43	0.48
LSTM-1 + head's lemma	0.52	0.47	0.49

Table 7.11: Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the English dataset.

	Perm. Precision	Perm. Recall
LSTM-1	0.72	0.77
LSTM-1 + syntax	0.74	0.77
LSTM-1 + head	0.74	0.77
LSTM-1 + sentiment	0.73	0.77
LSTM-1 + head's lemma	0.71	0.78

Table 7.12: Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset.

Table 7.19 shows the metrics that we described in Section 5.5 and used in the previous sections to compare the models. We can see that the values are roughly the same for both models. We can observe a slight trend similar to the previous results, i.e. the precision goes decreases and recall increases. However, such small differences are not sufficient to draw any conclusions.

The SemEval evaluation tool evaluates the F-measure to be 0.64 and 0.65 for for the basic and the model with syntactic features respectively. The precision is 0.61 and 0.59, recall 0.68 and 0.68, both for the basic model and the model with syntactic features respectively. Both outperform the baseline used for the task at SemEval 2016 (see Table 4.2).

McNemar's test gives a p-value of approximately 0.47. Therefore, we cannot reject the null hypothesis that the model without and with syntactic features produce different kinds of errors.

To explore how the individual features may have influenced the predictions, we used graphs that we call confusion graph (see Figure 7.1). It is a difference of of the confusion matrix of the model with syntactic features and the model without syntactic features. Therefore, positive numbers on the diagonal indicate that an error has been fixed, while outside of the diagonal it is indicated by negative numbers.

To make the graph more intuitive, we display all the matrix elements that represent an improvement in green and the rest in red. The hue is proportional to the percentage of the changes for the specific value of the feature. For confusion graphs for all the features and their values see Attachment A.2–A.10. Feature values for which no change occurs are not displayed. There does not seem to be any noticeable pattern in the amount of errors that were corrected.

There are some sentences, where the model with syntactic features produced a better result. However, these are only anecdotal, and we cannot use them to

	B Prec.	B Recall	B F-measure
LSTM-1 + OOV	0.71	0.46	0.56
LSTM-1 + OOV + syntax	0.71	0.47	0.57
LSTM-1 + OOV + head	0.70	0.47	0.56
LSTM-1 + OOV + sentiment	0.68	0.50	0.57
LSTM-1 + OOV + head's lemma	0.68	0.50	0.57

Table 7.13: Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the English dataset.

	I Precision	I Recall	I F-measure
LSTM-1 + OOV	0.52	0.17	0.25
LSTM-1 + OOV + syntax	0.56	0.20	0.29
LSTM-1 + OOV + head	0.57	0.21	0.31
LSTM-1 + OOV + sentiment	0.55	0.20	0.29
LSTM-1 + OOV + head's lemma	0.48	0.33	0.39

Table 7.14: Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the English dataset.

make the claim that the model generally improved the predcition as it introduces new errors. The model with syntactic features managed to identify some of the targets that the model without the features did not, for example:

(19) "*As usual the **omikase** didn't disappoint in freshness, although it scored low on creativity and selection.*"

(20) "*The **coffe** (sic!) is very good, too.*"

Example 20 is interesting since the target contains a spelling mistake.

On the other hand, it did not find some that the model without the features managed to detect, such as:

(21) "*Everything, and I mean everything on the **menu** is delectable.*"

Sometimes the model identified a whole phrase, that was only partially selected by the model without the features, such as Examples 22 and 23 respectively.

(22) "*One of the best **Sushi place** in town.*"

(23) "*One of the best Sushi **place** in town.*"

In Examples24 the model identified only a part of the target "Creme Brulee", however, it is still much better than the prediction of the model without the syntactic features (Example 25).

(24) "*The appetizer was interesting, but the **Creme** Brulee was very savory and delicious.*"

(25) "*The appetizer was interesting, but the Creme Brulee was very **savory** and delicious.*"

	Perm. Precision	Perm. Recall
LSTM-1 + OOV	0.78	0.55
LSTM-1 + OOV + syntax	0.78	0.55
LSTM-1 + OOV + head	0.77	0.55
LSTM-1 + OOV + sentiment	0.78	0.56
LSTM-1 + OOV + head's lemma	0.74	0.60

Table 7.15: Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset.

	B Prec.	B Recall	B F-measure
CRF	0.74	0.64	0.69
CRF + sentiment	0.73	0.65	0.69
LSTM-1	0.67	0.69	0.68
LSTM-1 + sentiment	0.67	0.69	0.68
LSTM-emb	0.71	0.71	0.71
LSTM-emb + sentiment	0.71	0.71	0.71
LSTM-2	0.63	0.70	0.66
LSTM-2 + sentiment	0.66	0.67	0.66

Table 7.16: Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the English dataset.

7.4 Czech Dataset of Customer Reviews

To provide a point of comparison, we also perform some of the experiments on the Czech dataset. We did not run the model with the pre-trained embeddings for the Czech data as we did not find any available embeddings of the same size (such that they would satisfy our device restrictions).

7.4.1 CRF models

We perform the same experiments as in the Section 7.2, except for the LSTM-emb model, due to the memory restriction of the hardware as we did not find pre-trained word vectors for Czech that would fit into the memory of the devices that we have available.

The results for the basic CRF model are even more discouraging than the equivalent results for the English dataset. Table 7.20 and Table 7.21 display the metrics for the "B" and "I" label, Table 7.22 shows the values of the permissive metrics.

7.4.2 Overview of Models

To provide a comparison, we performed the same experiments as in the Section 7.2.3. Table 7.20 and Table 7.21 show precision, recall and F-measure for "B" and "I" labels respectively. Table 7.22 displays the permissive metrics.

	I Precision	I Recall	I F-measure
CRF	0.65	0.41	0.50
CRF + sentiment	0.63	0.45	0.52
LSTM-1	0.55	0.43	0.48
LSTM-1 + sentiment	0.55	0.43	0.48
LSTM-emb	0.66	0.46	0.54
LSTM-emb + sentiment	0.66	0.48	0.56
LSTM-2	0.50	0.50	0.50
LSTM-2 + sentiment	0.48	0.50	0.49

Table 7.17: Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the English dataset.

	Perm. Precision	Perm. Recall
CRF	0.80	0.70
CRF + sentiment	0.79	0.71
LSTM-1	0.72	0.77
LSTM-1 + sentiment	0.73	0.77
LSTM-emb	0.75	0.81
LSTM-emb + sentiment	0.78	0.80
LSTM-2	0.70	0.79
LSTM-2 + sentiment	0.70	0.79

Table 7.18: Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset.

