University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

Aspects of Sentiment Analysis

Ing. Tomáš Hercig

Doctoral Thesis
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Computer Science and Engineering

Supervisor: doc. Ing. Pavel Král, Ph.D.

Pilsen, 2017

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Aspekty analýzy sentimentu

Ing. Tomáš Hercig

Disertační práce
k získání akademického titulu doktor
v oboru Informatika a výpočetní technika

Školitel: doc. Ing. Pavel Král, Ph.D.

Plzeň, 2017

Declaration of Authenticity

I hereby declare that this doctoral thesis is my own original and sole work. Only the sources listed in the bibliography were used.

In Pilsen on October 30, 2017

Prohlašení o původnosti

Prohlašuji tímto, že tato disertační práce je původní a vypracoval jsem ji samostatně. Použil jsem jen citované zdroje uvedené v přehledu literatury.

V Plzni dne 30. října 2017

Ing. Tomáš Hercig

Acknowledgment

First of all, I would like to thank to Dr. Pavel Král and Dr. Ivan Habernal for their guidance.

I would like to thank my colleagues and friends Dr. Tomáš Brychcín and Dr. Michal Konkol for stimulating discussions and great amount of valuable comments. I appreciate the friendly atmosphere in our department and I also thank everyone else who helped me in any way.

Last but not least, I would like to thank my parents for their love and support. My deepest gratitude goes to my wife and my daughter for being my inspiration.

Clarification

During my research, I got married on October 18, 2014, I gave up my former family name Ptáček, and I took my wife's family name Hercig.

Abstract

Sentiment analysis is a sub-field of natural language processing. Generally, it deals with an automatic extraction and analysis of sentiments, opinions, emotions, and beliefs expressed in written text.

Sentiment analysis has become a mainstream research field since the early 2000s. Its impact can be seen in many practical applications, ranging from analysing product reviews to predicting sales and stock markets using social media monitoring.

In order to correctly identify the sentiment hidden in a text, we need to sufficiently understand the meaning (semantics) of the text. However, the semantics of a sentence with figurative language can be quite different from the same sentence with literal meaning. Misinterpreting figurative language such as irony, sarcasm, and metaphor represents a significant challenge in sentiment analysis.

This thesis studies document-level sentiment analysis, aspect-based sentiment analysis, sarcasm detection, and the impact of figurative language on sentiment analysis. We place special emphasis on the Czech language.

Our research includes the creation of data resources for both document-level and aspect-based sentiment analysis, experiments with data preprocessing, feature selection, various features e.g. using semantic models, neural networks, classifiers, and pioneer research into sarcasm detection in Czech. We also explore the impact of figurative language on sentiment analysis.

Abstrakt

Analýza sentimentu je podúloha zpracování přirozeného jazyka, která se obecně zabývá automatickou extrakcí a analýzou pocitů, názorů, emocí a přesvědčení vyjádřených v psaném textu.

Analýza sentimentu se stala hlavní oblastí výzkumu již od počátku nového tisíciletí. Dopad analýzy sentimentu lze pozorovat v mnoha praktických aplikacích, od analýzy recenzí produktů až po předpovědi prodeje a akciových trhů pomocí monitorování sociálních médií.

Abychom správně identifikovali sentiment obsažený v textu, musíme dostatečně pochopit význam (sémantiku) textu. Sémantika věty s obrazným vyjádřením však může být zcela odlišná od téže věty s doslovným významem. Nesprávná interpretace obrazných vyjádření, jako je ironie, sarkasmus a metafora, představuje závažný problém v oblasti analýzy sentimentu.

Náš výzkum zahrnuje tvorbu datových zdrojů jak pro analýzu sentimentu na úrovni dokumentů, tak pro aspektově orientovanou analýzu sentimentu, dále pak experimenty s předzpracováním dat, výběrem příznaků, různými příznaky například s využitím sémantických modelů, neuronovými sítěmi, klasifikátory a průkopnický výzkum detekce sarkasmu v češtině. V práci zkoumáme také dopad použití obrazných vyjádření na analýzu sentimentu.

Contents

Part I

Introduction

1 Introduction

Sentiment analysis is a sub-field of natural language processing (NLP) that usually employs machine learning, computational linguistics, and data mining. Generally, it deals with an automatic extraction and analysis of sentiments, opinions, emotions, and beliefs expressed in written text.

Sentiment analysis has become a mainstream research field since the early 2000s. Its impact can be seen in many practical applications, ranging from analysing product reviews [Stepanov and Riccardi, 2011] to predicting sales and stock markets using social media monitoring [Yu et al., 2013]. The users' opinions are mostly extracted either on a certain polarity scale or on a binary (positive, negative) scale; various levels of granularity are also taken into account, e.g. document level, sentence level, or aspect-based sentiment [Hajmohammadi et al., 2012].

In order to correctly identify the sentiment hidden in a text, we need to sufficiently understand the meaning (semantics) of the text. If we understand the meaning then we will also uncover the sentiment hidden in the text. Thus substantial part of this thesis is dedicated to using distributional semantics to improve sentiment analysis.

A particularly important aspect of semantics is figurative language. The semantics of a sentence with figurative language can be quite different from the same sentence with literal meaning. Misinterpreting figurative language such as irony, sarcasm, and metaphor represents a significant challenge in sentiment analysis. However, the impact of figurative language on sentiment analysis has not yet been studied in depth.

Most of the research in automatic sentiment analysis of social media has been performed in English and Chinese, as shown by several surveys [Liu and Zhang, 2012, Tsytsarau and Palpanas, 2012]. Thus we place special emphasis on sentiment analysis in Czech.

1.1 Motivation

There are many researchers trying to surpass the latest best results and achieve the state of the art in English sentiment analysis by using handcrafted features. This approach may result into overfitting the data. However, sentiment analysis in Czech has not yet been thoroughly targeted by the research community.

Czech, as a representative of inflectional languages, is an ideal environment for the study of various aspects of sentiment analysis (overview or breadth study of sentiment analysis if you will). It is challenging because of its very flexible word order, multiple negatives, and many different word forms for each lemma.

We conceive this thesis to deal with several aspects of sentiment analysis. The breadth of this thesis can lead to more general view and better understanding of sentiment analysis. We can reveal and overcome unexpected obstacles, create necessary evaluation datasets and even come up with new creative solutions to sentiment analysis tasks.

Thus the aim of this doctoral thesis is to study various aspects of sentiment analysis with the emphasis on the Czech language.

1.2 Thesis Goals

The following goals were set for this thesis in author's Ph.D. thesis exposé [Hercig, 2015]. While the underlying goal is to propose novel methods for improving performance of sentiment analysis with special emphasis on inflectional languages (e.g. Czech), the focus is on the following research tasks:

1. Deal with specific properties of Czech language in the sentiment analysis environment.
2. Use additional semantic and/or syntactic information to improve sentiment analysis.
3. Explore the influence of figurative language (e.g. sarcasm) on sentiment analysis.

1.3 Outline

Chapter 2 describes the challenges in sentiment analysis and formulates the basic sentiment and aspect-based sentiment definitions.

It is necessary to define the state-of-the-art techniques before some results are presented, thus Chapter 3 is devoted to machine learning techniques. The commonly used features for sentiment analysis are covered in Chapter 4.

Distributional semantic models are introduced in Chapter 5. Semantic models can be used as additional sources of information for sentiment analysis.

The related work for sentiment analysis is presented in Chapter 6.

The rest of this thesis describes our experiments and results and represents our contribution related to the sentiment analysis task.

Chapter 7 covers our in-depth research on machine learning methods for document-level sentiment analysis of Czech social media.

In Chapter 8 we describe our approaches to the aspect-based sentiment analysis task in Czech and English.

Chapter 9 presents the first attempt at using neural networks for sentiment analysis in Czech.

Chapter 10 describes our approach to sarcasm detection in Czech and English.

We explore the effect of figurative language on sentiment analysis in Chapter 11.

Chapter 12 summarizes our work, fulfilment of the thesis goals and reveals our future plans.

Part II

Theoretical Background

2 Sentiment Analysis

Sentiment analysis in general is not only connected to opinions but to emotions, feelings, and attitudes as well. Sentiment polarity assigns a sentiment label (e.g. positive, negative, and neutral) to texts, however it is only a part of this field. In this thesis we mainly focus on the sentiment polarity task.

This chapter describes the core problems of the current state-of-the-art algorithms and presents the formal definition of sentiment analysis.

2.1 Challenges

In this section we present the most important issues in sentiment analysis.

The sentiment polarity of a word may have opposite orientations in different contexts. The word *"loud"* is generally negative (*"the fan is very loud"*), however in a certain situation it can be positive (*"wow the speakers are really loud"*).

A sentence containing sentiment bearing words may not express any sentiment. This frequently happens in questions and conditional sentences, e.g. *"Could you tell me which printer is the best?"* and *"If I can find a good laptop in the shop, I will buy it."*. Both sentences contain a positive sentiment bearing word, but neither expresses a positive or negative opinion on any specific product. However, not all questions and conditional sentences express no sentiments, e.g. *"Does anyone know how to get this terrible camera to work?"*.

Other aspects of subjective texts related to sentiment can be considered important as well. Various emotions such as anger, fear, disgust, happiness, sadness, and surprise can be extracted from affected texts in order to determine the state of mind of the author. This affected state can be later used to switch to a different mode of sentiment interpretation or hateful posts filtering in forums [Pang and Lee, 2008].

Sarcastic sentences with or without sentiment bearing words are hard to deal with, e.g. *"What a great car! It stopped working in two days."* Sarcasm will be discussed in more detail in Chapter 10.

Many sentences without sentiment bearing words can also imply opinions. These sentences usually express some factual information in an objective manner. The sentence *"This printer uses a lot of ink"* implies a negative sentiment about the printer since it uses a lot of resource (ink). This sentence is objective as it states a fact.

Unlike factual information, opinions, and sentiments have an important characteristic, namely, they are subjective. Single opinion from one person represents only the subjective view of that single person. It is thus important to examine a collection of opinions from many people rather than only a single person. Product reviews are highly focused with little irrelevant information and rich with opinions. They allow us to see different issues more clearly than from other forms of opinion text.

Texts from various sources have their own specific problems. Twitter postings (tweets) are short (at most 140 characters), informal, and use many Internet slangs and emoticons. Twitter postings are easier to analyse due to the length limit because the authors are usually straight to the point, but you have to deal with the Twitter specific slang [Liu, 2012].

Forum discussions are perhaps the hardest to deal with because the users there can discuss anything and also interact with one another. Different application domains are also considered very difficult to deal with. Social and political discussions are much harder than opinions about products and services, due to complex topic and sentiment expressions [Liu, 2012].

The task of aggregating and representing sentiment of one or majority of documents is called sentiment summarization. Since the amount of information available on the Internet is huge, a brief overview of market sentiment can be very helpful for both customers and producers. Unlike humans, automatic summarization should be unbiased, quick, and accurate. Moreover, the average human reader could have considerable difficulty doing the same.

There are even individuals who give fake opinions in reviews and forum discussions to promote or to discredit target products, services, organizations, or individuals. The fake opinions are called opinion spam and the authors are called opinion spammers. Opinion spamming has become a major issue. There is no easy way to detect these fake opinions [Liu, 2012].

2.2 Definition of Basic Sentiment Polarity

An opinion is a quadruple (G, S, H, T) [Liu, 2012], where

- G is the sentiment target,
- S is the sentiment about the target,
- H is the opinion holder,
- T is the time when the opinion was expressed.

Sentiment analysis can be done on different levels of granularity:

- **Document level** is usually used on various reviews, where the task is to determine the overall sentiment towards the target (e.g. product or movie).
- **Sentence level** analyses the overall sentiment of a sentence.
- **Aspect-based** sentiment analysis focuses on the precise features (aspects) of the sentiment target. Aspect-based sentiment analysis will be discussed in Sections 2.3 and 6.3.
- **Word level** identifies the polarity of words. For more information see Section 6.2.

Let us use the term entity to denote the target object that has been evaluated. An entity is a product, service, topic, issue, person, organization, or event. It is described with a hierarchy of parts, sub-parts, and so on, and a set of attributes. Each part or sub-part also has its own set of attributes [Liu, 2012]. Figure 2.1 shows an example of such hierarchy.

This entity (hierarchy of any number of levels) needs a nested relation to represent it. Recognizing parts and attributes of an entity at different levels of detail is extremely hard, fortunately most applications do not need such complex analysis. Thus, we simplify the hierarchy to two levels and use the term aspects to denote both parts and attributes. In the simplified tree, the root node is still the entity itself, but the second level (also the leaf level) nodes are different aspects of the entity. This simplified framework (Figure 2.2) is what is typically used in practical sentiment analysis systems. Note that in the research literature, entities are also called objects, and aspects are also called features (or product features).

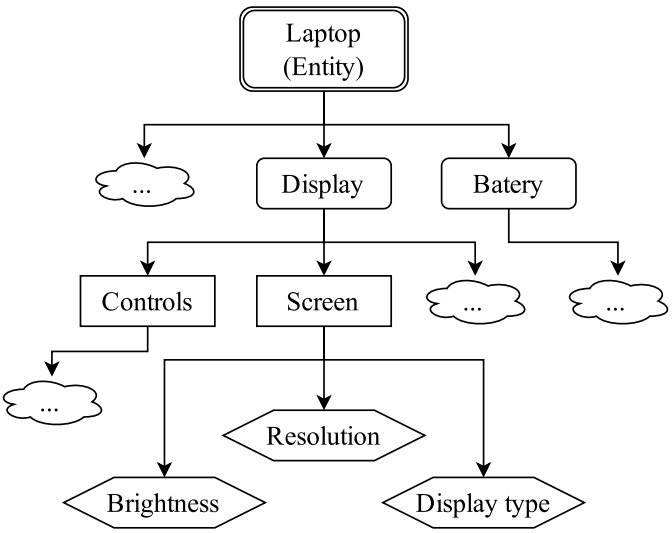


Figure 2.1: Example entity (laptop), its parts (rounded rectangle), sub-parts (rectangle) and attributes (hexagon). Clouds represent omitted hierarchical structures.

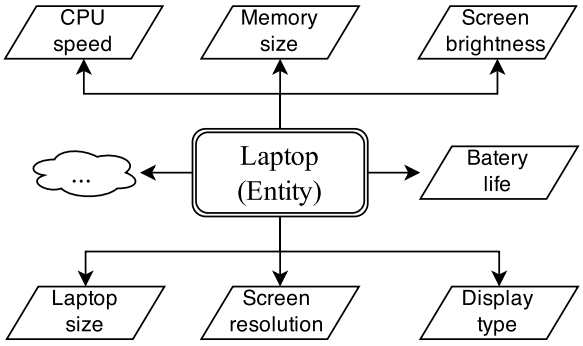


Figure 2.2: Example entity (laptop) and its aspects (rhomboids). Cloud represents omitted aspects.

2.3 Definition of Aspect-Based Sentiment

An opinion is a quintuple $(E_i, A_{ij}, S_{ijkl}, H_k, T_l)$ [Liu, 2012], where

- E_i is the name of an entity,
- A_{ij} is an aspect of E_i,
- S_{ijkl} is the sentiment about aspect A_{ij} of entity E_i expressed by H_k at the time T_l,

- H_k is the opinion holder,

- T_l is the time when the opinion is expressed by H_k.

The entity E_i and its aspects A_{ij} together represent the opinion target. The sentiment S_{ijkl} is positive, negative, or neutral, or expressed on a certain polarity scale, e.g. 1 to 5 stars as used by most review sites. Special aspect GENERAL is used to denote an opinion on the entity itself as a whole.

In this definition, subscripts are used to emphasize that the five pieces of information in the quintuple must correspond to one another. That is, the opinion S_{ijkl} must be given by opinion holder H_k about aspect A_{ij} of entity E_i at time T_l. Each of these five components is essential and any mismatch is problematic in general.

For example, in the sentence *"The English adore him but the Spanish hate him."* it is clearly important to distinguish between the two opinion holders. The time component may seem not very important, but in practise an opinion expressed two years ago is not the same as an opinion expressed yesterday.

The definition does not cover all possible ways to express an opinion. The definition would be too complex if it did, and thus would make the problem extremely difficult to solve. However, the definition is sufficient for most applications.

The limits of this simplification are evident, e.g. in the case of a comparative opinion. Comparative opinion expresses a relation of similarities or differences between two or more entities and/or a preference of the opinion holder based on some shared aspects of the entities [Liu, 2012].

There are other situations in which a more complex definition would be needed. For example, the situation in *"This car is too small for a tall person"*, which does not say the car is too small for everyone. The context of the opinion is an important information, which is not covered in the simplified definition.

Furthermore, we simplified the hierarchical structure of an entity. If we want to study different aspects of an aspect (e.g. phone battery and its price and capacity), then we need to treat an aspect (battery) of an entity (phone) as a separate entity.

Definition from Semantic Evaluation Workshop

The semantic evaluation workshop (*SemEval*) is an important series of workshops studying multiple tasks. Sentiment analysis is one of the tasks. There are several ways to define aspects and polarities.

The definition of the aspect-based sentiment analysis (ABSA) task from SemEval 2014 [Pontiki et al., 2014] distinguishes two types of aspect-based sentiment for aspect terms and aspect categories. Thus, the whole task is divided into four subtasks.

The later SemEval's ABSA tasks [Pontiki et al., 2015, 2016] further distinguish between more detailed aspect categories and associate aspect terms (targets) with aspect categories.

SemEval 2014

- **Subtask 1: Aspect Term Extraction (TE)**

  Given a set of sentences with pre-identified entities (e.g. restaurants), the task is to identify the aspect terms present in the sentence and return a list containing all the distinct aspect terms.

  ```
  Our server checked on us maybe twice during the entire meal.
  → {server, meal}
  ```

- **Subtask 2: Aspect Term Polarity (TP)**

  For a given set of aspect terms within a sentence, the task is to determine the polarity of each aspect term: positive, negative, neutral, or bipolar (i.e. both positive and negative).

  ```
  Our server checked on us maybe twice during the entire meal.
  → {server: negative, meal: neutral}
  ```

- **Subtask 3: Aspect Category Extraction (CE)**

  Given a predefined set of aspect categories (e.g. price, food), the task is to identify the aspect categories discussed in a given sentence. Aspect categories are typically coarser than the aspect terms of Subtask 1, and they do not necessarily occur as terms in the given sentence. In the analysed domain of *"restaurants"*, the categories include food, service, price, and ambience.

```
We were welcomed by a very nice waitress and a room with
time-worn furniture.
→ {service, ambience}
```

- **Subtask 4: Aspect Category Polarity (CP)**

  Given a set of pre-identified aspect categories (e.g. food, price), the task is to determine the polarity (positive, negative, neutral, or bipolar) of each aspect category.

```
We were welcomed by a very nice waitress and a room with
time-worn furniture.
→ {service: positive, ambience: negative}
```

SemEval 2016

The ABSA task from SemEval 2016 [Pontiki et al., 2016] has three subtasks: Sentence-level (SB1), Text-level (SB2), and Out-of-domain ABSA (SB3). The subtasks are further divided into three slots. The following example is from the training data (including the typographical error).

- **1) Aspect Category Detection** – identify (predefined) aspect category – entity and attribute (E#A) pair.

```
The pizza is yummy and I like the atmoshpere.
→ {FOOD#QUALITY, AMBIENCE#GENERAL}
```

- **2) Opinion Target Expression (OTE)** – extract the OTE referring to the reviewed entity (aspect category).

```
The  pizza is yummy and I like the  atmoshpere.
→ {pizza, atmoshpere}
```

- **3) Sentiment Polarity** – assign polarity (positive, negative, and neutral) to each identified E#A, OTE tuple.

```
The pizza is yummy and I like the atmoshpere.
→ {FOOD#QUALITY - pizza: positive,
   AMBIENCE#GENERAL - atmoshpere: positive}
```

2.4 Sentiment Analysis of Inflectional Languages

Highly inflectional languages such as Czech are hard to deal with because of the high number of different word forms. Czech is even more challenging because it has very flexible word order. Czech language permits and frequently uses double even a triple negation in one sentence, thus making it difficult for computers to understand the meaning of the sentence. Moreover, the subject can be omitted if it is known from the context.

Text is often preprocessed by various techniques in order to reduce the dictionary size. The importance of this preprocessing phase depends on the language. For highly inflectional languages like Czech, *stemming* or *lemmatization* is almost mandatory because it is necessary to reduce the high number of different word forms.

Lemmatization identifies the base (dictionary) form of a word which is known as the *lemma*. *Stemming* finds the base form of each word, usually by removing all affixes. The result of *stemming* is called *stem*. Sometimes a list of stop words is used to filter out words which occur in most documents and have only a small impact on the results.

2.5 Evaluation Criteria

Sentiment analysis is evaluated by accuracy, precision, recall, and F-measure (also denoted as F-score or F_1 score).

3 Machine Learning

Sentiment analysis can be treated as a text classification problem. The standard approach is to classify a document as being positive or negative using a machine learning algorithm (classifier). The performance of sentiment analysis is strongly dependant on the applied classifier.

Machine learning algorithms essentially learn and store characteristics of a category from the data during a training phase. This is achieved by observing the properties of the annotated training data. The acquired knowledge[1] is later applied to determine the best category for the unseen testing dataset. The training and testing datasets are both annotated by sentiment labels. Various model validation techniques can be used depending on the data-size. *Cross-validation* is commonly used for sentiment analysis evaluations. The annotated dataset is split into k equal parts, then the first part is treated as the testing data and the rest as training data, this selection process is repeated for each of the parts. Each part is used exactly once as the testing data.

The de-facto standard for sentiment analysis are the Maximum Entropy classifier and Support Vector Machines (SVM) classifier, however a simple Naive Bayes classifier is often used as a baseline for evaluation. Recently the standard is gradually being taken over by neural networks.

We also present a brief overview of neural networks in Section 3.4, however for a detailed description see e.g. [Graupe, 2013, Schmidhuber, 2015, Goodfellow et al., 2016].

3.1 Naive Bayes Classifier

The Naive Bayes (NB) classifier is a simple classifier commonly used as a baseline for many tasks. The model computes the posterior probability of

[1] Trained classification model with parameters.

a sentiment label based on predefined features in a given text as shown in equation 3.1, where s is the sentiment label and x is the feature vector created from the given text.

$$P(s|x) = \frac{P(x|s)P(s)}{P(x)} \tag{3.1}$$

$$\hat{s} = \arg\max_{s \in S} P(s)P(x|s) \tag{3.2}$$

The NB classifier is described by equation 3.2, where $\hat{s}$ is the assigned sentiment label. The NB classifier makes the decision based on the maximum a posteriori rule. In other words it picks the sentiment label that is the most probable. The NB classifier assumes conditional independence of features.

3.2 Maximum Entropy Classifier

The Maximum Entropy (MaxEnt) classifier is based on the Maximum Entropy principle. The principle says that we are looking for a model which will satisfy all our constraints in the most general way. We are looking for the maximum value of the Entropy. We want to find the conditional probability distribution $p(s|x)$ with maximum Entropy H.

$$\arg\max_{p(s|x)} H(s|x) = -\sum_{x} p(x) \sum_{s} p(s|x)\ log\ p(s|x) \tag{3.3}$$

To define a constraint we firstly need to define a feature. A feature is typically a binary function[2]. For example, consider the following dictionary feature designed to capture positive emoticons in the given text x.

$$f(x,s) = \begin{cases} 1 & \text{if } s \text{ is positive and } x \text{ contains a positive emoticon} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The constraint is then defined as equality of mean values for a given feature.

$$E_p(f_i(x,s)) = E_{\tilde{p}}(f_i(x,s)) \tag{3.5}$$

[2] In general any non-negative function can be used.

$E_{\tilde{p}}(f_i(x,s))$ is the mean value of a feature computed over the training data and $E_p(f_i(x,s))$ is the mean value of the model. It is guaranteed that such a model exists, it is unique and follows the maximum-likelihood distribution (equation 3.6)[Berger et al., 1996].

$$p(s|x) = \frac{1}{Z(x)} \exp \sum_i \lambda_i f_i(x,s) \tag{3.6}$$

The $f_i(x,s)$ is a feature and λ_i is a parameter to be estimated. $Z(x)$ is just a normalizing factor and ensures that $p(s|x)$ is a probability distribution.

$$Z(x) = \sum_s \exp \sum_i \lambda_i f_i(x,s) \tag{3.7}$$

Various training algorithms can be used for finding appropriate parameters. Limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method [Nocedal, 1980] proved very good performance.

3.3 SVM Classifier

Support Vector Machines (SVM) is a machine learning method based on vector spaces, where the goal is to find a decision boundary between two classes that represents the maximum margin of separation in the training data [Manning et al., 2008].

SVM can construct a non-linear decision surface in the original feature space by mapping the data instances non-linearly to an inner product space where the classes can by separated linearly with a hyperplane.

Support Vector Machines

Following the original description [Cortes and Vapnik, 1995] we describe the principle in the simplest possible way. We will assume only binary classifier for classes $y = -1, 1$ and linearly separable training set $\{(x_i, y_i)\}$, so that the conditions 3.8 are met.

$$\begin{array}{rcrl} \mathbf{w} \cdot \mathbf{x_i} + b & \leq & -1 & \text{if } y_i = -1 \\ \mathbf{w} \cdot \mathbf{x_i} + b & \geq & 1 & \text{if } y_i = 1 \end{array} \tag{3.8}$$

Equation 3.9 combines the conditions 3.8 into one set of inequalities.

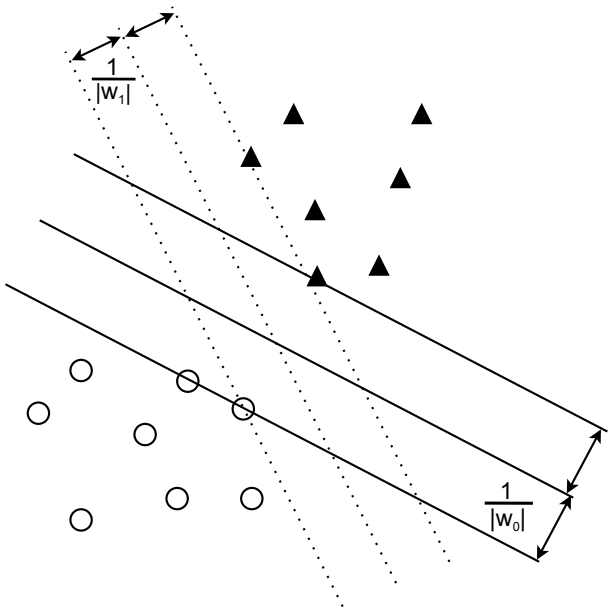


Figure 3.1: Optimal and suboptimal hyperplanes.

$$y_i \cdot (\mathbf{w}_0 \cdot \mathbf{x} + b_0) \geq 1 \quad \forall i \tag{3.9}$$

SVM search the optimal hyperplane (equation 3.10) that separates both classes with the maximal margin. The formula 3.11 measures the distance between the classes in the direction given by $\mathbf{w}$.

$$\mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0 \tag{3.10}$$

$$d(\mathbf{w}, b) = \min_{x;y=1} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} - \max_{x;y=-1} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} \tag{3.11}$$

The optimal hyperplane, expressed in equation 3.12, maximizes the distance $d(\mathbf{w}, b)$. Therefore the parameters $\mathbf{w}_0$ and b_0 can be found by maximizing $|\mathbf{w}_0|$. For better understanding see the optimal and suboptimal hyperplanes in Figure 3.1.

$$d(\mathbf{w}_0, b_0) = \frac{2}{|\mathbf{w}_0|} \tag{3.12}$$

The classification is then just a simple decision on which side of the hyperplane the object is.

For non-linear datasets a kernel function is used to map the data into a higher dimensional space in which they can be linearly separated. There are number of kernels that can be used e.g. linear, polynomial, radial basis function, and sigmoid function. Sequential Minimal Optimization [Platt, 1998] breaks multi-class classification problems into multiple binary classification problems (one-vs-one or one-vs-all). Crammer and Singer [2002] cast this problem into a single optimization problem rather then decomposing it into multiple binary classification problems.

3.4 Neural Networks

Artificial neural network (ANN or just NN) is loosely modeled after the human brain and consists of many simple connected units called neurons (or nodes). Neurons process their inputs and based on assigned weights and their activation function produce an output. The activation function is a nonlinear transformation of inputs. Both input and output are real-valued numbers. The connection is associated with a weight and passes the output of one neuron to the input of another. These interconnected neurons usually form layers of the neural network. The input layer receives our data (real-valued word vectors) and produce an output which is the input of the next layer. The output layer is usually smaller and corresponds to the given problem (e.g. classification). In NLP applications the input is usually a feature vector for each word in the sentence.

The activation function is a typically a nonlinear real-valued activation function. Some currently popular activation functions are:

- Hyperbolic tangent ($tanh$)
- Rectified linear unit (ReLU) $f(x) = x$ for $x \geq 0$, $f(x) = 0$ otherwise.
- Sigmoid $f(x) = \dfrac{e^x}{e^x + 1}$

The simplest neural network is a perceptron (a single neuron). When the weighted sum of inputs is greater than the selected threshold the output is 1 and zero otherwise.

A connected network of neurons, where the connections do not form a cycle is called a feed-forward neural network (e.g. Figure 3.2).

In Figure 3.2 we denote $a_i^{(l)}$ the activation (meaning the output value) of

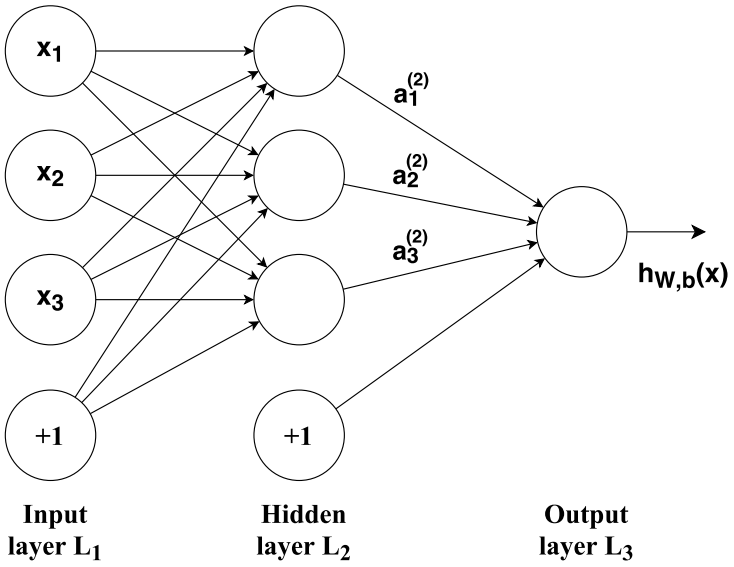


Figure 3.2: Example of a feed-forward neural network.

a neuron in layer l ($a_i^{(1)} = x_i$ denotes the i-th input). The circles labeled "+1" are called bias units. Hypothesis $h_{W,b}(x)$ is defined by equation 3.13, where W are weights associated with connections and function f applies to vectors in an element-wise fashion. This is called forward propagation and it can be generalized for layer l as $a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)})$.

$$h_{W,b}(x) = a^{(3)} = f(W^{(2)}a^{(2)} + b^{(2)}) = f(W^{(2)}f(W^{(1)}x + b^{(1)}) + b^{(2)}) \quad (3.13)$$

Training of neural networks involves both forward-propagation and back-propagation, where the gradients are calculated for all weights. The optimization of the weights is usually done by the stochastic gradient descent [Bottou, 1998]. We calculate error using the loss function (also called error, objective, or cost function). The error gradients are sent back through the network using the same weights that were used in the forward pass. Learning rate α controls how much we change the weights. This process (training epochs) is usually repeated several times.

Deep neural network (DNN) is a neural network where the number of hidden layers is 2 or more [Goodfellow et al., 2016].

Recurrent neural networks (RNN) include feedback connections (they have loops in the network diagram) allowing them to keep information from

previous events. They are usually used to sequences of data (time data, events, recurring events)

Long short-term memory (LSTM) [Schmidhuber and Hochreiter, 1997] is a recurrent neural network that uses memory cells to remember previous values. Thus, they are able to learn long-term dependencies. The key to learning the long-term dependencies are gates. The single gating unit (Gated Recurrent Unit – GRU) simultaneously controls the forgetting factor and decides to update the state [Goodfellow et al., 2016].

Convolutional Neural Network (CNN) [LeCun et al., 1989, Kim, 2014] employ mathematical operation called convolution and usually an operation called pooling [Goodfellow et al., 2016]. A pooling function approximates the output of the neural network at a certain location with a summary statistic of the nearby outputs (max pooling is often used in NLP)[Goodfellow et al., 2016].

Recursive Neural Network is another generalization of recurrent networks, where the network is structured as a deep tree rather than the chain-like RNN [Goodfellow et al., 2016]. The tree structure is modeled e.g. by the sentence parse tree [Socher et al., 2013].

4 Features

Choosing the best feature set for sentiment analysis has high importance as it has a strong impact on the evaluation results. This chapter describes the most common features.

Features are often preprocessed by various techniques in order to reduce the feature space. The importance of this preprocessing phase depends on the language (for more information see Section 2.4).

A *stem* or a *lemma* can be used directly as a feature similarly to a simple unigram feature. *Stemming* or *lemmatization* can also improve the performance of other features.

4.1 N-gram Features

N-grams and their frequency or presence is often used as a valid baseline. In some cases word positions and TF-IDF weighting scheme [Manning et al., 2008] may be considered effective features.

N-gram Word n-grams are used to capture frequent word sequences. The presence of unigrams, bigrams, and trigrams is often used as binary features. The feature space is pruned by the minimum n-gram occurrence (e.g. 5). Note that this is the baseline feature in most of the related works.

Character N-gram Similarly to the word n-gram features, character n-gram features can be used, as proposed by, e.g. Blamey et al. [2012]. Character trigrams are often used to capture frequent emoticons. The feature set usually contains 3-grams to 6-grams. The feature space is further pruned by the minimum occurrence of a particular character n-gram.

Skip-bigram Instead of using sequences of adjacent words (n-grams) we can use skip-grams [Guthrie et al., 2006], which skip over arbitrary gaps. Basic

approach uses skip-bigrams with 2 or 3 word skips and removes skip-grams with low frequency.

Bag of Words Set of words without any information on the word order is called bag of words. This can be viewed as a special case of n-grams where $n = 1$.

Other N-gram features do not have to use only words, any item will do. For example POS patterns are simply POS n-grams.

4.2 POS-related Features

Direct usage of part-of-speech (POS) n-grams that cover sentiment patterns has not shown any significant improvement in the related work. Still, POS tags do provide certain characteristics of a text. Various POS-related features have been used in related work e.g. the number of nouns, verbs, and adjectives [Ahkter and Soria, 2010], the ratio of nouns to adjectives and verbs to adverbs [Kouloumpis et al., 2011], and the number of negative verbs obtained from POS tags.

4.3 Lexical Features

Additional lexical resources such as sentiment lexicons or *SentiWordNet* [Baccianella et al., 2010] can be used as features. These resources use external knowledge to improve the results of sentiment analysis. This is a form of supervision without context. More lexical resources are mentioned in Section 6.2.

4.4 Semantic Features

Distributional Semantics (see Section 5) represents the new trend in sentiment analysis. This is because of its ability to represent the meaning of texts simply by using a statistical analysis. For example the direct application of a joint sentiment and topic model[1] proved to be useful [Lin and He, 2009]. Alternatively, semantics models can be used as new sources of information for classification (e.g. feature vectors, bag of words, or bag of clusters).

[1]Statistical model discovering abstract topics in documents.

4.5 Other Features

Syntactic Features Features trying to capture word dependencies and sentence structure usually by exploiting syntactic information generated from parse trees.

Orthographic Features Features based on the appearance of the word (sometimes called word shape), e.g. the first letter is a capital letter, all letters are capital or the words consists of digits (e.g. [Go et al., 2009, Agarwal et al., 2011]).

Emoticons Lists of positive and negative emoticons (e.g. Montejo-Ráez et al. [2012]) capture the number of occurrences of each class of emoticons within the text.

Punctuation-Based Features Features consisting of special characters, number of words, exclamation marks, question marks, and quotation marks. These features usually do not significantly improve the results (e.g. [Davidov et al., 2010]).

5 Distributional Semantics

As mentioned in Chapter 4, semantics models represent the new trend in sentiment analysis. They can be applied directly to jointly model sentiment and topics or alternatively, the features derived from semantics models can be used as new sources of information for classification.

The backbone principle of methods for discovering hidden meaning in a plain text is the formulation of the *Distributional Hypothesis* [Harris, 1954, Firth, 1957]. The famous quote of Firth [1957] says that *"A word is characterized by the company it keeps."* The direct implication of this hypothesis is that the meaning of a word is related to the context where it usually occurs and thus it is possible to compare the meanings of two words by a statistical comparison of their contexts. This implication was confirmed by empirical tests carried out on human groups in [Rubenstein and Goodenough, 1965, Charles, 2000]. The models based on the Distributional Hypothesis are often referred to as *distributional semantics models*.

Some distributional semantic models use the *Bag-of-word* hypothesis (e.g. LDA). *Bag-of-word* hypothesis assumes that the word order has no meaning. The term bag means a set where the order of words has no role.

Distributional semantics models typically represent the meaning of a word as a vector: the vector reflects the contextual information of the word throughout the training corpus. Each word $w \in W$ (where W denotes the word vocabulary) is associated with a vector of real numbers. Represented geometrically, the word meaning is a point in a high-dimensional space. The words that are closely related in meaning tend to be closer in the space.

The ability to compare two words enables us to use a clustering method. Similar words are clustered into bigger groups of words (clusters). Example of such a method is the k-means algorithm, which is often used because of its efficiency and acceptable computational requirements. Cosine similarity is commonly used as a similarity measure of two words. It is calculated as

the cosine of the angle between the corresponding word vectors.

5.1 HAL

Hyperspace Analogue to Language (HAL) [Lund and Burgess, 1996] is a simple method for building semantic space. HAL records the co-occurring words into a matrix. The words are observed in a small context window around the target word in the given corpus. The Co-occurring words are weighted inversely to their distance from the target word. This results in the co-occurrence matrix $\mathbb{M} = |W| \times |W|$, where $|W|$ is the size of the vocabulary. Finally, the row and column vectors of $\mathbb{M}$ represent the co-occurrence information of the words appearing before and after the target word.