	B Precision	B Recall	B F-measure
LSTM-1	0.68	0.67	0.67
LSTM-1 + sentiment	0.66	0.67	0.67

	I Precision	I Recall	I F-measure
LSTM-1	0.58	0.42	0.48
LSTM-1 + sentiment	0.56	0.44	0.49

	Perm. Precision	Perm. Recall
LSTM-1	0.73	0.77
LSTM-1 + sentiment	0.72	0.79

Table 7.19: The performance measures for the

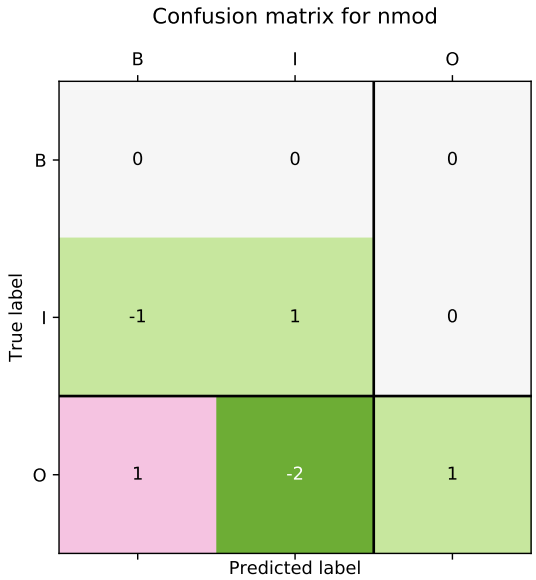


Figure 7.1: A confusion graph for "nmod" value of the DEPREL feature, obtained by evaluating the model on the SemEval test data.

	B Precision	**B Recall**	**B F-measure**
CRF	0.78	0.60	0.68
CRF + syntax	0.72	0.60	0.65
CRF + head	0.71	0.59	0.64
CRF + sentiment	0.70	0.59	0.64
CRF + head's lemma	0.74	0.57	0.65

Table 7.20: Cross-validated precision, recall and F-measure for "B" label for the CRF models on the Czech dataset.

	I Precision	**I Recall**	**I F-measure**
CRF	0.45	0.22	0.29
CRF + syntax	0.38	0.20	0.26
CRF + head features	0.33	0.20	0.25
CRF + sentiment features	0.36	0.20	0.26
CRF + head's lemma	0.40	0.16	0.22

Table 7.21: Cross-validated precision, recall and F-measure for "I" label for the CRF models on the Czech dataset.

	Permissive Precision	Permissive Recall
CRF	0.80	0.62
CRF + syntax	0.73	0.63
CRF + head	0.72	0.62
CRF + sentiment	0.72	0.62
CRF + head's lemma	0.76	0.60

Table 7.22: Cross-validated permissive precision, recall for the CRF models on the Czech dataset.

	B Prec.	B Recall	B F-measure
CRF	0.78	0.60	0.68
CRF + sentiment	0.70	0.59	0.64
LSTM-1	0.77	0.61	0.68
LSTM-1 + sentiment	0.78	0.61	0.68
LSTM-emb	—	—	—
LSTM-emb + sentiment	—	—	—
LSTM-2	0.64	0.63	0.63
LSTM-2 + sentiment	0.66	0.64	0.65

Table 7.23: Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the Czech dataset.

	I Precision	I Recall	I F-measure
CRF	0.45	0.22	0.29
CRF + sentiment	0.36	0.20	0.26
LSTM-1	0.39	0.12	0.18
LSTM-1 + sentiment	0.50	0.17	0.26
LSTM-emb	—	—	—
LSTM-emb + sentiment	—	—	—
LSTM-2	0.31	0.16	0.19
LSTM-2 + sentiment	0.25	0.05	0.08

Table 7.24: Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the Czech dataset.

	Perm. Precision	**Perm. Recall**
CRF	0.80	0.62
CRF + sentiment	0.72	0.62
LSTM-1	0.79	0.63
LSTM-1 + sentiment	0.79	0.63
LSTM-emb	—	—
LSTM-emb + sentiment	—	—
LSTM-2	0.67	0.65
LSTM-2 + sentiment	0.68	0.66

Table 7.25: Cross-validated permissive precision, recall for the LSTM-1 models on the Czech dataset.

Conclusion

In this thesis we aimed to investigate whether an automatic syntactic analysis can be used to improve performance of the models used for opinion target extraction. Opinion target extraction is a subtask of sentiment analysis that aspires to extract the entity that is being evaluated in an opinionated piece of text. We did not pursue the goal of finding the best fine-tuned model that would achieve the best performance. Rather, we use the models to determine if the features can be used in OTE.

For the experiments, we proposed 10 features based on the annotation schema of Universal Dependencies. Based on the experiments, we decide to use 9 of them as the feature set, namely UPOS, DEPREL, XPOS, FEATS provided directly by the tool for automatic UD analysis. We also used UPOS and DEPREL of the dependency head of each word (token). In addition to that we propose 3 features based on a subjectivity lexicon, two of which are also based on the dependency tree.

We used two kinds of model to conduct the experiments—CRF and a neural model with LSTM units. We proposed 3 versions of the LSTM model. We selected those two models specifically because they were both successfully employed in OTE. Both are sequential models—CRF is a non-neural discriminative model, whereas LSTM is a recurrent neural model.

For the experiments, we used the dataset designed specifically for an OTE task at SemEval 2016. We selected two versions of the LSTM model with pre-trained embeddings—one with the aforementioned features and one without—to compare it to the submissions to SemEval 2016. Our model outperformed the baseline of the competition. We also used a Czech dataset designed for the same task to see whether the effects of the features are the same for other languages too. Since Czech is a morphologically rich language, it is a suitable candidate for such comparison.

Nevertheless, our results are inconclusive. We did not observe any pattern of improvement that would appear when the features have been added to a model. The changes in the performance are generally small and can go either direction.

We managed to use the features to improve the performance of the CRF model if surface forms of unique words have been masked by a special OOV value. This could be useful in cases where the surface form contains personal information and has to be removed from the raw test, but the results of morpho-syntactic analysis can be stored. However, this improvement did not translate well to our LSTM models, which may suggest that the recurrent neural models are already capable of learning some sort of dependency relations.

In some cases, the model with syntactic features managed to provide a better prediction.

(26) "*It?s (sic!) served with either a* ***peppercorn sauce*** *or* ***red wine reduction****, though both were indistinguishable in taste.*"

E.g. in Example26 the model with the features managed to correctly identify "peppercorn sauce" as opposed to just "sauce", which is what the model without the features selected. However, in the same sentence, the model with the features

picked **red wine** as a target, unlike the model without the features that managed to pick the complete target.

Although in some cases the feature might solve a particular error, it also introduces some new errors. This is also dependent on the quality of the analysis provided by the UD analyzer. We used a version of McNemar's test to determine whether the two models perform different kinds of errors, but we failed to reject the null hypothesis that they do not.

There is no substantial basis to claim that the proposed features improve performance of the models in general. In some specific cases they may be useful, but further investigation is necessary to support such claim.

The number of models that we used in this thesis is quite small, as the focus of the thesis is not to search for the best model. Every model that we proposed can be fine-tuned. Other more complex models could be used to boost the performance as well.

In addition to fine-tuning the models, it would be interesting to compare results of the models and the syntactic features across different domains and languages. As we discussed in Chapter 3, the number of datasets for this task is modest. To further the investigation, it would be suitable to create new manually annotated datasets. Another issue is that the quality of the features depends on the quality of the syntactic analysis. Although we did not find convincing evidence that the proposed features are useful in general, there may be some specific configuration in which they may help.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yaser Jararweh, and Omar Qawasmeh. Enhancing aspect-based sentiment analysis of arabic hotels' reviews using morphological, syntactic and semantic features. *Information Processing & Management*, 56(2):308–319, 2019.

Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Janaka Chathuranga, Shanika Ediriweera, Ravindu Hasantha, Pranidhith Munasinghe, and Surangika Ranathunga. Annotating opinions and opinion targets in student course feedback. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.

Maryna Chernyshevich. Ihs r&d belarus: Cross-domain extraction of product features using crf. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 309–313, 2014.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

William G Cochran. The comparison of percentages in matched samples. *Biometrika*, 37(3/4):256–266, 1950.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–92, 2014.

Lingjia Deng and Janyce Wiebe. Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, pages 1323–1328, 2015.

Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

Ying Ding, Jianfei Yu, and Jing Jiang. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer, 2009.

Vladan Glončák. Artificial neural network for opinion target identification in czech. Bachelor thesis, Charles University in Prague, Prague, Czech Republic, 2016.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. Sentiment analysis in czech social media using supervised machine learning. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 65–74, 2013.

Hussam Hamdan, Patrice Bellot, and Frederic Bechet. Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 753–758, 2015.

Tomáš Hercig, Tomáš Brychcín, Lukáš Svoboda, and Michal Konkol. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 342–349, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045. Association for Computational Linguistics, 2010.

Soufian Jebbara and Philipp Cimiano. Improving opinion-target extraction with character-level word embeddings. *CoRR*, abs/1709.06317, 2017. URL `http://arxiv.org/abs/1709.06317`.

Jason S Kessler and Nicolas Nicolov. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

Abhishek Laddha and Arjun Mukherjee. Aspect specific opinion expression extraction using attention based lstm-crf network. *arXiv preprint arXiv:1902.02709*, 2019.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics, 2010.

Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, 2015.

Wencan Luo and Diane Litman. Summarizing student responses to reflection prompts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, 2015.

Wencan Luo, Fei Liu, and Diane J. Litman. An improved phrase-based approach to annotating and summarizing student course responses. *CoRR*, abs/1805.10396, 2018. URL `http://arxiv.org/abs/1805.10396`.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

Asha S Manek, P Deepa Shenoy, M Chandra Mohan, and KR Venugopal. Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier. *World wide web*, 20(2):135–154, 2017.

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

Igor Aleksandrovic Meľcuk et al. *Dependency syntax: theory and practice*. SUNY press, 1988.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 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, 2013c.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, 2015.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30, 2016.

Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Petr Sgall, Ladislav Nebeský, Alla Goralčíková, and Eva Hajičová. *A functional approach to syntax in generative description of language*. American Elsevier New York, 1969.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/K/K17/K17-3009.pdf`.

Aleš Tamchyna and Kateřina Veselovská. Ufal at semeval-2016 task 5: recurrent neural networks for sentence classification. In *Proceedings of the 10th international workshop on semantic evaluation (SEMEVAL-2016)*, pages 367–371, 2016.

Ales Tamchyna, Ondrej Fiala, and Katerina Veselovská. Czech aspect-based sentiment analysis: A new dataset and preliminary results. In *ITAT*, pages 95–99, 2015.

Zhiqiang Toh and Jian Su. Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 496–501, Denver, Colorado, June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S15-2083`.

Zhiqiang Toh and Jian Su. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 282–288, 2016.

Zhiqiang Toh and Wenting Wang. Dlirec: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 235–240, 2014.

Kateřina Veselovská and Ondřej Bojar. Czech SubLex 1.0, 2013. URL `http://hdl.handle.net/11858/00-097C-0000-0022-FF60-B`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kateřina Veselovská and Jan Hajič Jr. Why words alone are not enough: Error analysis of lexicon-based polarity classifier for czech. In *Proceedings of the 3rd Workshop on Sentiment Analysis where AI meets Psychology*, pages 1–5, 2013.

Kateřina Veselovská and Aleš Tamchyna. Úfal: Using hand-crafted rules in aspect based sentiment analysis on parsed data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 694–698, 2014.

Kateřina Veselovská. *Sentiment analysis in Czech*, volume 16 of *Studies in Computational and Theoretical Linguistics*. ÚFAL, Praha, Czechia, 2017. ISBN 978-80-88132-03-5.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679*, 2016.

Michael Wiegand, Christine Bocionek, and Josef Ruppenhofer. Opinion holder and target extraction on opinion compounds–a linguistic approach. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 800–810, 2016.

Dionysios Xenos, Panagiotis Theodorakakos, John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Aueb-absa at semeval-2016 task 5: Ensembles of classifiers and embeddings for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 312–317, 2016.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.

List of Figures

2.1 A dependency tree of a sentence from the data described Section 3.1. The dependency relation is displayed in lower case, the POS tag is in upper cases. The nominal subjects (*nsubj*) are two targets of this sentence. 9
2.2 Decorations are the object of this sentence. Moreover, a sentimentally charged adjective is its dependent, which are clues that "decorations" is the target. 12
2.3 The head is an adjective that carries a sentiment, while the target is a subordinate element of the sentimentally charged adjective. . 13
2.4 "Friendly" is a word associated with positive emotions. In this sentence, it is a sibling to an opinion target "service". 14

3.1 The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the training dataset of the restaurant reviews. 18
3.2 The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the test dataset of the restaurant reviews. 19
3.3 The distribution of the length of sequences (in number of tokens after tokenization with UDPipe) for the Czech dataset of customer reviews. 20

5.1 A scheme of a single LSTM unit. The gates are denoted by σ.[1] . . 28

7.1 A confusion graph for "nmod" value of the DEPREL feature, obtained by evaluating the model on the SemEval test data. 41

List of Tables

2.1 Dependency relations used in UD schema. 10
2.2 Universal part of speech tags used in the Universal Dependency framework. 11
2.3 An overview of proposed features. 14

4.1 Precision, recall and F-measure for a version of dataset with only the tokens and POS tags and a version enhanced by the two syntactic features [Jakob and Gurevych, 2010]. 24
4.2 Performance of the three best ranked models submitted to SemEval 2016 and the official baseline. 24
4.3 Precision, recall and f-measure obtained using various feature sets for the segments of reviews [Tamchyna et al., 2015]. 25