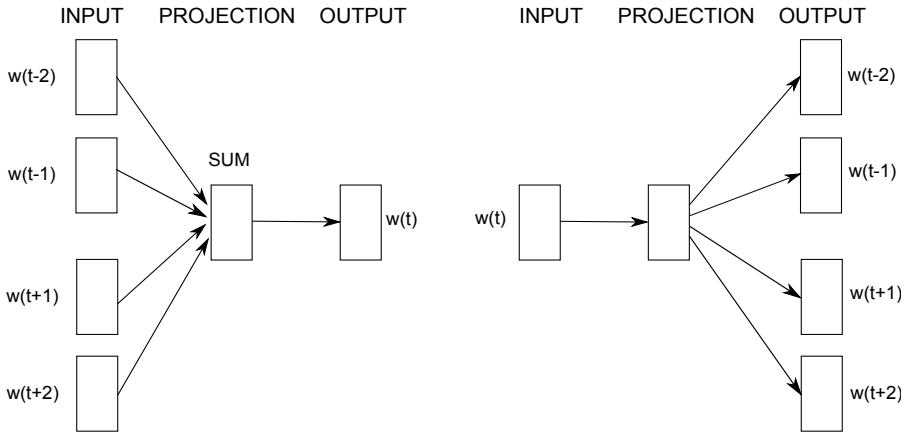
5.2 COALS

Correlated Occurrence Analogue to Lexical Semantics (COALS) [Rohde et al., 2004] extends the HAL model. The difference is that after recording the co-occurrence information, the raw counts of $\mathbb{M}$ are converted into Pearson's correlations. Negative values are reset to zero and other values are replaced by their square roots. The optional final step, inspired by LSA [Deerwester et al., 1990], is to apply the singular value decomposition (SVD) to $\mathbb{M}$, resulting in a dimensionality reduction and also the discovery of latent semantic relationships between words.

5.3 CBOW

Continuous Bag-of-Words (CBOW) [Mikolov et al., 2013a] tries to predict the current word using a small context window around the word. This model estimates word vector representation based on the context. Instead of using a co-occurrence matrix this model uses a neural network for the meaning extraction.

The architecture is similar to the feed-forward Neural Network Language Model (NNLM) proposed in [Bengio et al., 2006]. The NNLM is computationally expensive between the projection and the hidden layer. In CBOW, the (non-linear) hidden layer is removed and the projection layer is shared between all the words. The word order in the context does not influence the projection (see Figure 5.1a). This architecture has proved to be of low computational complexity.



(a) CBOW (b) Skip-gram

Figure 5.1: Neural network model architectures. Previous word is denoted as $w(t-1)$, current word is $w(t)$ and next word is $w(t+1)$.

5.4 Skip-Gram

The Skip-gram architecture is analogous to CBOW. However, instead of predicting the current word based on the context, it tries to predict a word's context based on the word itself [Mikolov et al., 2013b]. Thus, the intention of the Skip-gram model is to find word patterns that are useful for predicting the surrounding words within a certain range in a sentence (see Figure 5.1b). The Skip-gram model estimates the syntactic properties of words slightly worse than does the CBOW model, but it is much better at modeling their semantics [Mikolov et al., 2013a].

5.5 GloVe

The Global Vectors (GloVe) [Pennington et al., 2014] model focuses more on the global statistics of the trained data. This approach uses log-bilinear regression models that effectively capture the global statistics and word analogies. The authors propose a weighted least squares regression model that is trained by the global co-occurrence counts. The main concept of this model is the observation that the ratios of the co-occurrence probabilities have the potential for encoding the meanings of the words.

5.6 LDA

Latent Dirichlet Allocation (LDA) [Blei et al., 2003] is a well known topic model. LDA is based on the *Distributional Hypothesis* and the *Bag-of-words Hypothesis*, i.e. that the word order does not matter and there is some latent relation between the words within the same document (within the same context).

The underlying idea is that document is a mixture of topics and topic is a mixture of words. The meaning of words can be represented by the associated topic distribution as word vectors. The model can be extended to jointly model topic and sentiment [Lin and He, 2009].

6 Related Work

There are many ways to categorize sentiment analysis approaches e.g. by machine learning methods, by used resources, or by the granularity level of sentiment analysis.

Whereas dictionary-based methods usually depend on a sentiment dictionary (or a polarity lexicon) and a set of handcrafted rules [Taboada et al., 2011], machine learning-based methods require labeled training data that are later represented as features (see Section 4) and fed into a classifier (see Section 3). Later attempts have also investigated semi-supervised methods that incorporate unlabeled data [Zhang et al., 2012].

All of these are plausible, although in some cases it is difficult to determine the correct category. However, the granularity level of sentiment analysis seems to be the most natural way to categorize the related work.

The most of the research in automatic sentiment analysis has been devoted to English. There were several attempts in other languages (e.g. [Banea et al., 2010, Ghorbel and Jacot, 2011, Vilares et al., 2015, Basile and Nissim, 2013]), but in this chapter we will focus only on Czech and English.

Although we have devoted substantial effort to clearly describe all methods in the following sections in detail, we would like to direct curious readers to in-depth surveys [Pang and Lee, 2008, Liu and Zhang, 2012, Tsytsarau and Palpanas, 2012, Martínez-Cámara et al., 2014, Liu, 2015] for additional information.

6.1 Document Level and Sentence Level

Pang et al. [2002] experimented with unigrams (presence of a certain word, frequencies of words), bigrams, POS tags, and adjectives on a movie review dataset. Martineau and Finin [2009] tested various weighting schemes for

unigrams based on the TF-IDF model and proposed delta weighting for a binary scenario (positive, negative). Their approach was later extended by Paltoglou and Thelwall [2010] who proposed further improvements in delta TF-IDF weighting achieving the accuracy of 96.9% on the movie review dataset and 85.0% on the BLOG06 dataset.

The focus of current sentiment analysis research is shifting towards social media, mainly targeting Twitter [Kouloumpis et al., 2011, Pak and Paroubek, 2010] and Facebook [Go et al., 2009, Ahkter and Soria, 2010, Zhang et al., 2011, López et al., 2012]. Analyzing media with a very informal language benefits from involving novel features, such as emoticons [Pak and Paroubek, 2010, Montejo-Ráez et al., 2012], character n-grams [Blamey et al., 2012], POS and POS ratio [Ahkter and Soria, 2010, Kouloumpis et al., 2011], or word shape [Go et al., 2009, Agarwal et al., 2011].

In many cases, the gold data for training and testing the classifiers are created semi-automatically [Kouloumpis et al., 2011, Go et al., 2009, Pak and Paroubek, 2010]. In the first step, random samples from a large dataset are drawn according to the presence of emoticons (usually positive and negative) and are then filtered manually. Although large high-quality collections can be created very quickly with this approach, it makes a strong assumption that every positive or negative post must contain an emoticon.

Balahur and Tanev [2012] performed experiments with Twitter posts as part of the CLEF 2012 RepLab[1]. They classified English and Spanish tweets with a small but precise lexicon, which also contained slang, combined with a set of rules that captured the manner in which sentiment is expressed in social media.

Balahur and Turchi [2012] studied the manner in which sentiment analysis can be done for French, German, and Spanish, using machine translation (MT). They employed three different MT systems (Google Translate, Bing Translator, and Moses [Koehn et al., 2007]) in order to obtain training and test data for the three target languages. Subsequently, they extracted features for a machine learning model. They additionally employed meta-classifiers to test the possibility to minimize the impact of noise (incorrect translations) in the obtained data. Their experiments showed that training data obtained using machine translation do not significantly decrease performance of sentiment analysis and thus it can be a solution in the case of unavailability of the target language annotated data.

[1] <http://www.limosine-project.eu/events/replab2012>

Kiritchenko et al. [2014b] and Zhu et al. [2014] described a state-of-the-art sentiment analysis system that detects the sentiment of short informal textual messages (tweets and SMS messages) and the sentiment of terms. Their supervised system is based on a machine learning approach leveraging a variety of features. They employed automatically generated lexicons using tweets with sentiment-word hashtags and tweets with emoticons. Separate sentiment lexicon captured negated words. The system ranked first in the SemEval-2013 shared task *"Sentiment Analysis in Twitter"* (Task 2), obtaining an F-measure of 69.0% in the message-level task and 88.9% in the term-level task. Additional improvements boosted the F-measure to 70.5% (message-level task) and 89.5% (term-level task).

6.1.1 Neural Networks for Sentiment Analysis

First attempt to estimate sentiment using a neural network was presented in [Zhou et al., 2010]. The authors propose using Active Deep Networks which is a semi-supervised algorithm. The network is based on Restricted Boltzmann Machines. The approach is evaluated on several review datasets containing an earlier version of the movie review dataset created by Pang et al. [2002]. It outperforms the state-of-the-art approaches on these datasets.

Ghiassi et al. [2013] use Dynamic Artificial Network for sentiment analysis of Tweets. The network uses n-gram features and creates a Twitter-specific lexicon. The approach is compared to Support Vector Machines classifier and achieves better results.

Socher et al. [2013] presented their *Recursive Neural Tensor Network* and *Stanford Sentiment Treebank* (SST). The SST contained five sentiment labels (very positive to very negative) for 215,154 phrases in the parse trees of 11,855 sentences. They trained the *Recursive Neural Tensor Network* on the new treebank and evaluated against the state-of-the-art methods. This model outperformed all previous methods on several metrics and pushed the state of the art in binary sentence-level classification on the Rotten Tomatoes (RT) dataset from 80% up to 85.4%. The accuracy of predicting the five sentiment labels for all phrases reached 80.7%, an improvement of 9.7% over bag of words baselines. This is due to the fact that the model accurately captured sentence composition and the effects of negation at various tree levels for both positive and negative phrases.

A Deep Convolutional Neural Network is utilized for sentiment classification in [dos Santos and Gatti, 2014]. Classification accuracies of 48.3% (5

sentiment levels) and 85.7% (binary) on the the SST dataset are achieved.

Several papers propose more general neural networks used for NLP tasks that are tested also on sentiment datasets. One of such methods is presented in [Kim, 2014]. A Convolutional Neural Network (CNN) architecture is proposed and tested on several datasets such as Movie Review (MR) dataset and SST. The tasks were sentiment classification (binary or 5 sentiment levels), subjectivity classification (subjective/objective) and question type classification. It proved state-of-the-art performance on all datasets.

Kalchbrenner et al. [2014] proposed a Dynamic Convolutional Neural Network. A concept of dynamic k-max pooling is used in this network. It is tested on sentiment analysis and question classification tasks.

Zhang and LeCun [2015] propose two CNNs for ontology classification, sentiment analysis and single-label document classification. Their networks are composed of 9 layers out of which 6 are convolutional layers and 3 fully-connected layers with different numbers of hidden units and frame sizes. They show that the proposed method significantly outperforms the baseline approaches (bag of words) on English and Chinese corpora.

6.2 Word Level

Identifying the semantic orientation of subjective terms[2] (words or phrases) is a fundamental task for sentiment lexicon generation. These sentiment or opinion lexicons are compiled in an automatic manner with an optional final human check. The task of identifying semantic word orientation is also called words polarity detection. There are publicly available resources containing sentiment polarity of words e.g. *General Inquirer*[3], *Dictionary of Affect of Language*[4], *WordNet-Affect*[5], or *SentiWordNet* [Baccianella et al., 2010]. These resources are mainly used for computing the sentence or document sentiment by dictionary methods or as features for machine learning methods. Another use is the generation of a domain specific lexicon.

Turney [2002] and Turney and Littman [2003] estimate the semantic orientation of words by computing the *Pointwise Mutual Information* (PMI) between the given word and paradigm words (e.g. good, bad, nice, nasty).

[2]Also called sentiment words, opinion words and polar words
[3]http://www.wjh.harvard.edu/ inquirer/
[4]http://www.hdcus.com/
[5]http://wndomains.fbk.eu/wnaffect.html

Another approach [Kamps et al., 2004] measures the synonym relation of words based on *WordNet*[6].

Another popular way of using *WordNet* obtains a list of sentiment words by an iterative process of expanding the initial set with synonyms and antonyms [Kim and Hovy, 2004, Hu and Liu, 2004]. Kim and Hovy [2004] determine the sentiment polarity of unknown words according to the relative count of their positive and negative synonyms.

Wiebe et al. [2005] and Wilson et al. [2005] create the *Multi-Perspective Question Answering* (MPQA) corpus containing 535 news articles from a wide variety of news sources and describe the overall annotation scheme. They also compile a subjectivity lexicon with tagged prior[7] polarity values of words.

Rao and Ravichandran [2009] treat the sentiment polarity detection as a semi-supervised label propagation problem in a graph, where nodes represent words and edges are the relations between words. They use *WordNet* and Open Office thesaurus and positive and negative seed sets.

As demonstrated by Fahrni and Klenner [2008] the polarity of words is domain specific and lexicon-based approaches have difficulty with some domains. Machine learning algorithms naturally adapt to the corpus domain by training. Statistical approach to lexicon generation adapts the lexicon to the target domain. Fahrni and Klenner [2008] propose to derive posterior polarities using the co-occurrence of adjectives to create a corpus-specific dictionary.

He et al. [2008] use *Information Retrieval* methods to build a dictionary by extracting frequent terms from the dataset. The sentiment polarity of each document is computed as a relevance score to a query composed of the top terms from this dictionary. Finally, the opinion relevance score is combined with the topic relevance score, providing a ranking of documents for topics.

Choi and Cardie [2008] determine the polarity of terms using a structural inference motivated by compositional semantics. Their experiments show that lexicon–based classification with compositional semantics can perform

[6]WordNet [Miller and Fellbaum, 1998] is a hierarchical lexical database containing nouns, verbs, adjectives, and adverbs grouped into synonym sets (synsets). The synsets are related by different types of relationships to other synsets.

[7]*"Prior polarity refers to the sentiment a term evokes in isolation, as opposed to the sentiment the term evokes within a particular surrounding context."* [Pang and Lee, 2008]

better than supervised learning methods that do not incorporate compositional semantics (accuracy of 89.7% vs. 89.1%), but a method that integrates compositional semantics into the learning process performs better than the previous approaches (90.7%). The results were achieved on the MPQA dataset. Later, they study the adaptability of lexicons to other domains using an integer linear programming approach [Choi and Cardie, 2009].

Xu et al. [2012] have developed an approach based on HAL (see Section 5.1) called *Sentiment Hyperspace Analogue to Language* (S-HAL). The semantic orientation of words is characterized by a specific vector space. These feature vectors were used to train a classifier to identify the sentiment polarity of terms.

Saif et al. [2014] adapt the social-media sentiment lexicon from [Thelwall et al., 2012] by extracting semantics of words to update prior sentiment strength in lexicon and apply it to three different Twitter datasets. They achieve an average improvement of 2.5% and 4.5% in terms of accuracy and F-measure respectively.

6.3 Aspect-Based Sentiment Analysis

Recently a lot of attention has been targeted on sentiment analysis at finer levels of granularity, namely, aspect-based sentiment analysis (ABSA). The goal of ABSA is to extract aspects and to estimate the sentiment associated with the given aspect [Liu, 2012]. For the task definition see Section 2.3.

6.3.1 Aspect Term Extraction

The basic approach to aspect extraction is finding frequent nouns and noun phrases [Liu et al., 2005, Blair-Goldensohn et al., 2008, Moghaddam and Ester, 2010, Long et al., 2010].

Sequential learning methods (e.g. *Hidden Markov Models* (HMM) [Rabiner, 2010] or *Conditional Random Fields* (CRF) [Lafferty et al., 2001]) can be applied to aspect extraction. This approach treats aspect extraction as a special case of the general information extraction problem.

Hu and Liu [2004] extract the most frequent features (noun or noun phrases) and then remove meaningless feature phrases and redundant single-word features. Wei et al. [2010] further prune the feature space using a list

of subjective (positive and negative) adjectives. Pavlopoulos and Androutsopoulos [2014] propose adding a pruning mechanism that uses continuous space vector representations of words and phrases to further improve the results.

Another widely used approach to this problem is the use of topic models. Brody and Elhadad [2010] present a system that uses local (sentence-level) LDA (see Section 5.6) to discover aspect terms (nouns). Observing that every opinion has a target, a joint model can be designed to model the sentiment of words and topics at the same time [Xianghua et al., 2013, Mei et al., 2007, Titov and McDonald, 2008]. A topic-based model for jointly identifying aspect and sentiment words was proposed by Zheng et al. [2014].

6.3.2 Aspect Sentiment Polarity

Aspect sentiment polarity classification can be divided into lexicon-based approaches and machine learning approaches. Machine learning performs better in a particular application domain, however it is difficult to scale up to a large number of domains, thus lexicon-based techniques are more suitable for open-domain applications [Liu, 2012].

Lexicon-based approaches (e.g. [Xianghua et al., 2013, Ding et al., 2008, Hu and Liu, 2004]) use a list of aspect-related sentiment phrases as the core resource for aspect sentiment polarity classification.

Jiang et al. [2011] use a dependency parser to generate a set of aspect dependent features for classification. Boiy and Moens [2009] weights each feature based on the position of the feature relative to the target aspect in the parse tree.

Brody and Elhadad [2010] extract sentiment polarity from a constructed conjunction polarity graph.

Jo and Oh [2011] propose probabilistic generative models that outperform other generative models and are competitive in terms of accuracy with supervised aspect sentiment classification methods.

6.3.3 Semantic Evaluation Workshop

The current state of the art of aspect-based sentiment analysis methods for English was presented at the latest SemEval ABSA tasks [Pontiki et al., 2014, 2015, 2016]. For task definitions please refer to Section 2.3.

Semantic Evaluation Workshop SemEval 2014

We briefly describe the highest ranking systems of SemEval 2014 Task 4 [Pontiki et al., 2014]. The top results are shown in Table 6.1.

Kiritchenko et al. [2014a] (NRC-Canada) proposed a hybrid system that incorporates both machine learning n-gram features and automatically constructed sentiment lexicons for affirmative and negated contexts.

Brun et al. [2014] (XRCE) train one classifier to detect the categories and for each category they train a separate classifier for category detection of the corresponding polarities. They extend their previous system built on a robust deep syntactic parser which calculates semantic relations of words. The adaptation includes additional hand-written rules (regular expressions), extending dependency grammar and lexicons.

Castellucci et al. [2014] (UNITOR) exploit kernel methods within the SVM framework. They model the aspect term extraction task as a sequential tagging task by using implementation of structural SVMs for sequence tagging (SVM^{hmm}). The tasks of aspect term polarity detection, aspect category detection, and aspect category polarity detection are tackled as a classification problem where multiple kernels are linearly combined to generalize several linguistic information. Tree kernels proposed in [Collins and Duffy, 2001] are adapted to model syntactic similarity through convolutions among syntactic tree substructures.

Chernyshevich [2014] (IHS_RD) relies on a rich set of lexical, syntactic, and statistical features and the CRF model to correctly extract the aspect terms. She also runs a preprocessing step that performs e.g. slang and misspelling corrections, POS tagging, parsing, noun phrase extraction, semantic role labeling, and entity boundary detection.

Toh and Wang [2014] (DLIREC) ranked the first in the aspect term extraction task in the restaurant domain and second in the laptop domain. They use a CRF based classifier for aspect term extraction and linear classifier for aspect term polarity classification with lexicon, syntactic and semantic features. They created semantic clusters using Word2Vec [Mikolov et al., 2013c][8]

Wagner et al. [2014] (DCU) combine various lexicons such as MPQA, *SentiWordNet* and *General Inquirer* and use both rule-based and machine

[8]https://code.google.com/p/word2vec/

learning approach. They focus on fine tuning of parameters and the systems efficiency.

Brychcín et al. [2014] (UWB) present a system based on supervised machine learning extended by unsupervised methods for latent semantics discovery (LDA and semantic spaces - HAL and COALS see Section 5) and sentiment vocabularies. Their approach to aspect term extraction is based on CRF.