6.1 The sizes of the embeddings for the features of the neural models. 32

7.1 Cross-validated precision, recall and F-measure for "B" label for the CRF models on the English dataset. 33
7.2 Cross-validated precision, recall and F-measure for "B" label for the CRF models on the English dataset. 33
7.3 Cross-validated precision, recall and F-measure for "I" label for the CRF models on the English dataset. 34
7.4 Cross-validated permissive precision, recall for the CRF models on the English dataset. 34
7.5 Cross-validated precision, recall and F-measure for "T" label for the CRF models on the English dataset. 34
7.6 Cross-validated permissive precision, recall for the CRF models on the English dataset, using only binary labels. 35
7.7 Cross-validated precision, recall and F-measure for "B" label for the CRF models on the filtered English dataset. 35
7.8 Cross-validated precision, recall and F-measure for "I" label for the CRF models on the filtered English dataset. 35
7.9 Cross-validated permissive precision, recall for the CRF models on the filtered English dataset. 36
7.10 Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the English dataset. 36
7.11 Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the English dataset. 37
7.12 Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset. 37
7.13 Cross-validated precision, recall and F-measure for "B" label for the LSTM-1 models on the English dataset. 38
7.14 Cross-validated precision, recall and F-measure for "I" label for the LSTM-1 models on the English dataset. 38
7.15 Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset. 39

7.16 Cross-validated precision, recall and F-measure for “B” label for the LSTM-1 models on the English dataset. 39
7.17 Cross-validated precision, recall and F-measure for “I” label for the LSTM-1 models on the English dataset. 40
7.18 Cross-validated permissive precision, recall for the LSTM-1 models on the English dataset. 40
7.19 The performance measures for the 40
7.20 Cross-validated precision, recall and F-measure for “B” label for the CRF models on the Czech dataset. 41
7.21 Cross-validated precision, recall and F-measure for “I” label for the CRF models on the Czech dataset. 41
7.22 Cross-validated permissive precision, recall for the CRF models on the Czech dataset. 42
7.23 Cross-validated precision, recall and F-measure for “B” label for the LSTM-1 models on the Czech dataset. 42
7.24 Cross-validated precision, recall and F-measure for “I” label for the LSTM-1 models on the Czech dataset. 42
7.25 Cross-validated permissive precision, recall for the LSTM-1 models on the Czech dataset. 43