		Aspect detection					Aspect polarity		
		Const.	Team	P[%]	R[%]	F_1[%]	Const.	Team	ACC[%]
Restaurants	Term	U	DLIREC	85.35	82.71	84.01	C	DCU	80.95
		C	XRCE	86.25	81.83	83.98	SC	NRC-Can.	80.16
		C	NRC-Can.	84.41	76.37	80.19	U	UWB	77.69
		C	UNITOR	82.45	77.87	80.09	C	XRCE	77.69
	Category	C	NRC-Can.	91.04	86.24	88.58	C	NRC-Can.	82.92
		U	UNITOR	84.98	85.56	85.27	C	XRCE	78.15
		C	XRCE	83.23	81.37	82.29	U	UNITOR	76.29
		U	UWB	84.36	78.93	81.55	C	SAP_RI	75.61
Laptops	Term	SC	IHS_RD	84.80	66.51	74.55	C	DCU	70.49
		U	DLIREC	81.90	67.13	73.78	C	NRC-Can.	70.49
		C	DLIREC	79.31	63.30	70.41	C	SZTE-NLP	66.97
		C	NRC-Can.	78.77	60.70	68.57	C	UBham	66.66

Table 6.1: Comparison of the four best participating systems in each subtask. (SC) indicates a strongly constrained system that was not trained on the in-domain training data, (C) constrained system that was trained on the in-domain training data and (U) unconstrained system. ACC, P, R, and F_1 denote accuracy, precision, recall and F-measure, respectively.

Semantic Evaluation Workshop SemEval 2015

We briefly introduce the top ranking systems of SemEval 2015 [Pontiki et al., 2015]. Table 6.2 shows their results.

Toh and Su [2015] (NLANGP) modeled aspect category extraction as a multi-class classification problem and used features based on n-grams, head words (from parse trees), and word clusters learnt from Amazon and Yelp data. Target extraction was done using a CRF model with features based on n-grams, head words, learned target dictionaries, and Brown clusters. Category & Target subtask system was a simple combination of both outputs.

Task	Laptops			Restaurants			Hotels		
Category	NLANGP	#1	50.86*	NLANGP	#1	62.68*			
	Sentiue	#2	50.00*	Sentiue	#4	54.10*			
Target				EliXa	#1	70.05*			
				NLANGP	#2	67.11*			
				Lsislif	#4	62.22			
C&T				NLANGP	#1	42.90*			
Sentiment	Sentiue	#1	79.34*	Sentiue	#1	78.69*	Lsislif	#1	85.84
	ECNU	#2	78.29	ECNU	#2	78.10*	EliXa	#3	79.64*
	Lsislif	#3	77.87	Lsislif	#3	75.50	Sentiue	#4	78.76*

Table 6.2: Comparison of top ranking teams in the ABSA task of SemEval 2015. F-measure is used as an evaluation measure for all tasks except Sentiment where we use accuracy. * indicates unconstrained system.

Saias [2015] (Sentiue) used a separate MaxEnt classifier with n-grams and lemmas for each entity and for each attribute. Then they determine which categories will be assigned to each sentence based on category probabilities for given domain. For sentiment polarity classification they use a MaxEnt classifier with bag of words, lemmas, bigrams after verbs, presence of polarized terms, and punctuation, negation words and sentiment lexicons (AFINN, Bing Liu's lexicon, MPQA). They trained a single classifier for all three domains on all available training data (restaurants and laptops).

San Vicente et al. [2015] (EliXa) addressed the problem of opinion target extraction by using an averaged perceptron with a BIO tagging scheme. They used n-grams, word shape, word prefixes and suffixes, and word clusters (using additional data Yelp and Wikipedia) as features.

Zhang and Lan [2015] (ECNU) used SVM classifier and engineered features for the sentiment polarity classification task. The features include n-grams, pointwise mutual information (PMI) scores, POS tags, parse trees, negation words and scores based on 7 sentiment lexicons.

Hamdan et al. [2015] (Lsislif) relied on a logistic regression model with a weighting schema of positive and negative labels and various features for the sentiment polarity classification task. The features include n-grams, lexicons, category, negations, a term importance score (Z score), and word clusters. For the hotel domain they used a model trained on the restaurant domain.

SemEval-2015 Task 10B [Rosenthal et al., 2015] Sentiment analysis in

Twitter is a re-run of previous years (SemEval-2013 Task 2 and SemEval-2014 Task 9). The goal of this task is to classify Twitter messages (tweets) into positive, negative, or neutral sentiment classes. Teams evaluate their results on five datasets from previous years and on two new datasets. We mention only this task because in Chapter 11 we use the data from this task.

Astudillo et al. [2015] treat sentiment analysis as a regression problem which allows more fine-grained sentiment assessment. They model tweets using Word2Vec or GloVe embeddings that are averaged or summed over the given tweet. A regression model is than trained on the resulting representations. This model achieved the fourth place in the SemEval-2015 Task 10B.

Semantic Evaluation Workshop SemEval 2016

The latest SemEval ABSA task [Pontiki et al., 2016] provided 20 testing datasets from 7 domains and 8 languages, attracting 245 submissions from 29 teams. We briefly introduce some of the top ranking systems. The top ranking teams on English are in Table 6.3.

Toh and Su [2016] (NLANGP) extend their previous submission to SemEval 2015 and use neural network output as additional features. For the opinion target extraction task they train RNN and use the probability output as additional features for the CRF model. Category detection is done as a set of binary classification problems for each category with enhanced features by the output of a CNN model.

Brun et al. [2016] (XRCE) use CRF with a window of 7 words and various features including POS tags, lemma, and detailed syntactic parser output. Category detection was done within a feedback ensemble method pipeline using Elastic Net regression model [Zou and Hastie, 2005].

Kumar et al. [2016] (IIT-TUDA) use SVM classifier for sentiment polarity classification along with unigrams, bigrams, induced sentiment lexicons, and category. The English induced lexicon consists of approximately 13k words and uses five seed lexicons.

Jiang et al. [2016](ECNU) extract various n-gram and character n-gram features, punctuation, negation, Word2Vec, and sentiment lexicon features. They extract and use as a feature 25 words with highest probability for each of 20 LDA topics. They use logistic regression for the sentiment polarity task.

Álvarez López et al. [2016] (GTI) filter unigrams and lemmas by POS tags and use them along with bigrams and POS tags as features for 12 binary SVM classifiers (one for each category) for the category detection task. They use a rule-based algorithm based on noun dictionaries to select the resulting categories. Target extraction is done by using a CRF model.

Hercig et al. [2016a] (UWB) We also participated in the latest SemEval ABSA task. For more information see Section 8.2.

Domain	Lang	Level	Category		Target		Cat. & Target		Sentiment	
Restaurants	EN	Sentence	NLANGP	#1	NLANGP	#1	NLANGP	#1	XRCE	#1
			XRCE	#6	UWB	#3	XRCE	#2	IIT-TUDA	#2
			UWB	#7	GTI	#4	UWB	#5	ECNU	#5
Laptops	EN	Sentence	NLANGP	#1					IIT-TUDA	#1
			UWB	#5					ECNU	#3
Restaurants	EN	Text	GTI	#1					UWB	#1
			UWB	#2					ECNU	#2
Laptops	EN	Text	UWB	#1					UWB	#1 - 2
									ECNU	#1 - 2

Table 6.3: Top ranking teams in the ABSA task of SemEval 2016 on English.

6.4 Sentiment Analysis in Czech

Veselovská et al. [2012] presented an initial research on Czech sentiment analysis. They created a corpus which contains polarity categories of 410 news sentences. They used the Naive Bayes classifier and a classifier based on a lexicon generated from annotated data. The corpus is not publicly available, and because of its small size no strong conclusions can be drawn. Error analysis of lexicon-based classifier on this dataset was done by Veselovská and Hajič jr. [2013].

Subjectivity Lexicon for Czech [Veselovská, 2013, Veselovská et al., 2014] consists of 4947 evaluative expressions annotated with part of speech and tagged with positive or negative sentiment polarity. Although the lexicon did not significantly help to improve the polarity classification it is still a lexical resource worth mentioning.

Steinberger et al. [2012] proposed a semi-automatic *"triangulation"* approach to creating sentiment dictionaries in many languages, including Czech. They first produced high-level gold-standard sentiment dictionaries for two languages and then translated them automatically into a third language by means of a state-of-the-art machine translation service. Finally, the resulting

sentiment dictionaries were merged using the overlap of the two automatic translations.

A multilingual parallel news corpus annotated with opinions on entities was presented in [Steinberger et al., 2011]. Sentiment annotations were projected from one language to several others, which saved annotation time and guaranteed comparability of opinion mining evaluation results across languages. The corpus consists of 1,274 news sentences. It contains seven languages including Czech.

The first extensive evaluation of Czech sentiment analysis was done by Habernal et al. [2013]. Three different classifiers, namely Naive Bayes, SVM, and Maximum Entropy classifiers were tested on large-scale labeled corpora (10k Facebook posts, 90k movie reviews, and 130k product reviews). Habernal et al. [2014] further experimented with feature selection methods. For more information see Chapter 7.

Habernal and Brychcín [2013] used semantic spaces (see [Brychcín and Konopík, 2014]) created from unlabeled data as an additional source of information to improve results. Brychcín and Habernal [2013] explored the benefits of the global target context and outperformed the previous unsupervised approach.

The first attempt at aspect-based sentiment analysis in Czech was presented in [Steinberger et al., 2014]. This work provides an annotated corpus of 1244 sentences from the restaurant reviews domain and a baseline model achieving 68.7% F-measure in aspect term extraction, 74.0% F-measure on aspect category extraction, 66.3% accuracy in aspect term polarity classification, and 66.6% accuracy in aspect category polarity classification.

Tamchyna et al. [2015] created a dataset in the domain of IT product reviews. The dataset contains 200 sentences and 2000 short segments, both annotated with sentiment and marked aspect terms (targets) without any categorization and sentiment toward the marked targets. Using 5-fold cross validation on the aspect term extraction task they achieved 65.8% F-measure on the short segments and 30.3% F-measure on the long segments. Prior experiments with Conditional Random Fields (CRF) were done in [Veselovská, 2015] achieving 64.1% F-measure on the short segments.

6.5 Sarcasm Detection

The issue of automatic sarcasm detection has been addressed mostly in English, although there has been some research in other languages, such as Dutch [Liebrecht et al., 2013], Italian [Bosco et al., 2013], or Brazilian Portuguese [Vanin et al., 2013].

Experiments with semi-supervised sarcasm identification on a Twitter dataset (5.9 million tweets) and on 66,000 product reviews from Amazon were conducted in [Davidov et al., 2010, Tsur et al., 2010]. They used 5-fold cross validation on their kNN-like classifier and obtained an F-measure of 0.83 on the product reviews dataset and 0.55 on the Twitter dataset. For acquiring the Twitter dataset they used hashtag `#sarcasm` as an indicator of sarcastic tweets. They further created a balanced evaluation set of 180 tweets using 15 human annotators via Amazon Mechanical Turk[9] and achieved an inter-annotator agreement 0.41 (Fleiss' κ).

González-Ibáñez et al. [2011] experimented with Twitter data divided into three categories (sarcastic, positive sentiment and negative sentiment), each containing 900 tweets. They used the `#sarcasm` and `#sarcastic` hashtags to identify sarcastic tweets. They used two classifiers – SVM with sequential minimal optimization (SMO) and logistic regression. They tried various combinations of unigrams, dictionary-based features and pragmatic factors (positive and negative emoticons and user references), achieving the best result (accuracy 0.65) for sarcastic and non-sarcastic classification with the combination of SVM with SMO and unigrams. They employed 3 human judges to annotate 180 tweets (90 sarcastic and 90 non-sarcastic). The human judges achieved Fleiss' $\kappa = 0.586$, demonstrating the difficulty of sarcasm classification. Another experiment included 50 sarcastic and 50 non-sarcastic (25 positive, 25 negative) tweets with emoticons annotated by two judges. The automatic classification and human judges achieved the accuracy of 0.71 and 0.89 respectively. The inter-annotator agreement (Cohen's κ) was 0.74.

Reyes et al. [2012] proposed features to capture properties of a figurative language such as ambiguity, polarity, unexpectedness and emotional scenarios. Their corpus consists of five categories (humor, irony, politics, technology and general), each containing 10,000 tweets. The best result in the classification of irony and general tweets was F-measure 0.65.

Reyes et al. [2013] explored the representativeness and relevance of con-

[9] `<http://www.mturk.com>`

ceptual features (signatures, unexpectedness, style and emotional scenarios). These features include punctuation marks, emoticons, quotes, capitalized words, lexicon-based features, character n-grams, skip-grams [Guthrie et al., 2006], and polarity skip-grams. Their corpus consists of four categories (irony, humor, education and politics), each containing 10,000 tweets. Their evaluation was performed on two distributional scenarios, balanced distribution and imbalanced distribution (25% ironic tweets and 75% tweets from all three non-ironic categories) using the Naive Bayes and decision trees algorithms from the Weka toolkit [Witten and Frank, 2005]. The classification by the decision trees achieved an F-measure of 0.72 on the balanced distribution and an F-measure of 0.53 on the imbalanced distribution.

The work of Riloff et al. [2013] identifies one type of sarcasm: contrast between a positive sentiment and negative situation. They used a bootstrapping algorithm to acquire lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. They proposed a method which classifies tweets as sarcastic if it contains a positive predicative that precedes a negative situation phrase in close proximity. Their evaluation on a human-annotated dataset[10] of 3000 tweets (23% sarcastic) was done using the SVM classifier with unigrams and bigrams as features, achieving an F-measure of 0.48. The hybrid approach that combines the results of the SVM classifier and their contrast method achieved an F-measure of 0.51.

Sarcasm and nastiness classification in online dialogues was also explored in [Lukin and Walker, 2013] using bootstrapping, syntactic patterns and a high precision classifier. They achieved an F-measure of 0.57 on their sarcasm dataset.

Maynard and Greenwood [2014] performed experiments with a rule-based approach to sarcasm detection and sentiment analysis. They manually annotated 266 sentences from 134 collected tweets. Their corpus contains 68 opinionated sentences (62 negative, 6 positive), out of these 61 were deemed to be sarcastic. Their regular sentiment polarity analyser achieved 0.27 accuracy while the sentiment polarity analyser considering sarcasm achieved 0.77 accuracy using hand-crafted rules and lexicons. However this dataset is imbalanced and very small to draw any conclusions.

[10]They used three annotators. Each annotator was given the same 100 tweets with the sarcasm hashtag and 100 tweets without the sarcasm hashtag (the hashtags were removed). On these tweets the pairwise inter-annotator scores were computed (Cohen's Kappa $\kappa_1 = 0.80, \kappa_2 = 0.81$ and $\kappa_3 = 0.82$). Then each annotator labeled additional 1000 tweets.

Second experiment measured the accuracy of sarcasm and polarity detection. The corpus consists of 400 tweets (91 sarcastic sentences). Regrettably, the previous regular vs. sarcasm analyser comparison exploring the impact of sarcasm on polarity detection is not included. They only measured the performance of the sarcastic analyser.

6.5.1 SemEval Workshop

The goal of **SemEval-2015 Task 11** was to perform fine-grained sentiment analysis over texts containing figurative language. Ghosh et al. [2015] have created a dataset of figurative tweets using Twitter4j API and a set of hashtag queries (*#sarcasm*, *#sarcastic*, *#irony* and words such as *figuratively*). The dataset has been annotated for sentiment analysis on a fine-grained 11-point scale (-5 to 5, including 0). Evaluation measures for this task were mean squared error (MSE) and cosine similarity, both with penalization for not giving scores for all tweets.

CLaC [Özdemir and Bergler, 2015b] presented the best result for SemEval-2015 Task 11 using decision tree regression M5P [Wang and Witten, 1997]. They combined various lexicons with negation and modality scopes. They also participated in SemEval-2015 Task 10B achieving ninth place. Özdemir and Bergler [2015a] performed a comprehensive ablation study of features.

Both CPH and PRHLT teams did not use lexicons and therefore provide comparable baselines to models with no additional resources. CPH [McGillion et al., 2015] used ensemble methods and ridge regression. PRHLT [Gupta and Gómez, 2015] used ensembles of extremely random trees with character n-grams.

Sulis et al. [2016] analyse the corpus from Semeval-2015 Task 11 in terms of hashtags (*#irony*, *#sarcasm*, and *#not*) and confirm that messages using figurative language mostly express a negative sentiment. They experimented with binary classification (separation) of tweets with these hashtags.

6.5.2 Neural Networks for Sarcasm Detection

A neural network model for sarcasm detection is proposed in [Ghosh and Veale, 2016]. The model is composed from a CNN followed by a long short term memory (LSTM) network. First a CNN is applied to the input. LSTM is then applied directly on the output of the convolutional layer. Output of

the LSTM is fed to a fully connected layer and a softmax layer determines the class. F-score of 0.92 is achieved on their dataset containing 39k tweets.

Another approach is presented in [Zhang et al., 2016]. A deep neural network is used for tweet sarcasm detection. The network has two components for local and contextual (history) tweets. The local one is a bi-directional gated recurrent unit that extracts dense real-valued output. The other component applies a pooling layer directly to the word embeddings for words in the contextual tweets and maps it to a fixed length vector. A hidden layer then combines these two components and is followed by a softmax layer. Embeddings are initialized using GloVe. Results are compared with manually created features.

CNNs are utilized for feature extraction in [Poria et al., 2016]. Sentiment, emotion, and personality features are utilized for sarcasm detection. CNN models are separately trained on datasets corresponding to the three types of features. The three CNNs are then merged. The final classification is done either using a support vector machines classifier or another CNN which uses the merged features as a static channel and connects it to the penultimate layer before the softmax layer.

Part III

Research Contributions

7 Document-Level Sentiment Analysis

This chapter describes our in-depth research on machine learning methods for sentiment analysis of Czech social media [Habernal et al., 2013, 2014].

Automatic sentiment analysis in the Czech environment has not yet been thoroughly targeted by the research community. Therefore it is necessary to create a publicly available labeled dataset as well as to evaluate the current state of the art.

This chapter focuses on the document-level[1] sentiment analysis performed on three different Czech datasets using supervised machine learning.

7.1 Datasets

7.1.1 Social Media Dataset

The initial selection of Facebook brand pages for our dataset was based on the *"top"* Czech pages, according to the statistics from SocialBakers[2]. We focused on pages with a large Czech fan base and a sufficient number of Czech posts. Using Facebook Graph API and Java Language Detector[3] we acquired 10,000 random posts in the Czech language from nine different Facebook pages. The posts were then completely anonymized as we kept only their textual contents.

Sentiment analysis of posts at Facebook brand pages usually serves as marketing feedback on user opinions about brands, services, products, or current campaigns. Thus we consider the sentiment target to be the given product, brand, etc. Typically, users' complaints constitute negative sentiment, whereas joy or happiness about the brand is treated as positive.

[1] Or *post-level*, as documents correspond to *posts* in social media.

[2] <http://www.socialbakers.com/facebook-pages/brands/czech-republic/>

[3] <http://code.google.com/p/jlangdetect/>

We also added another class called *bipolar* which represents both positive and negative sentiment in one post.[4] In some cases, the user's opinion, although positive, does not relate to the given page.[5] Therefore the sentiment is treated as neutral in these cases, in accordance with our above-mentioned assumption.

The complete 10k dataset was independently annotated by two annotators. The inter-annotator agreement (Cohen's κ) reached 0.66 which represents a substantial agreement level [Pustejovsky and Stubbs, 2013], and therefore the task can be considered as well-defined.

The gold data were based on the agreement of the two annotators. They disagreed in 2,216 cases. To solve these conflicts, we involved a third super-annotator to assign the final sentiment label. Even after the third annotator's labeling, however, there was still no agreement for 308 labels. These cases were later solved by a fourth annotator. We discovered that most of these conflicting cases were classified as either neutral or bipolar. These posts were often difficult to label because the author used irony, sarcasm or the context of previous posts. These issues remain open.

The Facebook dataset contains 2,587 positive, 5,174 neutral, 1,991 negative, and 248 bipolar posts, respectively. We ignore the bipolar class later in all experiments. The sentiment distribution among the source pages is shown in Figure 7.1. The statistics reveal negative opinions towards cell phone operators[6] and positive opinions towards e.g. perfume sellers[7] and Prague Zoo[8].

7.1.2 Movie Review Dataset

Movie reviews as a corpus for sentiment analysis have been used in research since the pioneering research conducted by Pang et al. [2002]. Therefore we covered the same domain in our experiments as well. We downloaded 91,381 movie reviews from the Czech Movie Database[9] and split them into three

[4] For example *"to bylo moc dobry ,fakt jsem se nadlabla :-D skoda ze uz neni v nabidce"*—*"It was very tasty, I really stuffed myself :-D sad it's not on the menu anymore"*.

[5] Certain campaigns ask the fans to e.g. write a poem—these posts are mostly positive (or funny, at least) but are irrelevant in terms of the desired task.

[6] <www.facebook.com/o2cz>, <www.facebook.com/TmobileCz>, <www.facebook.com/vodafoneCZ>

[7] <www.facebook.com/Xparfemy.cz>

[8] <www.facebook.com/zoopraha>

[9] <http://www.csfd.cz>

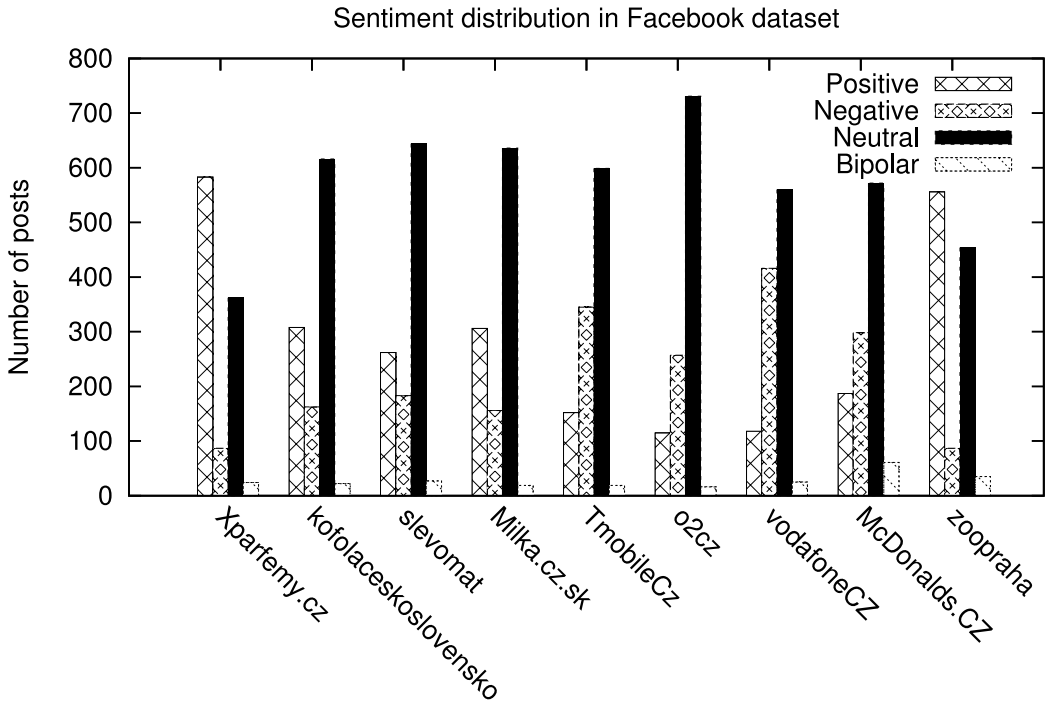


Figure 7.1: Social media dataset statistics

categories according to their star rating (0–2 stars as negative, 3–4 stars as neutral, 5–6 stars as positive). The dataset contains 30,897 positive, 30,768 neutral, and 29,716 negative reviews, respectively.

7.1.3 Product Review Dataset

Another very popular domain for sentiment analysis deals with product reviews [Hu and Liu, 2004]. We scraped all user reviews from a large Czech e-shop Mall.cz[10] which offers a wide range of products. The product reviews are accompanied by star ratings on a scale of zero to five. We took a different strategy for assigning sentiment labels. Whereas in the movie dataset the distribution of stars was rather uniform, in the product review domain the ratings were skewed towards the higher values. After a manual inspection we discovered that four-star ratings mostly correspond to neutral opinions and three or fewer stars denote mostly negative comments. Thus we split the dataset into three categories according to this observation. The final dataset consists of 145,307 posts (102,977 positive, 31,943 neutral, and 10,387 negative).

[10] <http://www.mall.cz>

7.2 Classification

7.2.1 Preprocessing

As pointed out by Laboreiro et al. [2010], tokenization significantly affects sentiment analysis, especially in the case of social media. Although Ark-tweet-nlp tool [Gimpel et al., 2011] was developed and tested in English, it yields satisfactory results in Czech as well, according to our initial experiments on the Facebook corpus. Its significant feature is proper handling of emoticons and other special character sequences that are typical of social media. We also remove stopwords using the stopword list from Apache Lucene[11].

In many NLP applications, a very popular preprocessing technique is stemming. We tested the Czech light stemmer [Dolamic and Savoy, 2009] and High Precision Stemmer[12]. Another widely-used method for reducing the vocabulary size, and thus the feature space, is lemmatization. For the Czech language the only currently available lemmatizer is shipped with the Prague Dependency Treebank (PDT) toolkit [Hajic et al., 2006]. We, however, used our in-house Java HMM-based implementation with the PDT training data as we needed better control over each preprocessing step. Following the work of Kanis and Skorkovská [2010], we developed an in-house lemmatizer using rules and dictionaries from the *OpenOffice* suite.

Part-of-speech tagging was done with our in-house Java solution that utilizes Prague Dependency Treebank (PDT) data as well. Since, however, PDT is trained on news corpora, we doubt it is suitable for tagging social media that are written in very informal language (see e.g. [Gimpel et al., 2011] where similar issues were tackled in English).

Since the Facebook dataset contains a huge number of grammar mistakes and misspellings (typically *'i/y'*, *'ě/je/ie'*, and others), we incorporated phonetic transcription to the International Phonetic Alphabet (IPA) in order to reduce the effect of these mistakes. We relied on eSpeak[13] implementation. Another preprocessing step might involve removing diacritics, as many Czech users type only unaccented characters. Posts without diacritics, however, represent only about 8% of our datasets, and thus we decided to keep diacritics intact.

[11] <http://lucene.apache.org/core/>
[12] <http://liks.fav.zcu.cz/HPS/>
[13] <http://espeak.sourceforge.net>

We were also interested in whether named entities (e.g. product names, brands, places, etc.) carry sentiment and how their presence influences classification accuracy. For these experiments, we employ a CRF-based named entity recognizer [Konkol and Konopík, 2013] and replace the words identified as entities with their respective entity type (e.g. *McDonald's* becomes *company*). This preprocessing has not been widely discussed in the literature devoted to document-level sentiment analysis, but Boiy and Moens [2009], for example, remove the *"entity of interest"* in their approach.

The complete preprocessing diagram and its variants is depicted in Table 7.1. Overall, there are 16 possible preprocessing *"pipe"* configurations.

Pipe 1	**Pipe 2**	**Pipe 3**
Tokenizing		
ArkTweetNLP		
POS tagging		
PDT		
Named entity filtering (N) [optional]		
remove (r)		
Stem (S)		**Lemma (L)**
none (n)		PDT (p)
light (l)		OpenOffice (o)
HPS (h)		
Stopwords		
remove		
Casing (C)	**Phonetic (P)**	–
keep (k)	eSpeak (e)	
lower (l)		

Table 7.1: The preprocessing pipes (top-down). Various combinations of methods can be denoted using the appropriate labels, e.g. "SnCk" means *1. tokenizing, 2. POS-tagging, 3. no stemming, 4. removing stopwords, and 5. no casing*, or "NrLp" means *1. tokenizing, 2. POS-tagging, 3. removing named entities, 4. lemmatization using PDT, and 5. removing stopwords.*

7.2.2 Features

N-gram features We use presence of unigrams and bigrams as binary features. The feature space is pruned by minimum n-gram occurrence empirically set to five. Note that this is the baseline feature in most of the related work.

Character n-gram features Similarly to the word n-gram features, we added character n-gram features, as proposed by e.g. [Blamey et al., 2012]. We set the minimum occurrence of a particular character n-gram to five, in order to prune the feature space. Our feature set contains 3-grams to 6-grams.

POS-related features Direct usage of part-of-speech n-grams that cover sentiment patterns has not shown any significant improvement in the related work. Still, POS tags provide certain characteristics of a particular post. We implemented various POS features that include e.g. the number of nouns, verbs, and adjectives [Ahkter and Soria, 2010], the ratio of nouns to adjectives and verbs to adverbs [Kouloumpis et al., 2011], and the number of negative verbs obtained from POS tags.

Emoticons We adapted the two lists of emoticons that were considered as positive and negative from [Montejo-Ráez et al., 2012]. The feature captures the number of occurrences of each class of emoticons within the text.

Delta TF-IDF variants for binary scenarios Although simple binary word features (presence of a certain word) achieve a surprisingly good performance, they have been surpassed by various TF-IDF-based weightings, such as Delta TF-IDF [Martineau and Finin, 2009], and Delta BM25 TF-IDF [Paltoglou and Thelwall, 2010]. Delta-TF-IDF still uses traditional TF-IDF word weighting but treats positive and negative documents differently. All the existing related works which use this kind of feature, however, deal only with binary decisions (positive/negative), and thus we filtered out neutral documents from the datasets.[14] We implemented the most promising weighting schemes from [Paltoglou and Thelwall, 2010], namely *Augmented TF*, *LogAve TF*, *BM25 TF*, *Delta Smoothed IDF*, *Delta Prob. IDF*, *Delta Smoothed Prob. IDF*, and *Delta BM25 IDF*.

7.2.3 Feature Selection

The basic reason for using feature selection (or reduction) methods for supervised sentiment analysis is twofold: first, the reduced feature set decreases the computing demands for the classifier, and, second, removing irrelevant features can lead to better classification accuracy.

[14] Opposite of leave-one-out cross-validation in [Paltoglou and Thelwall, 2010], we still use 10-fold cross-validation in all experiments.

Feature selection methods assign a certain weight to each feature, depending on its significance (discriminative power) for each class. After the weights are obtained, the top k features can be kept for the classifier, or the features with low weight can be cut off at a certain threshold.

Let t_k and $\bar{t_k}$ denote the presence and absence, respectively, of a particular feature in a certain class (e.g. c_1, and c_2). To estimate the joint probability of a feature in a given class, we will use the following table with appropriate feature counts:

	c_1	c_2
t_k	a	b
$\bar{t_k}$	c	d

Then N denotes the total number of features in all classes, $N = a+b+c+d$. The joint probability $p(t_k, c_1)$ can then be estimated as

$$p(t_k, c_1) = \frac{a}{N}, \tag{7.1}$$

and similarly for $p(t_k, c_2)$. The probability of a particular feature in all classes $p(t_k)$ is given by

$$p(t_k) = \frac{a+b}{N}. \tag{7.2}$$

Furthermore, c_1 can be estimated as

$$p(c_1) = \frac{a+c}{N}. \tag{7.3}$$

The conditional probability of t_k given c_1 is given by

$$p(t_k|c_1) = \frac{a}{a+c}. \tag{7.4}$$

Henceforth, let n denote the number of classes, $m = \{t_k, \bar{t_k}\}$, and all logarithms are to the base 2.

We follow with the formulas for the particular feature selection methods. For a more detailed discussion of these methods, please refer to e.g. [Forman, 2003, Zheng et al., 2004, Uchyigit, 2012, Patočka, 2013].

Mutual Information (MI)

Mutual Information is always non-negative and symmetrical [Battiti, 1994], $MI(X,Y) = MI(Y,X)$.

$$MI = \sum_{i=0}^{n}\sum_{k=0}^{m} \log \frac{p(t_k, c_i)}{p(c_i)p(t_k)} \tag{7.5}$$

Information Gain (IG)

Also known as *Kullback-Leibler divergence* or *relative entropy*. It is a non-negative and asymmetrical metric.

$$IG = \sum_{i=0}^{n}\sum_{k=0}^{m} p(t_k, c_i) \log \frac{p(t_k, c_i)}{p(c_i)p(t_k)} + p(\bar{t_k}, c_i) \log \frac{p(\bar{t_k}, c_i)}{p(c_i)p(\bar{t_k})} \tag{7.6}$$

Chi Square (CHI)

Chi Square (χ^2) can be derived as follows.

$$GSS(t_k, c_i) = p(t_k, c_i)p(\bar{t_k}, \bar{c_i}) - p(t_k, \bar{c_i})p(t_k, \bar{c_i}), \tag{7.7}$$

$$NGL(t_k, c_i) = \frac{\sqrt{N} \cdot GSS(t_k, c_i)}{\sqrt{p(t_k)p(\bar{t_k})p(c_i)p(\bar{c_i})}}, \tag{7.8}$$

$$\chi^2 = \sum_{i=0}^{n}\sum_{k=0}^{m} NGL(t_k, c_i)^2 \tag{7.9}$$

Odds Ratio (OR)

$$OR = \sum_{i=0}^{n}\sum_{k=0}^{m} \log \frac{p(t_k|c_i)p(\bar{t_k}|\bar{c_i})}{p(\bar{t_k}|c_i)p(t_k|\bar{c_i})} \tag{7.10}$$

Relevancy Score (RS)

$$RS = \sum_{i=0}^{n}\sum_{k=0}^{m} \log \frac{p(t_k|c_i)}{p(\bar{t_k}|\bar{c_i})} \tag{7.11}$$

7.2.4 Classifiers

All evaluation tests were performed with two classifiers, Maximum Entropy (MaxEnt) and SVM. Although the Naive Bayes classifier is also widely used in related work, we did not include it as it usually performs worse than SVM or MaxEnt. We used a pure Java framework for machine learning[15] with default settings (the linear kernel for SVM).

7.3 Results

For each combination from the preprocessing pipeline (refer to Table 7.1) we assembled various sets of features and employed two classifiers. In the first scenario, we classified into all three classes (positive, negative, and neutral).[16] In the second scenario, we followed a strand of related research e.g. [Martineau and Finin, 2009, Celikyilmaz et al., 2010], that deals only with positive and negative classes. For these purposes we filtered out all the neutral documents from the datasets. Furthermore, in this scenario we evaluate only features based on weighted delta-TF-IDF, as e.g. in [Paltoglou and Thelwall, 2010]. We also involved only the MaxEnt classifier into the second scenario.

All tests were conducted in the 10-fold cross validation manner. We report the macro F-measure, as it allows comparison of classifier results on different datasets. Moreover, we do not report the micro F-measure (accuracy) as it tends to prefer performance on dominant classes in highly unbalanced datasets [Manning et al., 2008], which is e.g. the case of our Product Review dataset where most of the labels are positive.

7.3.1 Social Media

Table 7.2 shows the results for the three-class classification scenario on the Facebook dataset. The row labels denote the preprocessing configuration according to Table 7.1. In most cases, the Maximum Entropy classifier significantly outperforms SVM. The combination of all features (the last column) yields the best results regardless of the preprocessing steps. The reason might be that the character n-gram feature captures subtle sequences which represent subjective punctuation or emoticons, that were not covered by the *emoticon* feature. On average, the best results were obtained when HPS

[15] Later released as Brainy [Konkol, 2014].

[16] We ignore the bipolar posts in the current research.

stemmer and lower-casing or phonetic transcription were involved (lines *ShCl* and *ShPe*). This configuration significantly outperforms other preprocessing techniques for token-based features (see column FS4: *Unigr + bigr + POS + emot.*).

Facebook dataset, 3 classes

Feat. set	FS1		FS2		FS3		FS4		FS5	
Classifier	ME	SVM	ME	SVM	ME	SVM	ME	SVM	ME	SVM
SnCk	63	64	63	64	66	64	66	64	**69**	67
SnCl	63	64	63	64	66	63	66	63	**69**	68
SlCk	65	67	66	67	68	66	67	66	**69**	67
SlCl	65	67	65	67	68	66	**69**	66	**69**	67
ShCk	66	67	66	67	68	67	67	67	**69**	67
ShCl	66	66	66	67	**69**	67	**69**	67	**69**	67
SnPe	64	65	64	65	67	65	67	65	68	68
SlPe	65	67	65	67	68	67	67	66	68	67
ShPe	66	67	66	67	**69**	66	**69**	66	68	67
Lp	64	65	63	65	67	64	67	65	68	67
Lo	65	66	64	66	67	66	67	65	68	67

Table 7.2: Results for the Facebook dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 1$. Bold numbers denote the best results. **FS1**: unigrams; **FS2**: unigrams, bigrams; **FS3**: unigrams, bigrams, POS features; **FS4**: unigrams, bigrams, POS, emoticons; **FS5**: unigrams, bigrams, POS, emoticons, character n-grams.

In the second scenario we evaluated various TF-IDF weighting schemes for binary sentiment classification. The results are shown in Table 7.3. The three-character notation consists of term frequency, inverse document frequency, and normalization. Because of the large number of possible combinations, we report only the most successful ones, namely *Augmented*—*a* and *LogAve*—*L* term frequency, followed by *Delta Smoothed*—$\Delta(t')$, *Delta Smoothed Prob.*—$\Delta(p')$, and *Delta BM25*—$\Delta(k)$ inverse document frequency; normalization was not involved. We can see that the baseline (the first column *bnn*) is usually outperformed by any weighted TF-IDF technique. Moreover, using any kind of stemming (the row entitled *various**) significantly improves the results. For the exact formulas of the delta TF-IDF variants please refer to [Paltoglou and Thelwall, 2010].

We also tested the impact of TF-IDF word features when added to other features from the first scenario (refer to Table 7.2). Column *FS1* in Table 7.3 displays results for a feature set with the simple binary presence-of-the-word

Facebook dataset, positive and negative classes only

	bnn	$a\Delta(t')n$	$a\Delta(p')n$	$a\Delta(k)n$	$L\Delta(t')n$	$L\Delta(p')n$	$L\Delta(k)n$	FS1	FS2
SnCk	83	86	86	86	85	86	86	**90**	89
SnCl	84	86	86	86	86	86	86	**90**	**90**
various*	85	88	88	88	88	88	88	**90**	**90**
SnPe	84	86	86	86	86	86	86	**90**	**90**
Lp	84	86	85	85	86	86	86	88	88
Lo	84	88	87	87	87	87	87	**90**	**90**

* same results for ShCk, ShCl, SlCl, SlPe, SlCk, and ShPe
FS1: Unigr + bigr + POS + emot. + char n-grams
FS2: $a\Delta(t')n$ + bigr + POS + emot. + char n-grams

Table 7.3: Results for the Facebook dataset for various TF-IDF-weighted features, classification into two classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 1$. Underlined numbers show the best results for TF-IDF-weighted features. Bold numbers denote the best overall results.

feature (binary unigrams). In the last column *FS2* we replaced this binary feature with the TF-IDF-weighted feature $a\Delta(t')n$. It turned out that the weighted form of the word feature does not improve the performance, compared with the simple binary unigram feature. Furthermore, a set of different features (words, bigrams, POS, emoticons, character n-grams) significantly outperforms a single TF-IDF-weighted feature.

Furthermore, we report the effect of the dataset size on the performance. We randomly sampled 10 subsets from the dataset (1k, 2k, etc.) and tested the performance, still using 10-fold cross-validation. We took the most promising preprocessing configuration (*ShCl*) and MaxEnt classifier. As can be seen in Figure 7.2, while the dataset grows to approx 6k to 7k items, the performance rises for most combinations of features. With a 7k-item dataset, the performance begins to reach its limits for most combinations of features and hence adding more data does not lead to a significant improvement.