List of Abbreviations

ABSA aspect based sentiment analysis

BiLSTM bidirectional long short-term memory

CRF conditional random fields

DEPREL dependency relation

LSTM long short-term memory

MPQA Multi-Perspective Question Answering

NLP natural language processing

OTE opinion target expression

OOV out-of-vocabulary

SA sentiment analysis

UD Universal Dependencies

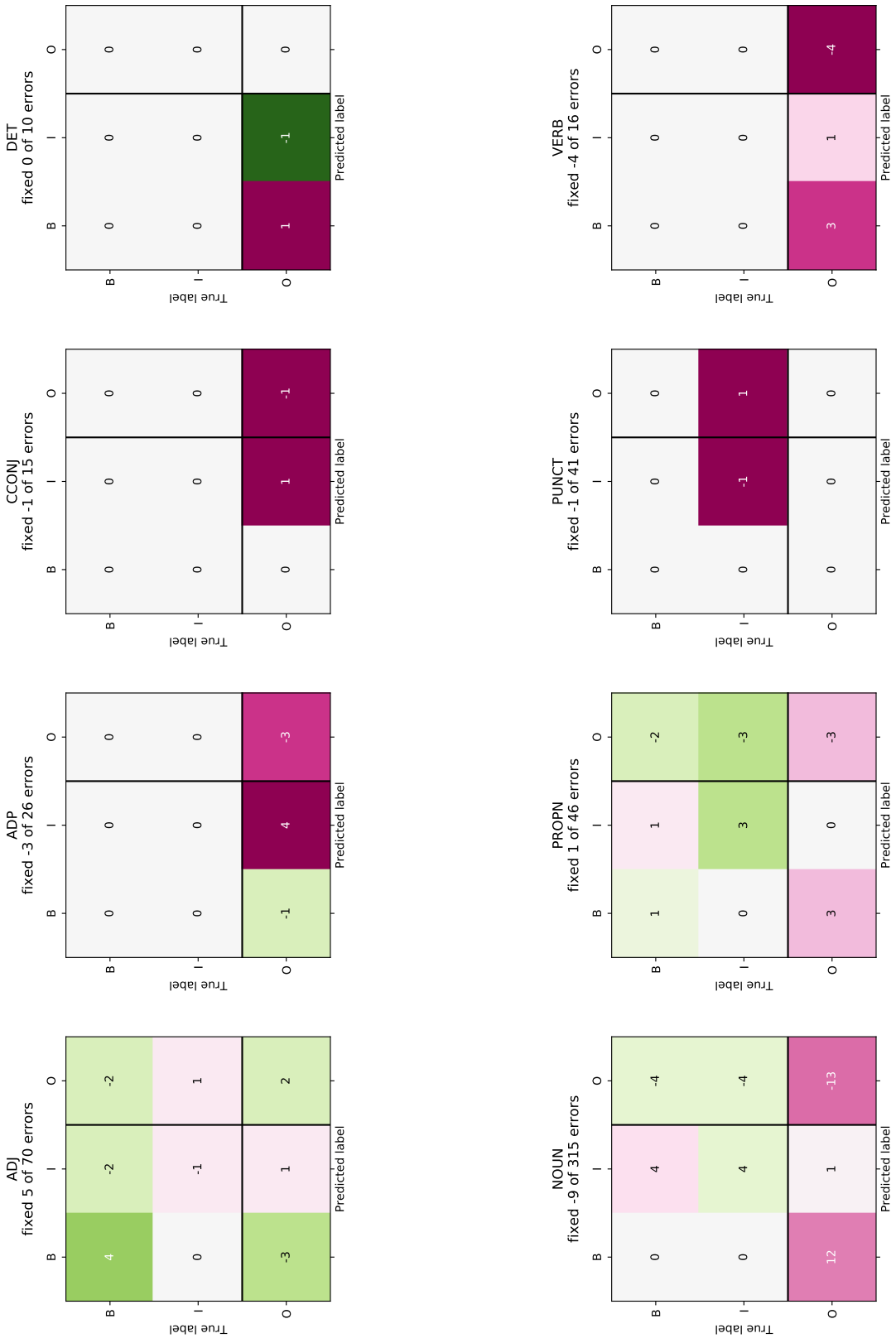
UPOS universal part-of-speech

A. Attachments

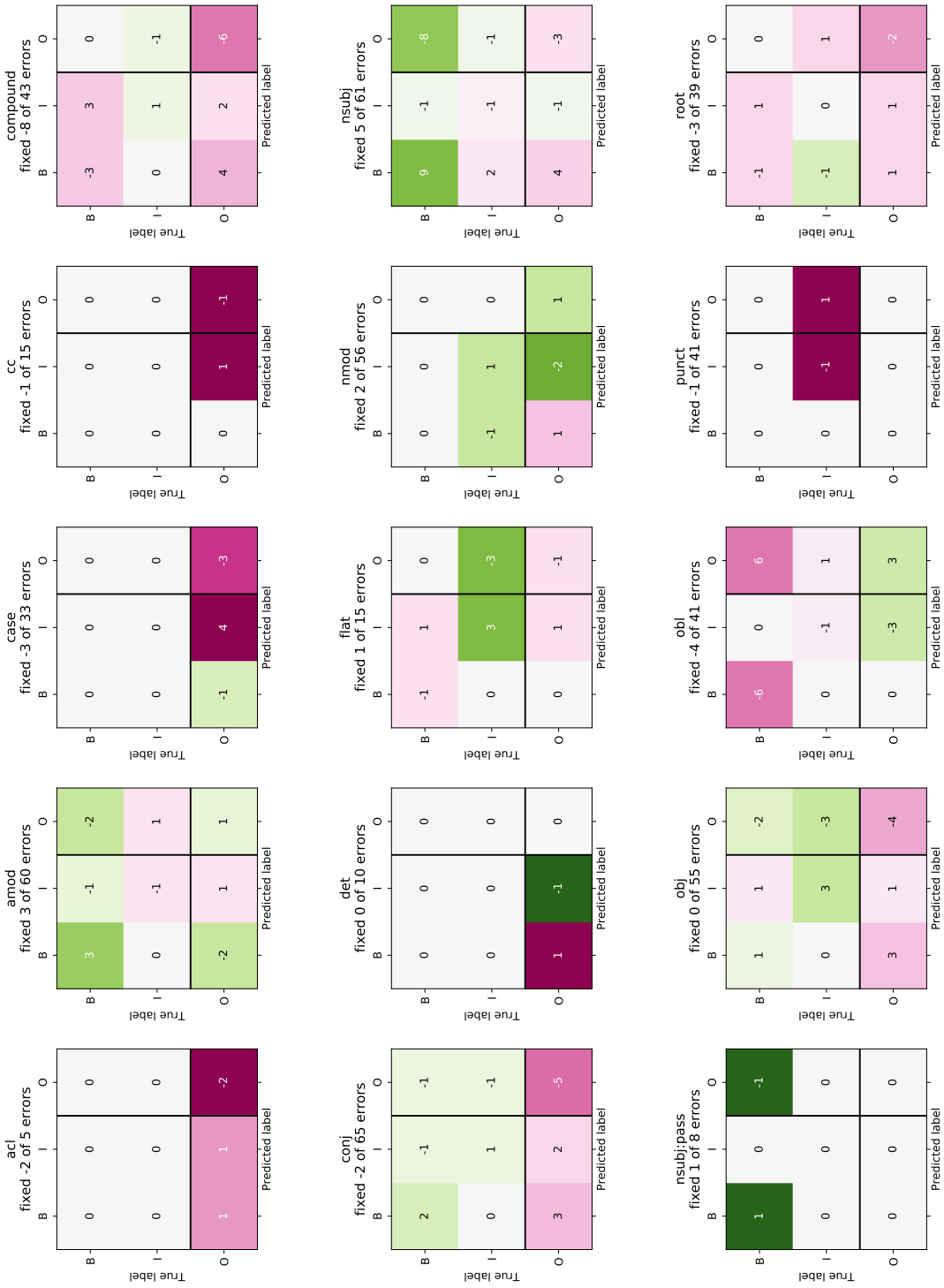
A.1 Source Code

The electronic attachment to the thesis is the source code used for the experiments.