The influence of named entity filtering is shown in Table 7.4. In most cases, removing named entities leads to a significant drop in classification. Thus we can conclude that in our corpus, the named entities themselves represent an important opinion-holder. This also corresponds to the sentiment distribution as shown in Figure 7.1 (e.g. sentiment towards mobile phone

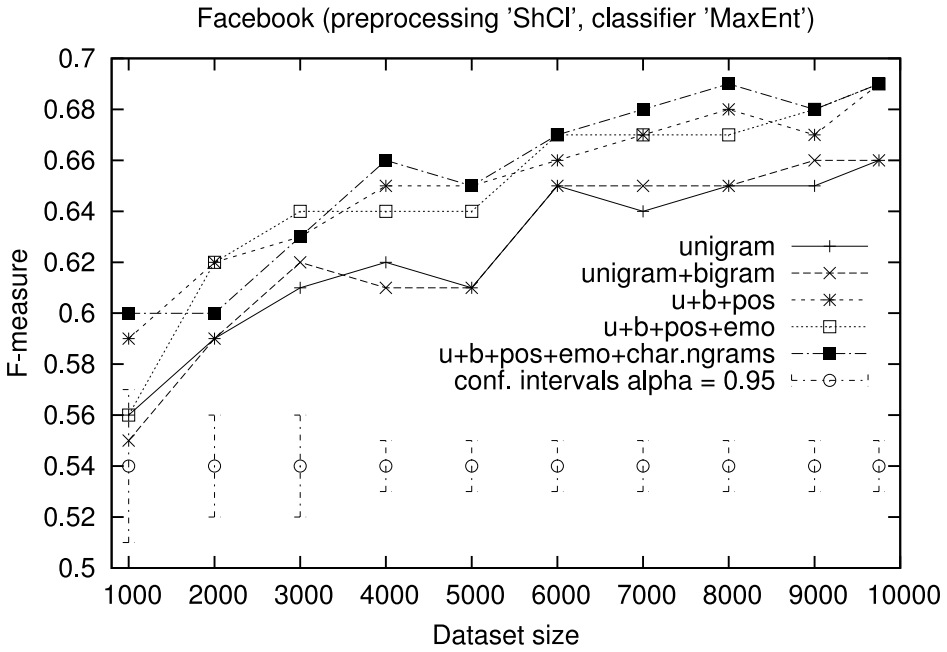


Figure 7.2: Learning curve; using *ShCl* preprocessing and MaxEnt classifier.

operators is rather negative) and thus by removing the brand name from the data the classifier loses useful information.

Upper Limits of Automatic Sentiment Analysis

To see the upper limits of the task itself, we also evaluate the annotator's judgments. Although the gold labels were chosen after a consensus of at least two people, there were many conflicting cases that had to be solved by a third or even a fourth person. Thus even the original annotators do not achieve a 1.00 F-measure on the gold data.

We present *"performance"* results of both annotators and of the best system as well (MaxEnt classifier, all features, *ShCl* preprocessing). Table 7.5 shows the results as confusion matrices. For each class (*p*—positive, *n*—negative, *0*—neutral) we also report precision, recall, and F-measure. The row headings denote gold labels; the column headings represent values assigned by the annotators or the system.[17] The annotators' results show what can be expected from a "perfect" system that solves the task the way a human would.

[17]Even though the task has three classes, the annotators also used *"b"* for *"bipolar"* and *"?"* for *"cannot decide"*.

Facebook dataset, 3 classes

Feat. set	FS1		FS2		FS3		FS4		FS5	
Classifier	ME	SVM	ME	SVM	ME	SVM	ME	SVM	ME	SVM
SlPe	65	67	65	67	68	67	67	66	68	67
NrSlPe	65	66	65	65	67	65	68	65	68	66
SlCl	65	67	65	67	68	66	**69**	66	**69**	67
NrSlCl	65	66	65	66	67	65	68	65	68	67
SlCk	65	67	66	67	68	66	67	66	**69**	67
NrSlCk	65	66	65	66	67	65	67	65	68	67
ShPe	66	67	66	67	**69**	66	**69**	66	68	67
NrShPe	65	66	65	66	67	65	68	65	67	66
ShCl	66	66	66	67	**69**	67	**69**	67	**69**	67
NrShCl	65	66	65	66	67	66	68	64	68	66

Table 7.4: Comparison of the five best (on average) preprocessing pipes with and without NER (Nr prefix). Results for the Facebook dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 1$. Bold numbers denote the best results. **FS1**: unigrams; **FS2**: unigrams, bigrams; **FS3**: unigrams, bigrams, POS features; **FS4**: unigrams, bigrams, POS, emoticons; **FS5**: unigrams, bigrams, POS, emoticons, character n-grams.

In general, both annotators judge all three classes with very similar F-measures. By contrast, the system's F-measure is very low for negative posts (0.54 vs. ≈ 0.75 for neutral and positive). We offer the following explanation. First, many of the negative posts surprisingly contain happy emoticons, which could be a misleading feature for the classifier. Second, the language of the negative posts in not as explicit as for the positive ones in many cases; the negativity is *"hidden"* in irony, or in a larger context (i.e. *"Now I'm sooo satisfied with your competitor :))"*). This remains an open issue for future research.

7.3.2 Product and Movie Reviews

For the other two datasets, the product reviews and movie reviews, we slightly changed the configuration. First, we removed the character n-grams from the feature sets, otherwise the feature space would become too large for feasible computing. Second, we abandoned SVM as it became computationally infeasible for such large datasets.

	Annotator 1							
	0	**n**	**p**	**?**	**b**	**P**	**R**	**Fm**
0	4867	136	115	2	54	93	94	93
n	199	1753	6	0	33	93	88	90
p	175	6	2376	0	30	95	92	93
						Macro Fm:		92
	Annotator 2							
	0	**n**	**p**	**?**	**b**	**P**	**R**	**Fm**
0	4095	495	573	3	8	95	79	86
n	105	1878	6	0	2	79	94	86
p	100	12	2468	3	4	81	95	.88
						Macro Fm:		86
	Best system							
	0	**n**	**p**			**P**	**R**	**Fm**
0	4014	670	490			74	78	76
n	866	1027	98			57	52	54
p	563	102	1922			77	74	75
						Macro Fm:		69

Table 7.5: Confusion matrices for three-class classification. *"Best system"* configuration: all features (unigram, bigram, POS, emoticons, character n-grams), *ShCl* preprocessing, and MaxEnt classifier. 95% confidence interval $\approx \pm 1$.

Table 7.6 (left-hand side) presents the results of the product reviews. The combination of unigrams and bigrams works best, almost regardless of the preprocessing. By contrast, POS features rapidly decrease the performance. We suspect that POS-related features do not carry any useful information in this case and also bring too much *"noise"* to the classifier.

In the right-hand side of Table 7.6 we can see the results of the movie reviews. Again, the bigram feature performs best, paired with a combination of HPS stemmer and phonetic transcription (*ShPe*). Adding POS-related features causes a large drop in performance. We can conclude that for larger texts, the bigram-based feature outperforms unigram features and, in some cases, a proper preprocessing may further significantly improve the results.

Table 7.7 shows the effect of replacing named entities by their types. Again, named entities (e.g. actors, directors, products, brands) are very strong opinion-holders and thus their filtering significantly decreases classification performance.

	Product reviews, 3 classes				Movie reviews, 3 classes			
	FS1	FS2	FS3	FS4	FS1	FS2	FS3	FS4
SnCk	69.90	74.00	52.41	49.02	75.94	77.02	70.72	61.44
SnCl	70.79	75.05	50.93	51.73	76.03	77.15	70.60	69.70
SlCk	66.87	**75.18**	58.52	55.49	77.92	**78.26**	73.25	72.09
SlCl	67.26	74.74	56.48	56.99	77.60	**78.35**	70.77	71.23
ShCk	66.90	74.68	57.39	56.91	77.82	78.23	73.80	71.59
ShCl	66.83	74.02	54.88	57.43	77.06	78.21	73.14	73.16
SnPe	69.42	74.20	50.01	55.46	76.59	77.67	69.27	72.50
SlPe	66.70	**75.23**	55.08	57.03	77.60	**78.26**	72.94	73.22
ShPe	67.54	73.38	56.22	59.47	77.62	**78.50**	73.86	72.68
Lp	65.60	74.68	56.18	56.68	76.94	77.01	67.87	69.80
Lo	68.11	**75.30**	52.83	54.03	76.17	77.37	72.93	72.04

Table 7.6: Results for the product and movie review datasets, classification into three classes. FSx denote different feature sets. **FS1** = Unigrams; **FS2** = Uni + bigrams; **FS3** = Uni + big + POS features; **FS4** = Uni + big + POS + emot. Macro F-measure (in %), 95% confidence interval $\approx \pm 0.2$ (products), $\approx \pm 0.3$ (movies). Bold numbers denote the best results.

7.3.3 Feature Selection Experiments

Using the two most promising preprocessing pipelines (*ShCl*, *ShPe*), we conducted experiments with feature selection methods as introduced in Section 7.2.3. We classify into three classes using both MaxEnt and SVM classifiers on the Facebook dataset and using only MaxEnt on the other datasets (because of computational feasibility, as mentioned previously in Section 7.3.2).

Feature selection methods assign a certain weight to each feature and cut off those features whose weight is under a certain threshold. To estimate an optimal parameter automatically, we measured how the feature weight threshold influences the performance. For this purposes we used held-out data (10% of the training data). In each fold of the 10-fold cross validation, the optimum threshold for feature cut-off was set such that the performance on the held-out data was maximized.

In the previous experiments (Section 7.3), the feature space was pruned by a minimum occurrence which was empirically set to five. This prior pruning is not necessary for automatic feature selection. Therefore, we removed this prior filtering for the experiments on the Facebook data.[18]

[18]For the other two datasets, the product reviews and movie reviews, we still kept the

	Product reviews, 3 classes				Movie reviews, 3 classes			
	FS1	FS2	FS3	FS4	FS1	FS2	FS3	FS4
SlCk	66.87	**75.18**	58.52	55.49	77.92	**78.26**	73.25	72.09
NrSlCk	66.54	72.57	50.39	56.66	75.91	75.98	67.84	70.47
SlCl	67.26	74.74	56.48	56.99	77.60	**78.35**	70.77	71.23
NrSlCl	66.54	72.57	50.39	52.36	75.91	75.98	67.84	70.99
ShCl	66.83	74.02	54.88	57.43	77.06	78.21	73.14	73.16
NrShCl	66.19	71.94	56.13	58.91	75.79	75.86	72.99	72.39
SlPe	66.70	**75.23**	55.08	57.03	77.60	**78.26**	72.94	73.22
NrSlPe	64.98	74.45	49.39	55.97	76.09	76.07	72.09	68.33
ShPe	67.54	73.38	56.22	59.47	77.62	**78.50**	73.86	72.68
NrShPe	66.60	74.33	55.00	56.10	76.15	76.26	70.88	71.62

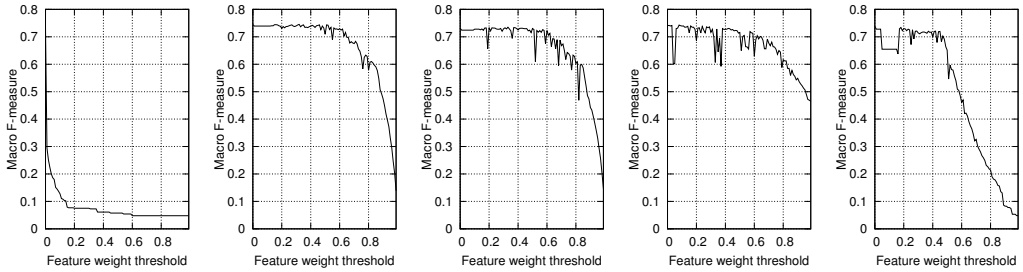
Table 7.7: Comparison of the five best (on average) preprocessing pipes with and without NER (Nr prefix). Results for the product and movie review datasets, classification into three classes. FSx denotes different feature sets. **FS1** = Unigrams; **FS2** = Uni + bigrams; **FS3** = Uni + big + POS features; **FS4** = Uni + big + POS + emot. Macro F-measure (in %), 95% confidence interval $\approx \pm 0.2$ (products), $\approx \pm 0.3$ (movies). Bold numbers denote the best results.

Figures 7.3, 7.4, 7.5, and 7.6 show dependency graphs of the macro F-measure given a feature weight threshold. Note that these figures depict parameter estimation for only one fold from the 10-fold cross-validation and thus serve only as an illustration of the feature selection behavior. It is apparent that *Information Gain* and *Mutual Information* are able to filter out noisy features to a large extent yet keep the performance almost unchanged. The worst selector is *Chi Square* as it drastically lowers the performance even with a very small filtering threshold.

Overall, a significant improvement from 73.38% (baseline) to 73.85% was achieved for the product reviews, by means of the *Mutual Information* feature selector and *ShPe* preprocessing pipeline (see Table 7.8). Yet very similar results were obtained with a different preprocessing pipeline (*ShCl*). For the movie reviews dataset (Table 7.10) and the Facebook dataset with and without feature space pruning (Tables 7.9, and 7.11, respectively) no significant improvement was achieved.

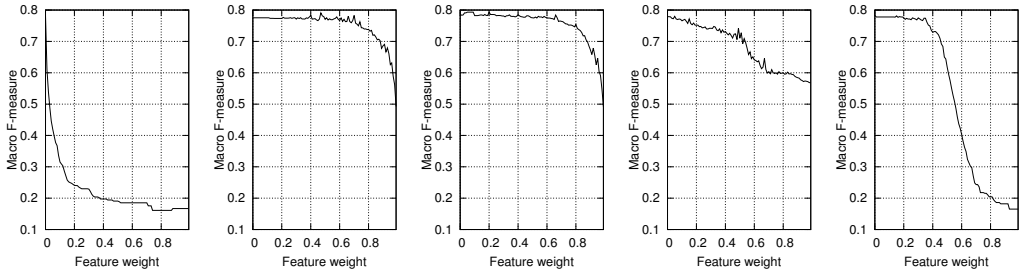
minimum occurrence set to five, as otherwise the feature space would become too large for feasible computing.

We can conclude that, in our settings, feature selection does not lead to a better overall performance, however, it can speed up the classification by filtering out noisy features.



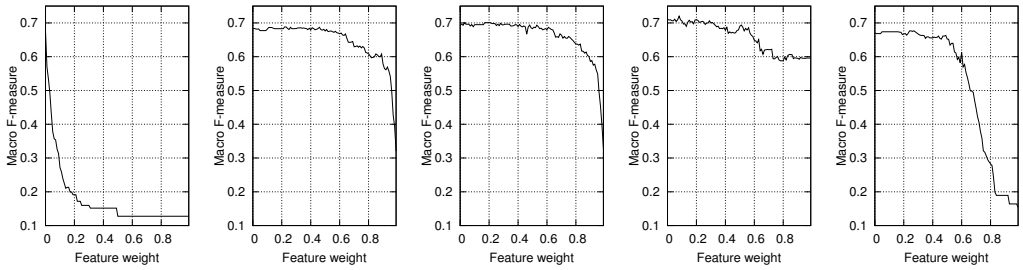
(a) Chi Square (b) Inf. Gain (c) Mutual Inf. (d) Odds Ratio (e) Rel. Score

Figure 7.3: Feature weight threshold estimation on heldout data.
Product reviews, *ShCl* preprocessing pipe, *MaxEnt* classifier, 3 classes, **FS2:** unigrams, bigrams



(a) Chi Square (b) Inf. Gain (c) Mutual Inf. (d) Odds Ratio (e) Rel. Score

Figure 7.4: Feature weight threshold estimation using heldout data.
Movie reviews, *ShCl* preprocessing pipe, *MaxEnt* classifier, three classes, **FS2:** unigrams, bigrams



(a) Chi Square (b) Inf. Gain (c) Mutual Inf. (d) Odds Ratio (e) Rel. Score

Figure 7.5: Feature weight threshold estimation using heldout data.
Facebook dataset, *ShCl* preprocessing pipe, *MaxEnt* classifier, three classes, no prior feature space pruning, **FS5:** unigrams, bigrams, POS, emoticons, character n-grams

Ft. selection	ChS		IG		MI		OR		RS		-	
Ft. set	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2
ShCl	65.93	73.50	66.28	**73.61**	66.16	73.56	65.50	72.74	66.03	73.40	66.83	**74.02**
ShPe	65.93	72.54	66.19	73.02	66.63	**73.85**	66.53	71.94	66.01	72.76	67.54	73.38

Table 7.8: Results for the product review dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 0.2$. Bold numbers denote the best results. **FS1**: Unigrams; **FS2**: unigrams, bigrams.

Feat.	Feat. set	FS1		FS2		FS3		FS4		FS5	
selection	Classifier	ME	SVM	ME	SVM	ME	SVM	ME	SVM	ME	SVM
ChS	ShCl	64.88	66.36	65.43	67.09	67.32	65.38	**68.51**	65.70	**68.63**	67.16
	ShPe	65.68	67.13	65.04	66.95	68.36	65.90	67.57	66.18	67.46	66.25
IG	ShCl	64.39	65.50	65.54	66.24	67.64	65.64	67.42	65.90	**68.56**	66.23
	ShPe	64.18	66.04	65.18	66.20	67.72	65.36	67.53	64.74	67.96	66.30
MI	ShCl	64.40	66.08	64.37	65.45	67.94	64.38	67.43	65.77	**68.73**	66.90
	ShPe	64.05	66.39	64.30	66.10	67.63	65.82	68.22	65.42	67.50	65.91
OR	ShCl	64.68	66.10	65.31	66.91	67.77	65.66	67.03	64.16	67.24	66.84
	ShPe	64.77	66.79	64.31	66.51	67.94	64.12	67.55	65.60	66.70	66.03
RS	ShCl	64.68	65.80	65.28	66.32	67.72	65.05	67.44	65.67	68.13	66.14
	ShPe	63.90	65.96	64.83	66.75	66.98	65.66	67.05	64.71	67.99	66.49
-	ShCl	65.69	66.26	65.73	66.89	**68.85**	66.75	**68.96**	67.06	**68.76**	66.71
	ShPe	65.74	66.76	65.66	66.95	**68.72**	65.81	**68.66**	66.05	68.45	66.57

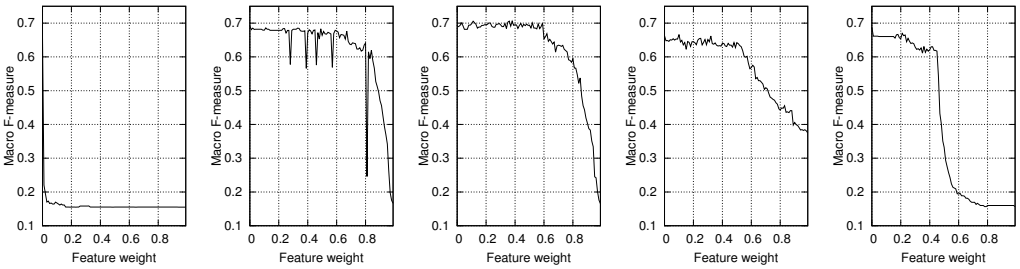
Table 7.9: Results for the Facebook dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 1$. Bold numbers denote the best results. **FS1**: Unigrams; **FS2**: unigrams, bigrams; **FS3**: unigrams, bigrams, POS features; **FS4**: unigrams, bigrams, POS, emoticons; **FS5**: unigrams, bigrams, POS, emoticons, character n-grams.

Feat. selection	ChS		IG		MI		OR		RS		-	
Feat. set	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2	FS1	FS2
ShCl	76.86	**78.46**	77.37	78.03	76.98	77.44	76.18	77.71	77.32	77.43	77.06	**78.21**
ShPe	77.43	77.83	76.64	77.85	76.77	77.62	75.89	78.06	77.25	77.38	77.62	**78.50**

Table 7.10: Results for the movie review dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 0.3$. Bold numbers denote the best results. **FS1**: Unigrams; **FS2**: unigrams, bigrams.

Feat.	Feat. set	FS1		FS2		FS3		FS4		FS5	
selection	Classifier	ME	SVM	ME	SVM	ME	SVM	ME	SVM	ME	SVM
ChS	ShCl	65.57	67.81	67.28	66.64	67.60	65.12	67.76	64.37	**69.04**	66.97
	ShPe	65.91	68.04	67.19	67.43	67.33	64.96	67.66	64.99	68.31	66.21
IG	ShCl	65.55	67.55	67.69	66.48	66.64	64.28	66.91	64.52	**69.01**	66.87
	ShPe	64.84	67.62	67.27	66.59	67.57	65.08	67.10	64.78	68.17	66.19
MI	ShCl	65.10	66.99	67.51	66.84	67.05	64.94	68.07	64.22	**69.35**	66.84
	ShPe	65.44	66.96	67.40	66.84	67.11	64.34	67.74	64.50	68.14	66.69
OR	ShCl	66.09	67.25	67.61	66.67	67.68	64.44	66.71	64.84	**69.27**	67.20
	ShPe	65.03	68.07	67.37	66.23	67.19	65.26	67.38	65.62	68.14	65.82
RS	ShCl	64.78	67.89	67.34	67.18	66.83	67.11	67.51	64.21	68.48	66.21
	ShPe	65.37	67.55	67.37	66.26	67.61	67.10	67.42	64.33	67.59	66.14
-	ShCl	65.21	68.55	67.85	66.97	68.10	65.48	68.33	65.01	**69.37**	67.46
	ShPe	66.09	68.27	67.65	67.44	68.11	65.94	67.27	65.90	68.57	66.90

Table 7.11: Results for the Facebook dataset, classification into three classes. Macro F-measure (in %), 95% confidence interval $\approx \pm 1$. Bold numbers denote the best results. **FS1**: Unigrams; **FS2**: unigrams, bigrams; **FS3**: unigrams, bigrams, POS features; **FS4**: unigrams, bigrams, POS, emoticons; **FS5**: unigrams, bigrams, POS, emoticons, character n-grams.



(a) Chi Square (b) Inf. Gain (c) Mutual Inf. (d) Odds Ratio (e) Rel. Score

Figure 7.6: Feature weight threshold estimation using heldout data. **Facebook dataset**, *ShCl* preprocessing pipe, *SVM* classifier, three classes, no prior feature space pruning, **FS5:** unigrams, bigrams, POS, emoticons, character n-grams

7.3.4 Summary of Results for Social Media

Given the results achieved on the Facebook dataset, the following strategies for sentiment analysis of social media in Czech can be considered. First, the preprocessing pipeline should take into account text properties typical of social media, such as proper tokenization (with respect to emoticons, URLs, etc.), stemming, and lower-casing. Additional normalization, such as phonetic transcription, can also increase performance because of the many grammatical errors present in such texts (the case of e.g. *i/y; ie/ě* in Czech). Second, the Maximum Entropy classifier yields better results than the linear kernel SVM; moreover, the training is significantly shorter. The feature set consisting of unigrams, bigrams, emoticons, and various POS features gives the best overall results. Third, filtering named entities or feature selection did not improve the overall performance for our dataset.

7.4 Conclusion

This chapter presented an in-depth research on supervised machine learning methods for sentiment analysis of Czech social media.

We created a large Facebook dataset containing 10,000 posts, accompanied by human annotation with substantial agreement (Cohen's κ 0.66) and two automatically labeled datasets - one for movie reviews and one for product reviews. All three labeled datasets are available under the Creative Commons BY-NC-SA licence[19] at `<http://liks.fav.zcu.cz/sentiment>`.

[19] `<http://creativecommons.org/licenses/by-nc-sa/3.0/>`

We thoroughly evaluated various state-of-the-art features and classifiers as well as different language-specific preprocessing techniques and feature selection algorithms. We significantly outperformed the baseline (unigram feature without preprocessing) in three-class classification and achieved an F-measure of 0.69 using a combination of features (unigrams, bigrams, POS features, emoticons, character n-grams) and preprocessing techniques (unsupervised stemming and phonetic transcription). In addition, we reported results in two other domains (movie and product reviews) with a significant improvement over the baseline.

To the best of our knowledge, our papers [Habernal et al., 2013, 2014] represent the first research that deals with sentiment analysis in Czech social media in such a thorough manner. Not only does it use a dataset that is magnitudes larger than any in the related work but also incorporates state-of-the-art features and classifiers. We believe that the outcomes of this research will not only help to set the common ground for sentiment analysis for the Czech language but also help to extend the research beyond the mainstream languages.

8 Aspect-Based Sentiment Analysis

In this chapter, we describe our approaches to the ABSA task in Czech and English.

8.1 Czech and English SemEval 2014

This section is based on the paper [Hercig et al., 2016b]. We examine the effectiveness of several unsupervised methods for latent semantics discovery as features for aspect-based sentiment analysis (ABSA). We use the shared task definition from SemEval 2014.

8.1.1 The ABSA Task

Aspect-based sentiment analysis firstly identifies the aspects of the target entity and then assigns a polarity to each aspect. There are several ways to define aspects and polarities. We use the definition based on the SemEval 2014's ABSA task, which distinguishes two types of aspect-based sentiment: aspect terms and aspect categories. The whole task is divided into four subtasks. For detailed description see Section 2.3.

8.1.2 The Data

The methods described in Section 5 require large unlabeled data in order to be trained. In this paper we used two types of corpora, labeled and unlabeled for both Czech and English. The properties of these corpora are shown in Table 8.1.

Labeled corpora for both languages are required to train the classifiers (see Section 8.1.3). For English, we use the corpora introduced in SemEval 2014 Competition Task 4 [Pontiki et al., 2014]. The main criterion in choosing the dataset was the dataset size (see Table 8.1).

Dataset	Sentences	Targets	Categories	Tokens	Words
English labeled 2016 train + test	2.7k	2.5k	3.4k	39.1k	4.4k
English labeled 2015 train + test	2k	1.9k	2.5k	29.1k	3.6k
English labeled 2014 train	3k	3.7k	3.7k	46.9k	4.9k
Czech labeled 2014 train	2.15k	3.3k	3k	34.9k	7.8k
English unlabeled	409k	–	–	27M	121k
Czech unlabeled	514k	–	–	15M	259k

Table 8.1: Properties of the SemEval ABSA tasks and corpora used in the experiments in terms of the number of *sentences*, aspect terms (*targets*), aspect categories (*categories*), *tokens* and unique *words*

For Czech, we extended the dataset from [Steinberger et al., 2014], nearly doubling its size. The annotation procedure was identical to that of the original dataset. The corpus was annotated by five native speakers. The majority voting scheme was applied to the gold label selection. Agreement between any two annotators was evaluated in the same way as we evaluate our system against the annotated data (taken as the gold standard). This means we take the output of the first annotator as the gold standard and the output of the second annotator as the output of the system. The same evaluation procedure as Pontiki et al. [2014] used, i.e. the F-measure for the aspect term and aspect category extraction, and the accuracy for the aspect term and aspect category polarity. The resulting mean values of annotator agreement for the Czech labeled corpus are 82.9% (aspect term extraction), 88.0% (aspect category extraction), 85.7% (aspect term polarity) and 88.4% (aspect category polarity). We believe this testifies to the high quality of our corpus. The corpus is available for research purposes at `<http://nlp.kiv.zcu.cz/research/sentiment>`.

The labeled corpora for both languages use the same annotation scheme and are in the same domain. This allows us to compare the effectiveness of the used features on the ABSA task for these two very different languages.

The lack of publicly available data in the restaurant domain in Czech forced us to create a cross-domain unlabeled corpus for Czech. The Czech unlabeled corpus is thus composed of three related domains: recipes (8.8M tokens, 57.1%), restaurant reviews (2M tokens, 12.8%), and hotel reviews (4.7M tokens, 30.1%). We selected these three domains because of their close relations, which should be sufficient for the purposes of the ASBA task.

The English unlabeled corpus was downloaded from the site `<http://opentable.com>`.

8.1.3 The ABSA System

We use and extend the systems created by Brychcín et al. [2014]. We implemented four separate systems – one for each subtask of ABSA. The required machine learning algorithms are implemented in the Brainy machine learning library [Konkol, 2014]. We further extended this system and competed in the SemEval 2016 ABSA task and we were ranked as one of the top performing systems [Hercig et al., 2016a].

The systems share a simple preprocessing phase, in which we use a tokenizer based on regular expressions. The tokens are transformed to lower case. Punctuation marks and stop words are ignored for the polarity task. In the case of Czech, we also remove diacritics from all the words, because of their inconsistent use.

The feature sets created for individual tasks are based on features commonly used in similar natural language processing tasks, e.g. named entity recognition [Konkol and Konopík, 2013], document-level sentiment analysis [Habernal et al., 2014], and document classification [Brychcín and Král, 2014]. The following baseline features were used:

Affixes (A) – Affix (length 2-4 characters) of a word at a given position.

TF-IDF (T) – Term frequency - inverse document frequency of a word.

Learned dictionary (LD) – Dictionary of aspect terms from training data.

Words (W) – The occurrence of word at a given position (e.g. previous word).

Bag of words (BoW) – The occurrence of a word in the context window.

Bigrams (B) – The occurrence of bigram at a given position.

Bag of bigrams (BoB) – The occurrence of a bigram in the context window.

The baseline feature set is then extended with semantic features. The features are based on the word clusters created using the semantic models described in Section 5. The following semantic features were used:

Clusters (C) – The occurrence of a cluster at a given position.

Bag of clusters (BoC) – The occurrence of a cluster in the context window.

Cluster bigrams (CB) – The occurrence of cluster bigram at a given position.

Bag of cluster bigrams (BoCB) – The occurrence of cluster bigram in the context window.

Each C (alternatively, CB, BoC, or BoCB) feature can be based on any of the models from Section 5. In the description of the systems for individual tasks, we use simply C to denote that we work with this type of feature. When we later describe the experiments, we use explicitly the name of the model (e.g. HAL).

Subtask 1: Aspect Term Extraction

The aspect term extraction is based on experiences in NER task [Konkol et al., 2015, Konkol and Konopík, 2013]. The NER task tries to find special expressions in a text and classify them into groups. The aspect term extraction task is very similar, because it also tries to identify special expressions. In contrast with NER, these expressions are not classified, and have different properties, e.g. they are not so often proper names.

We have decided to use CRF, because they are regarded as the state-of-the-art method for NER. The baseline feature set consists of W, BoW, B, LD, and A. In our experiments, we extend this with the semantic features C and CB. The context for this task is defined as a five word window centred at the currently processed word.

Subtask 2: Aspect Term Polarity

Our aspect term polarity detection is based on the Maximum Entropy classifier, which works very well in many NLP tasks, including document-level sentiment analysis [Habernal et al., 2014].

For each aspect term, we create a context window ten words to the left and right of the aspect term. The features for each word and bigram in this window are weighted based on their distance from the aspect term given by weighing function. This follows the general belief that close words are more

important than distant words, which is used in several methods [Lund and Burgess, 1996].

We have tested several weighing functions and selected the Gaussian function based on the results. The expected value μ and the variance σ^2 of the Gaussian function were found experimentally on the training data.

The feature set for our baseline system consists of *BoW* and *BoB*, and we further experiment with *BoC* and *BoCB*.

Subtask 3: Aspect Category Extraction

The aspect category extraction is based on research in multi-label document classification [Brychcín and Král, 2014]. The multi-label document classification system tries to assign several labels to a document. We do exactly the same, although our documents are only single sentences and the labels are aspect term categories.

We use one binary Maximum Entropy classifier for each category. It decides whether the sentence belongs to the given category. The whole sentence is used as the context.

The baseline uses the features *BoW*, *BoB*, and *T*. We try to improve it with *BoC* and *BoCB*.

Subtask 4: Aspect Category Polarity

The aspect category task is similar to document-level sentiment analysis [Habernal et al., 2014] when the document is of similar length. We create one Maximum Entropy classifier for each category. For a given category, the classifier uses the same principle as in global sentiment analysis. Of course, the training data are different for each category. The context in this task is the whole sentence.

We use the following features as a baseline: *BoW*, *BoB*, and *T*. In our experiments, we extend this with *BoC* and *BoCB*.

8.1.4 Experiments

In the following presentation of the results of the experiments, we use the notation *BL* for a system with the baseline feature set (i.e. without cluster

features). Cluster features based on HAL are denoted by *HAL*. For other semantic spaces, the notation is analogous.

Because Czech has rich morphology we use stemming to deal with this problem (stemming is denoted as S). Also we use the stemmed versions of semantic spaces (the corpora used for training semantics spaces are simply preprocessed by stemming). The system that uses this kind of cluster features is denoted by *S-HAL* for the HAL model, and analogously for the other models.

The union of feature sets is denoted by the operator *+*. For example *BL+S-BL+S-GloVe* denotes the baseline feature set extended by stemmed baseline features and by a stemmed version of GloVe clusters.

The number of clusters for a particular semantic space is always explicitly mentioned in the following tables.

Unsupervised Model Settings

All unsupervised models were trained on the unlabeled corpora described in Section 8.1.2.

The implementations of the HAL and COALS algorithms are available in an open source package S-Space [Jurgens and Stevens, 2010][1]. The settings of the GloVe, CBOW, and Skip-gram models reflect the results of these methods in their original publications [Pennington et al., 2014, Mikolov et al., 2013a] and were set according to a reasonable proportion of the complexity and the quality of the resulting word vector outputs. We used the GloVe implementation provided on the official website[2], CBOW and Skip-gram models use the Word2Vec[3] implementation and the LDA implementation comes from the MALLET [McCallum, 2002] software package.

The detailed settings of all these methods are shown in Table 8.2.

CLUTO software package [Karypis, 2003] is used for words clustering with the k-means algorithm and cosine similarity metric. All vector space models in this paper cluster the word vectors into four different numbers of clusters: 100, 500, 1000, and 5000. For stemming, we use the implementation of

[1] Available at <https://code.google.com/p/airhead-research/>.
[2] Available at <http://www-nlp.stanford.edu/projects/glove/>.
[3] Available at <https://code.google.com/p/word2vec/>.

	dimension	window	special settings
HAL	50,000	4	
COALS	14,000	4	without SVD
GloVe	300	10	100 iterations
CBOW	300	10	100 iterations
SKIP	300	10	100 iterations
LDA	100	sentence	1000 iterations

Table 8.2: Model settings

HPS [Brychcín and Konopík, 2015][4] that is the state-of-the-art unsupervised stemmer.

8.1.5 Results

We experimented with two morphologically very different languages, English, and Czech. English, as a representative of the Germanic languages, is characterized by almost no inflection. Czech is a representative of the Slavic languages, and has a high level of inflection and relatively free word order.

We provide the same evaluation as in the SemEval 2014 [Pontiki et al., 2014]. For the aspect term extraction (TE) and the aspect category extraction (CE) we use F-measure as an evaluation metric. For the sentiment polarity detection of aspect terms (TP) and aspect categories (CP), we use accuracy.

We use 10-fold cross-validation in all our experiments. In all the tables in this section, the results are expressed in percentages, and the numbers in brackets represents the absolute improvements against the baseline.

We started our experiments by testing all the unsupervised models separately. In the case of Czech, we also tested stemmed versions of all the models. For English, we did not use stemming, because it does not play a key role [Habernal et al., 2014]. The results are shown in Tables 8.3 and 8.4.

Each model brings some improvement in all the cases. Also, the stemmed versions of the models are almost always better than the unstemmed models. Thus, we continued the experiments only with the stemmed models for Czech. The stems are used as a separate features and are seen to be very useful for

[4]Available at <http://liks.fav.zcu.cz/HPS>.

Clusters	100		500		1000		5000	
Task	TE	TP	TE	TP	TE	TP	TE	TP
BL+HAL	**79.3 (+3.7)**	69.0 (+1.6)	**78.7 (+3.1)**	68.8 (+1.4)	78.3 (+2.7)	69.2 (+1.8)	**78.6 (+3.0)**	69.6 (+2.3)
BL+COALS	77.6 (+1.9)	67.5 (+0.1)	77.3 (+1.7)	67.7 (+0.3)	77.3 (+1.7)	67.5 (+0.1)	76.6 (+0.9)	68.8 (+1.5)
BL+CBOW	78.4 (+2.8)	69.4 (+2.0)	78.6 (+3.0)	69.4 (+2.0)	**79.1 (+3.5)**	**69.3 (+2.0)**	77.8 (+2.2)	**69.8 (+2.4)**
BL+SKIP	77.8 (+2.2)	**69.9 (+2.5)**	77.6 (+2.0)	68.2 (+0.9)	77.9 (+2.3)	68.7 (+1.4)	77.6 (+2.0)	68.3 (+0.9)
BL+GLOVE	77.5 (+1.9)	69.2 (+1.8)	77.6 (+2.0)	**69.8 (+2.4)**	77.7 (+2.0)	**69.3 (+2.0)**	77.1 (+1.5)	68.8 (+1.4)
BL+LDA	77.4 (+1.8)	68.5 (+1.1)	77.2 (+1.5)	68.6 (+1.2)	77.8 (+2.2)	68.7 (+1.3)	77.1 (+1.4)	68.9 (+1.5)
Clusters	100		500		1000		5000	
Task	CE	CP	CE	CP	CE	CP	CE	CP
BL+HAL	78.6 (+1.2)	68.4 (+0.2)	**79.3 (+1.9)**	69.0 (+0.7)	78.5 (+1.0)	69.7 (+1.5)	78.8 (+1.3)	69.5 (+1.2)
BL+COALS	78.2 (+0.7)	68.1 (–0.2)	78.8 (+1.3)	68.7 (+0.4)	78.1 (+0.6)	68.9 (+0.6)	77.9 (+0.4)	69.5 (+1.2)
BL+CBOW	78.8 (+1.3)	**70.5 (+2.2)**	79.3 (+1.8)	**70.7 (+2.4)**	79.0 (+1.6)	69.9 (+1.6)	78.7 (+1.3)	**70.9 (+2.6)**
BL+SKIP	78.4 (+0.9)	69.4 (+1.1)	78.1 (+0.7)	70.0 (+1.7)	78.7 (+1.3)	**70.6 (+2.3)**	**79.3 (+1.8)**	70.4 (+2.1)
BL+GLOVE	**79.3 (+1.8)**	69.8 (+1.5)	79.1 (+1.6)	70.1 (+1.8)	**79.4 (+1.9)**	70.4 (+2.1)	78.8 (+1.3)	69.9 (+1.6)
BL+LDA	78.4 (+0.9)	69.6 (+1.3)	78.5 (+1.0)	69.5 (+1.2)	78.6 (+1.1)	68.8 (+0.5)	77.7 (+0.3)	69.1 (+0.8)

Table 8.3: Aspect term and category extraction (TE, CE) and polarity (TP, CP) results on English dataset