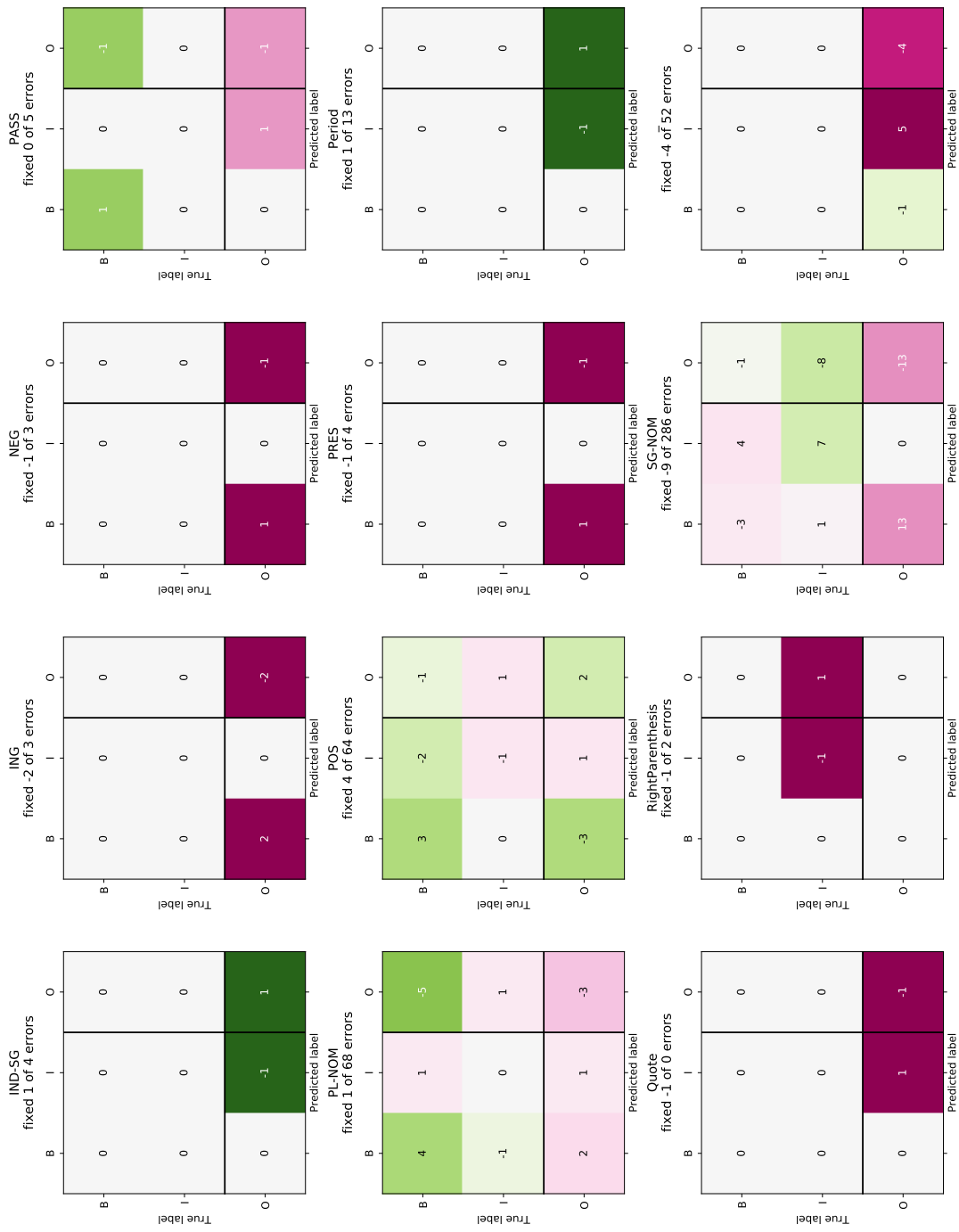
A.2 Confusion Tables for UPOS



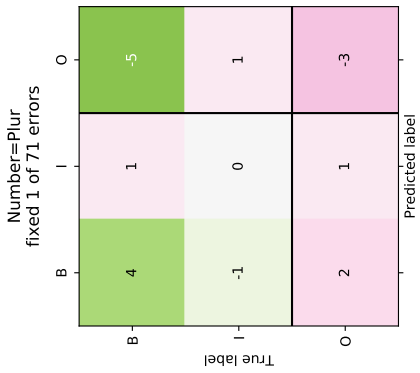
A.3 Confusion Tables for DEPREL

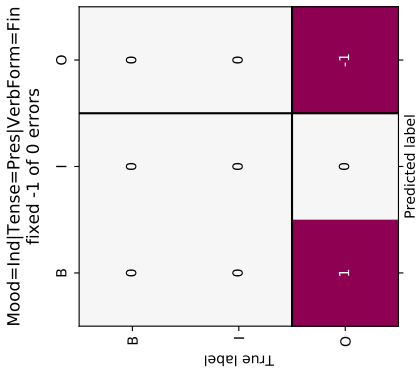


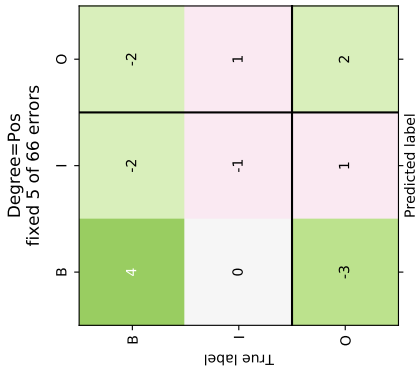
A.4 Confusion Tables for XPOS

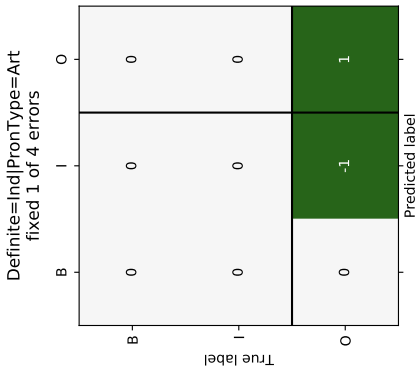


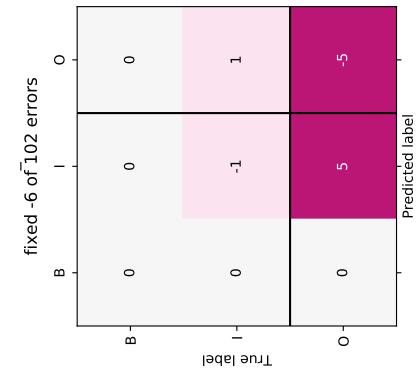
A.5 Confusion Tables for FEATS

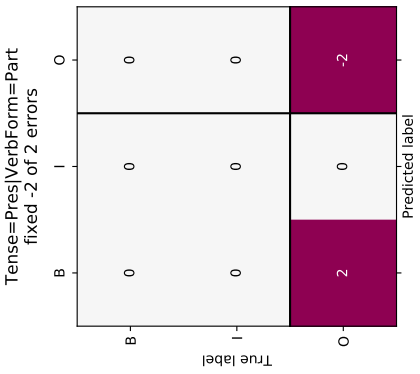


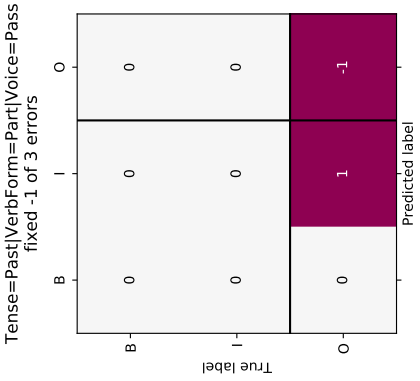


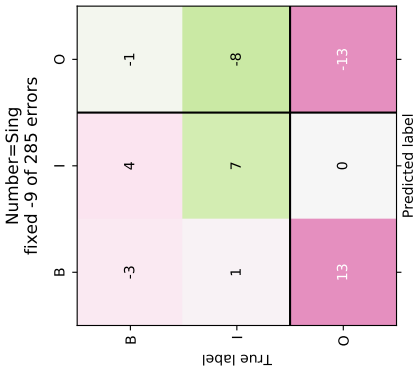




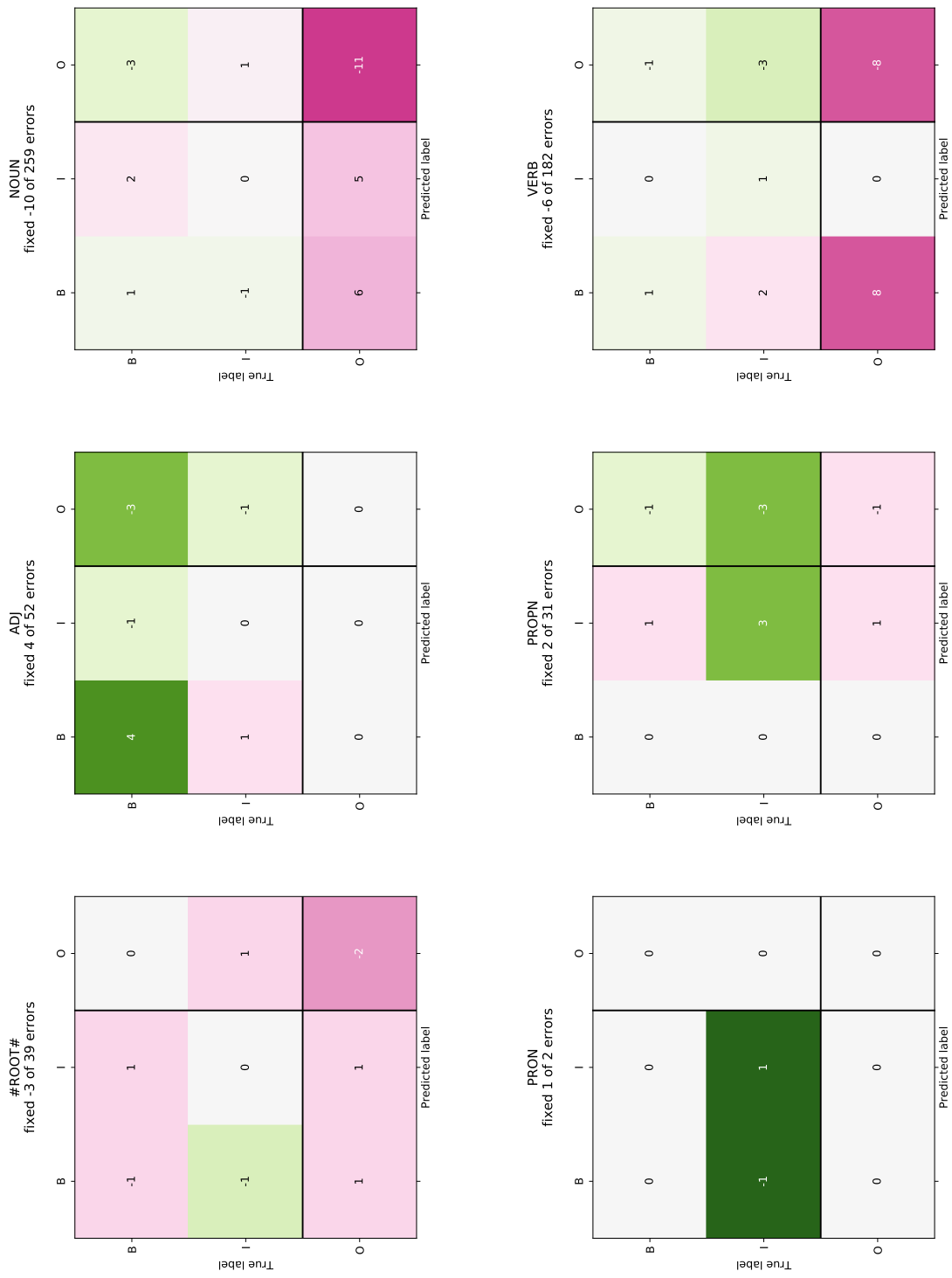




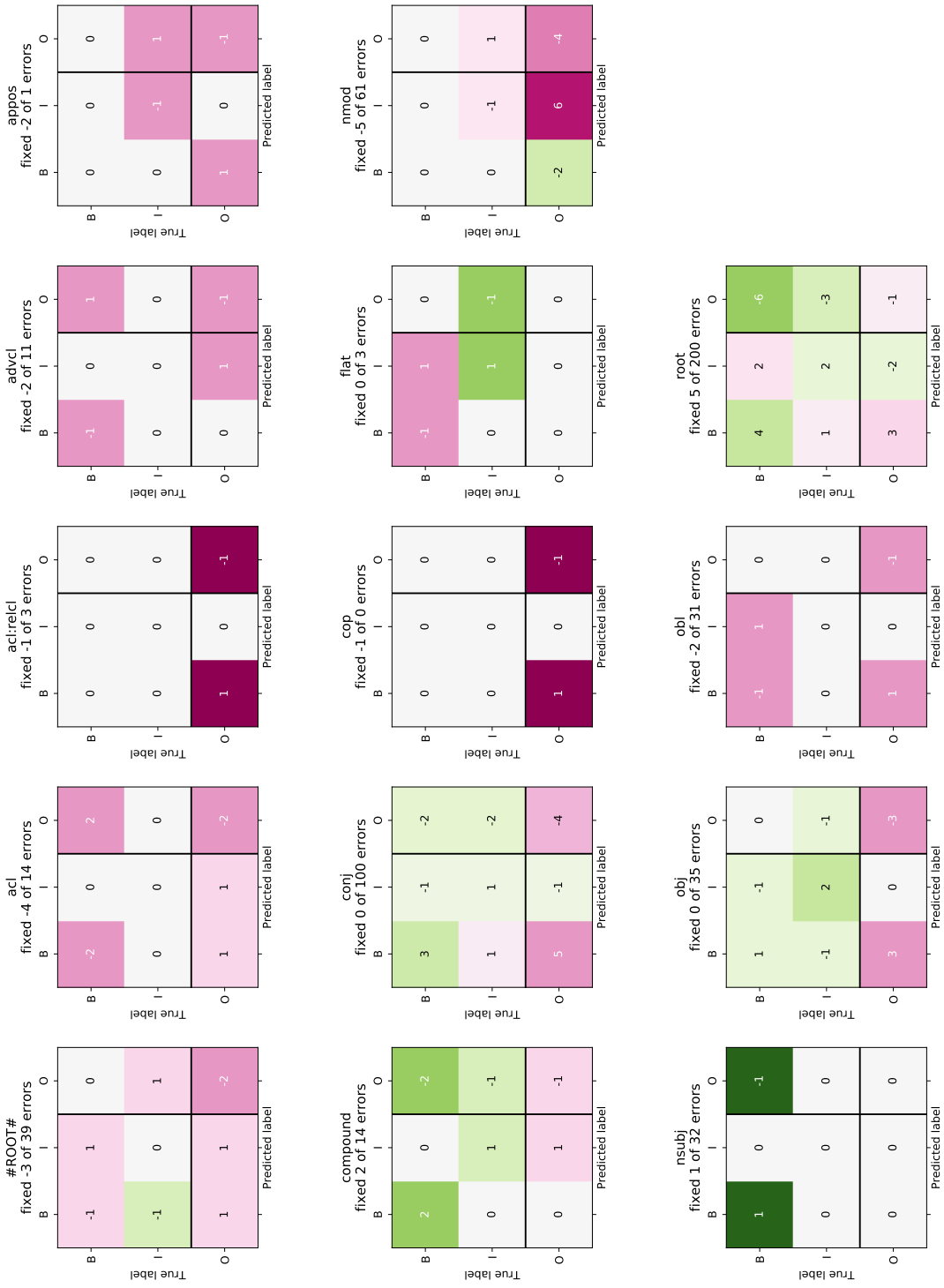




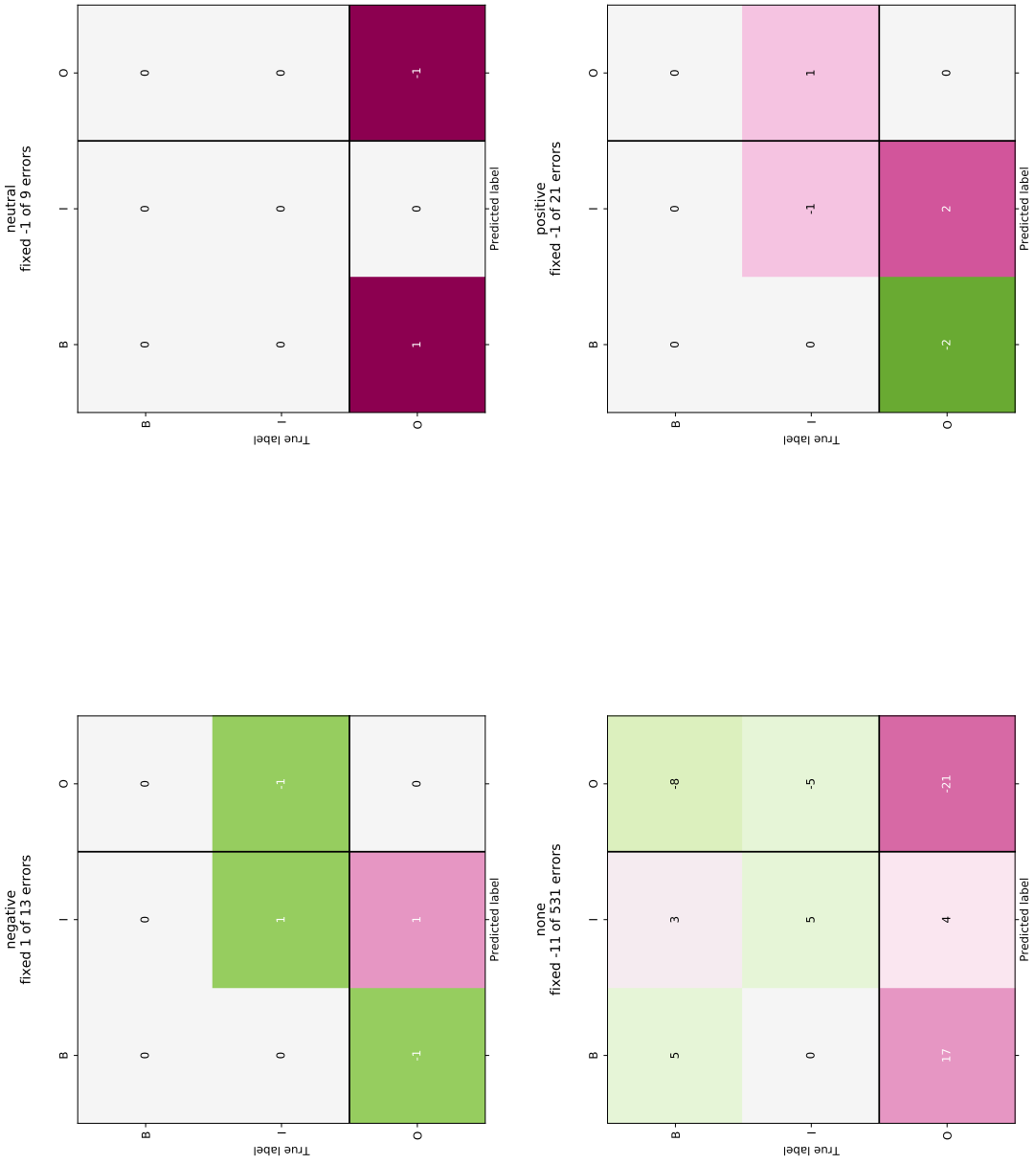
A.6 Confusion Tables for HEAD UPOS



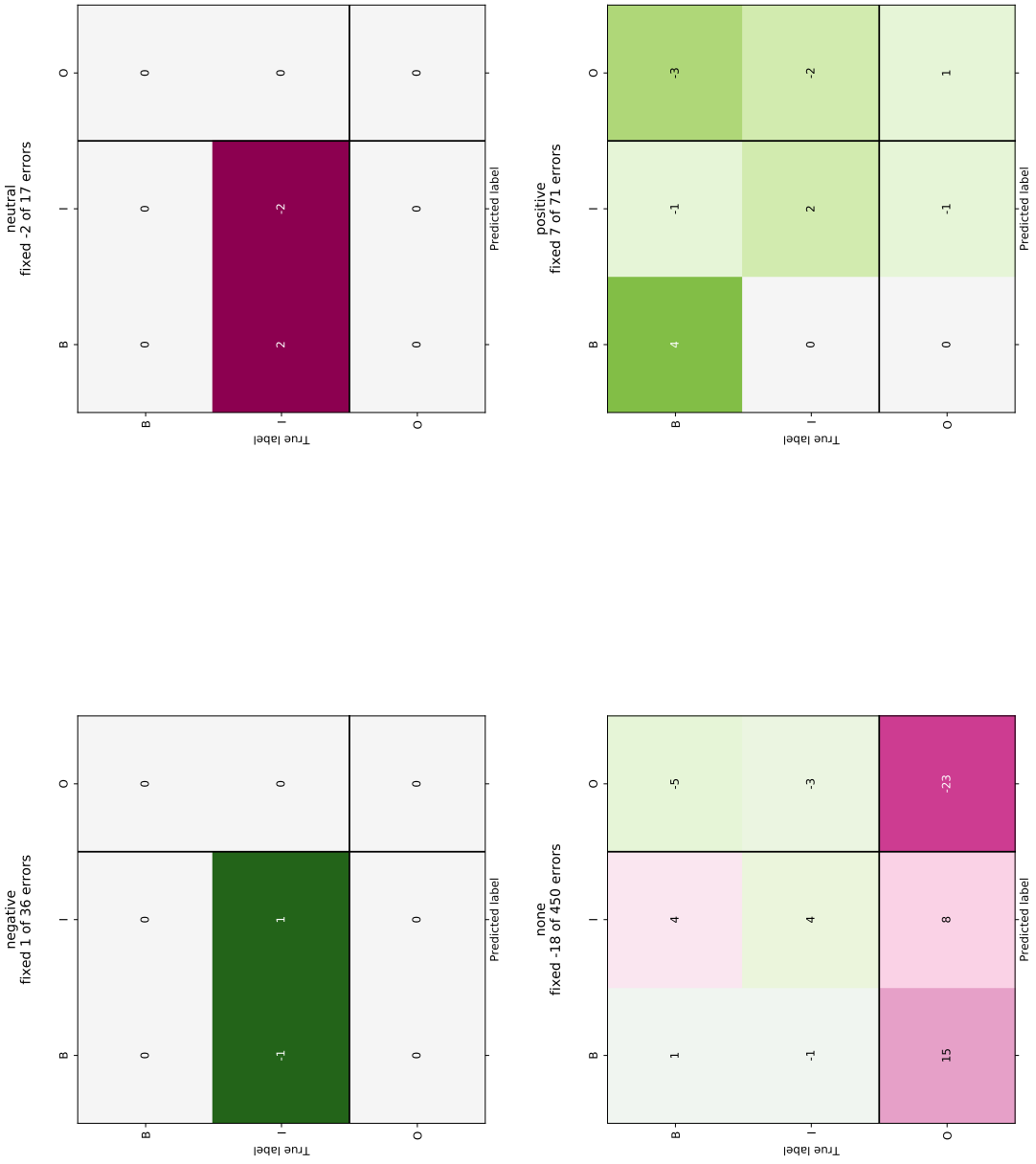
A.7 Confusion Tables for HEAD DEPREL



A.8 Confusion Tables for SENT



A.9 Confusion Tables for HEAD SENT



A.10 Confusion Tables for SIBLING SENT