Clusters	100		500		1000		5000	
Task	TE	TP	TE	TP	TE	TP	TE	TP
BL+HAL	75.6 (+4.2)	67.4 (+0.0)	75.4 (+4.0)	68.3 (+0.9)	75.5 (+4.0)	68.5 (+1.1)	75.0 (+3.5)	69.9 (+2.6)
BL+S-HAL	74.0 (+2.5)	66.8 (–0.5)	75.4 (+4.0)	68.5 (+1.1)	75.8 (+4.3)	69.4 (+2.0)	**76.9 (+5.5)**	70.3 (+2.9)
BL+COALS	75.1 (+3.6)	67.0 (–0.4)	74.5 (+3.0)	67.9 (+0.5)	74.6 (+3.2)	68.0 (+0.6)	74.3 (+2.8)	68.7 (+1.3)
BL+S-COALS	75.2 (+3.7)	69.5 (+2.1)	75.4 (+4.0)	69.2 (+1.9)	75.4 (+4.0)	68.8 (+1.4)	75.5 (+4.1)	69.5 (+2.1)
BL+CBOW	75.4 (+3.9)	68.2 (+0.9)	75.4 (+3.9)	69.7 (+2.3)	75.7 (+4.3)	70.5 (+3.1)	75.3 (+3.9)	70.1 (+2.7)
BL+S-CBOW	**75.8 (+4.3)**	69.6 (+2.2)	74.9 (+3.5)	**70.4 (+3.0)**	75.6 (+4.2)	70.1 (+2.7)	73.2 (+1.8)	**71.1 (+3.7)**
BL+SKIP	74.9 (+3.4)	69.4 (+2.0)	74.8 (+3.3)	70.2 (+2.8)	75.9 (+4.5)	**70.8 (+3.4)**	74.8 (+3.3)	69.4 (+2.0)
BL+S-SKIP	75.4 (+4.0)	**69.6 (+2.2)**	75.3 (+3.8)	70.6 (+3.2)	75.9 (+4.5)	69.7 (+2.3)	75.5 (+4.0)	69.9 (+2.5)
BL+GLOVE	74.3 (+2.8)	68.9 (+1.5)	75.4 (+4.0)	69.1 (+1.7)	75.6 (+4.2)	68.7 (+1.3)	74.4 (+3.0)	69.1 (+1.7)
BL+S-GLOVE	75.1 (+3.7)	69.0 (+1.6)	**76.3 (+4.9)**	70.0 (+2.6)	**76.0 (+4.6)**	69.5 (+2.1)	75.5 (+4.0)	69.6 (+2.2)
BL+LDA	74.4 (+2.9)	68.9 (+1.5)	74.5 (+3.0)	69.6 (+2.3)	73.7 (+2.3)	69.2 (+1.8)	73.1 (+1.7)	69.2 (+1.9)
BL+S-LDA	74.7 (+3.3)	69.2 (+1.8)	74.8 (+3.4)	70.2 (+2.8)	74.6 (+3.1)	69.5 (+2.1)	74.7 (+3.2)	69.8 (+2.4)
Clusters	100		500		1000		5000	
Task	CE	CP	CE	CP	CE	CP	CE	CP
BL+HAL	76.1 (+4.3)	68.9 (–0.8)	74.7 (+3.0)	71.1 (+1.4)	74.6 (+2.9)	70.8 (+1.1)	74.1 (+2.4)	71.7 (+2.0)
BL+S-HAL	75.4 (+3.7)	70.1 (+0.4)	75.6 (+3.8)	71.1 (+1.4)	75.2 (+3.5)	70.7 (+1.0)	**76.6 (+4.9)**	73.2 (+3.5)
BL+COALS	75.1 (+3.4)	69.9 (+0.2)	74.5 (+2.8)	72.5 (+2.8)	74.1 (+2.3)	70.9 (+1.2)	73.5 (+1.7)	70.9 (+1.2)
BL+S-COALS	75.0 (+3.2)	71.8 (+2.1)	75.7 (+3.9)	72.8 (+3.1)	74.8 (+3.1)	71.4 (+1.7)	74.8 (+3.1)	72.6 (+2.9)
BL+CBOW	74.8 (+3.0)	71.3 (+1.6)	74.9 (+3.2)	72.3 (+2.6)	74.4 (+2.7)	72.5 (+2.8)	74.6 (+2.8)	73.0 (+3.3)
BL+S-CBOW	76.2 (+4.5)	72.1 (+2.4)	74.0 (+2.3)	73.1 (+3.4)	75.3 (+3.6)	73.8 (+4.1)	75.3 (+3.5)	**73.9 (+4.2)**
BL+SKIP	75.6 (+3.9)	**73.6 (+3.9)**	74.7 (+3.0)	73.2 (+3.5)	75.9 (+4.1)	**74.1 (+4.4)**	74.1 (+2.3)	72.3 (+2.6)
BL+S-SKIP	75.9 (+4.1)	73.3 (+3.6)	74.5 (+2.8)	**74.1 (+4.4)**	76.1 (+4.4)	73.0 (+3.3)	75.6 (+3.8)	73.3 (+3.6)
BL+GLOVE	76.5 (+4.7)	70.9 (+1.2)	75.5 (+3.7)	71.7 (+2.0)	75.8 (+4.1)	72.1 (+2.4)	73.3 (+1.6)	71.9 (+2.2)
BL+S-GLOVE	**77.2 (+5.5)**	70.9 (+1.2)	**77.1 (+5.4)**	73.5 (+3.8)	**77.4 (+5.7)**	73.4 (+3.7)	75.9 (+4.2)	73.2 (+3.5)
BL+LDA	73.3 (+1.5)	72.3 (+2.6)	73.9 (+2.2)	72.9 (+3.2)	73.9 (+2.1)	72.8 (+3.1)	73.1 (+1.3)	72.0 (+2.3)
BL+S-LDA	73.4 (+1.7)	72.0 (+2.3)	74.4 (+2.7)	73.4 (+3.7)	74.1 (+2.3)	73.0 (+3.3)	74.2 (+2.4)	72.6 (+2.9)

Table 8.4: Aspect term and category extraction (TE, CE) and polarity (TP, CP) results on Czech dataset

Task	TE	TP	CE	CP
BL	75.6	67.4	77.5	68.3
BL+HAL	80.3 (+4.6)	70.6 (+3.2)	79.5 (+2.0)	69.5 (+1.3)
BL+COALS	78.7 (+3.0)	69.0 (+1.6)	78.6 (+1.1)	69.2 (+0.9)
BL+CBOW	**80.6 (+5.0)**	**71.1 (+3.7)**	79.3 (+1.8)	**71.4 (+3.2)**
BL+SKIP	78.9 (+3.2)	69.9 (+2.5)	**79.6 (+2.1)**	70.8 (+2.6)
BL+GLOVE	78.7 (+3.0)	70.2 (+2.8)	**79.5 (+2.1)**	70.8 (+2.5)
BL+LDA	78.5 (+2.9)	69.8 (+2.4)	78.4 (+0.9)	70.0 (+1.8)
BL+CBOW+GLOVE	80.4 (+4.8)	70.9 (+3.5)	**80.6 (+3.1)**	**72.1 (+3.8)**

Table 8.5: Models combinations on English dataset

Task	TE	TP	CE	CP
BL	71.4	67.4	71.7	69.7
BL+S-BL	74.9 (+3.4)	69.0 (+1.6)	73.6 (+1.9)	71.3 (+1.6)
BL+S-BL+S-HAL	78.5 (+7.0)	70.5 (+3.1)	78.5 (+6.8)	72.3 (+2.6)
BL+S-BL+S-COALS	77.8 (+6.3)	70.9 (+3.6)	77.5 (+5.7)	73.1 (+3.4)
BL+S-BL+S-CBOW	77.9 (+6.4)	**72.1 (+4.7)**	78.1 (+6.4)	73.6 (+3.9)
BL+S-BL+S-SKIP	77.8 (+6.3)	71.6 (+4.3)	78.0 (+6.3)	**75.2 (+5.5)**
BL+S-BL+S-GLOVE	**78.5 (+7.1)**	71.3 (+3.9)	**79.5 (+7.8)**	74.1 (+4.4)
BL+S-BL+S-LDA	77.4 (+6.0)	70.2 (+2.9)	75.6 (+3.8)	73.4 (+3.7)
BL+S-BL+S-CBOW+S-GLOVE	**78.7 (+7.3)**	**72.5 (+5.1)**	**80.0 (+8.3)**	74.0 (+4.3)

Table 8.6: Model combinations on Czech dataset

Czech (see Table 8.6).

In the subsequent experiments, we tried to combine all the clusters from one model. We assumed that different clustering depths could bring useful information into the classifier. These combinations are shown in Table 8.5 for English and Table 8.6 for Czech. We can see that the performance was considerably improved. Taking these results into account, the best models for ABSA seem to be GloVe and CBOW.

To prevent overfitting, we cannot combine all the models and all the clustering depths together. Thus, we only combined the two best models (GloVe, CBOW). The results are shown again in Tables 8.5 and 8.6 in the last row. In all the subtasks, the performance stagnates or slightly improves.

Our English baseline extracts aspect terms with 75.6% F-measure and aspect categories with 77.6% F-measure. The Czech baseline is considerably worse, and achieves the results 71.4% and 71.7% F-measures in the same subtasks. The behaviour of our baselines for sentiment polarity tasks is different. The baselines for aspect term polarity and aspect category polarity in both languages perform almost the same: the accuracy ranges between

67.4% and 69.7% for both languages.

In our experiments, the word clusters from semantic spaces (especially CBOW and GloVe models) and stemming by HPS proved to be very useful. Large improvements were achieved for all four subtasks and both languages. The aspect term extraction and aspect category extraction F-measures of our systems improved to approximately 80% for both languages. Similarly, the polarity detection subtasks surpassed 70% accuracy, again for both languages.

8.1.6 Conclusion

In our experiments we used labeled and unlabeled corpora within the restaurants domain for two languages: Czech and English.

We explored several unsupervised methods for word meaning representation. We created word clusters and used them as features for the ABSA task. We achieved considerable improvements for both the English and Czech languages. We also used the unsupervised stemming algorithm called HPS, which helped us to deal with the rich morphology of Czech.

Out of all the tested models, GloVe and CBOW seem to perform the best, and their combination together with stemming for Czech was able to improve all four ABSA subtasks. Moreover, we achieve new state-of-the-art results for Czech.

We created two new Czech corpora within the restaurant domain for the ABSA task: one labeled for supervised training, and the other (considerably larger) unlabeled for unsupervised training. The corpora are available to the research community.

Since none of the methods used to improve ABSA in this paper require any external information about the language, we assume that similar improvements can be achieved for other languages. Thus, the main direction for future research is to experiment with more languages from different language families.

8.2 English SemEval 2016

This section describes our system used in the ABSA task of SemEval 2016 [Hercig et al., 2016a]. Our system is build upon the previous one in Section 8.1. We use Maximum Entropy classifier for the aspect category detection

and for the sentiment polarity task. Conditional Random Fields (CRF) are used for opinion target extraction.

8.2.1 Introduction

In the current ABSA task - SemEval 2016 task 5 [Pontiki et al., 2016] has attracted 29 participating teams competing in 40 different experiments among 8 languages. The task has three subtasks: Sentence-level (SB1), Text-level (SB2), and Out-of-domain ABSA (SB3). The subtasks are further divided into three slots:

- 1) Aspect Category Detection – the category consists of an entity and attribute (E#A) pair.
- 2) Opinion Target Expression (OTE)
- 3) Sentiment Polarity (positive, negative, neutral, and for SB2 conflict)

In phase A we solved slots 1 and 2. In phase B we were given the results for slots 1 and 2 and solved slot 3. We participate in 19 experiments including Chinese, English, French, and Spanish.

8.2.2 System Description

Our approach to the ABSA task is based on supervised Machine Learning. Detailed description for each experiment can be found in Section 8.2.6 and Section 8.2.7.

For all experiments we use Brainy [Konkol, 2014] machine learning library. Data preprocessing includes lower-casing and in some cases lemmatization. We utilize parse trees, lemmatization and POS tags from the Stanford CoreNLP [Manning et al., 2014] v3.6 framework. We chose it because it has support for Chinese, English, French, and Spanish.

Our system combines a large number of features to achieve competitive results. In the following sections we will describe the features in detail.

8.2.3 Semantics Features

We use semantics models to derive word clusters from unlabeled datasets. Similarly to [Toh and Su, 2015] we use the Amazon product reviews from

	Dimension	**Window**	**Iterations**
GloVe	300	10	100
CBOW	300	10	100
LDA	–	sentence	1000

Table 8.7: Model settings.

[Blitzer et al., 2007], the user reviews from the Yelp Phoenix Academic Dataset[5], and a review Opentable dataset[6] to create semantic word clusters. We consider GloVe, CBOW, and LDA semantics models.

The settings of the GloVe and CBOW models reflect the results of these methods in their original publications [Pennington et al., 2014, Mikolov et al., 2013a]. For LDA we experiment with 50, 100, 200, 300, 400, and 500 topics. The detailed settings of all these methods are shown in Table 8.7.

We used the GloVe implementation provided on the official website[7], CBOW model uses the Word2Vec[8] implementation and the LDA implementation comes from the MALLET [McCallum, 2002] software package.

CLUTO software package [Karypis, 2003] is used for words clustering with the k-means algorithm and cosine similarity metric. All vector space models in this paper cluster the word vectors into four different numbers of clusters: 100, 500, 1000, and 5000.

The following features are based on the word clusters created using the semantic models.

Clusters (C) – The occurrence of a cluster at a given position.

Bag of Clusters (BoC) – The occurrence of a cluster in the context window.

Cluster Bigrams (CB) – The occurrence of cluster bigram at a given position.

Bag of Cluster Bigrams (BoCB) – The occurrence of cluster bigram in the context window.

[5] <https://www.yelp.com/dataset_challenge>
[6] downloaded from <http://opentable.com>
[7] <http://nlp.stanford.edu/projects/glove>
[8] <https://code.google.com/p/word2vec>

8.2.4 Constrained Features

Affixes (A) – Affix (length 2-4 characters) of a word at a given position with a frequency > 5.

Aspect Category (AC) – extracted aspect category. We use separately the entity, attribute, and the E#A pair.

Aspect Target (AT) – listed aspect target.

Bag of Words (BoW) – The occurrence of a word in the context window.

Bag of Words filtered by POS (BoW-POS) – The occurrence of a word in the context window filtered by POS tags.

Bag of Bigrams (BoB) – The occurrence of a bigram in the context window.

Bag of Words around Verb (5V) – Bag of 5 words before verb and a bag of 5 words after verb.

Bag of 5 Words at the Beginning of Sentence (5sS) – Bag of 5 words at the beginning of a sentence.

Bag of 5 Words at the End of Sentence (5eS) – Bag of 5 words at the end of a sentence.

Bag of Head Words (BoHW) – bag of extracted head words from the sentence parse tree.

Emoticons (E) We used a list of positive and negative emoticons [Montejo-Ráez et al., 2012]. The feature captures the presence of an emoticon within the text.

Head Word (HW) – extracted head word from the sentence parse tree.

Character N-gram (ChN) – The occurrence of character n-gram at a given position.

Learned Target Dictionary (LTD) – presence of a word from learned[9] dictionary of aspect terms.

Learned Target Dictionary by Category (LTD-C) – presence of a word from the learned dictionary[9] of aspect terms grouped by category.

[9]from training data

N-gram (N) – The occurrence of n-gram in the context window.

N-gram Shape (NSh) – The occurrence of word shape n-gram in the context window. We consider unigrams with frequency >5 and bigrams, trigrams with frequency > 20.

Paragraph Vectors (P2Vec) is an unsupervised method of learning text representation [Le and Mikolov, 2014]. Resulting feature vector has a fixed dimension while the input text can be of any length. The model is trained on the *One billion word benchmark* presented in [Chelba et al., 2013], resulting vectors[10] are used as features for a sentence. We use the implementation by Řehůřek and Sojka [2010].

POS N-gram (POS-N) – The occurrence of POS n-gram in the context window.

Punctuation (P) – The occurrence of a question mark, an exclamation mark or at least two dots in the context window.

Skip-bigram (SkB) – Instead of using sequences of adjacent words (n-grams) we used skip-grams [Guthrie et al., 2006, Reyes et al., 2013], which skip over arbitrary gaps. We consider skip-bigrams with 2 to 5 word skips and remove skip-grams with a frequency ≤ 5.

Target Bag of Words (T-BoW) – BoW containing parent, siblings, and children of the target from the sentence parse tree.

TF-IDF (TF-IDF) – Term frequency - inverse document frequency of a word computed from the training data.

Verb Bag of Tags (V-BoT) – Bag of syntactic dependency tags of parent, siblings, and children of the verb from the sentence parse tree.

Verb Bag of Words (V-BoW) – Bag of words for parent, siblings, and children of the verb from the sentence parse tree.

Word Shape (WSh) – we assign words into one of 24 classes[11] similar to the function specified in [Bikel et al., 1997].

Words (W) – The occurrence of word at a given position (e.g. previous word).

[10] Vector dimension has been set to 300.

[11] We use edu.stanford.nlp.process.WordShapeClassifier with the WORDSHAPE-CHRIS1 setting.

8.2.5 Unconstrained Features

Dictionary (DL) – presence of a word from dictionary extracted from the Annotation Guidelines for Laptops.

Dictionary (DR) – presence of a word from dictionary extracted from the Annotation Guidelines for Restaurants.

Enchanted Dictionary (ED) – presence of a word from a dictionary extracted from website[12].

Group of Words from ED (EDG) – presence of any word from a group from the ED dictionary.

Dictionary of Negative Words (ND) – presence of any negative word from the negative words list[13].

Sentiment (S) – this is a union of features dealing with sentiment. It consists of *BoG* features where the groups correspond to various sentiment lexicons. We used the following lexicon resources: Affinity lexicon [Nielsen, 2011], Senticon [Cruz et al., 2014], dictionaries from [Steinberger et al., 2012], MICRO-WNOP [Cerini et al., 2007], and the list of positive or negative opinion words from [Liu et al., 2005]. Additional feature includes the output of Stanford CoreNLP [Manning et al., 2014] v3.6 sentiment analysis package by Socher et al. [2013].

8.2.6 Phase A

Sentence-Level Category (SB1, slot 1) We use Maximum Entropy classifier for all classes. Then a threshold t is used to decide which categories will be assigned by the classifier.

Chinese We used identical features for both domains (*BoB, BoHW, BoW, ChN, N*), where *ChN* ranges from unigram to 4-gram and *ChN* with frequency < 20 are removed and N ranges from unigram to trigram and *ChN* with frequency < 10 are removed. The threshold was set to $t = 0.1$.

[12] <http://www.enchantedlearning.com/wordlist/>

[13] <http://dreference.blogspot.cz/2010/05/negative-ve-words-adjectives-list-for.html>

Spanish For Spanish we used the following features: *5V, 5eS, BoB, BoHW, BoW, BoW-POS, ChN, V-BoT*, where *5V* considers only adjective, adverb, and noun, *5eS* considers adjectives and adverbs with frequency > 5, *ChN* ranges from unigram to 4-gram and *ChN* with frequency < 20 are removed, *BoW-POS* is used separately for adverbs, nouns, verbs, and a union of adjectives, adverbs, nouns, and verbs, *V-BoT* is used separately for adverbs, nouns, and a union of adjectives and adverbs while reducing feature space by 50 occurrences. The threshold was set to $t = 0.2$.

English English features employ lemmatization. The threshold was set to $t = 0.14$. Common features for all experiments in this task are *5V, 5eS, BoB, BoHW, BoW, BoW-POS, P, TF-IDF, V-BoT*, where *5V* considers only adjective, adverb, and noun, *5eS* filters only adjective and adverb, *BoW-POS* contains adjectives, adverbs, nouns, and verbs, *V-BoT* filters adjectives and adverbs with frequency > 20.

The unconstrained model for the Laptops domain additionally uses *BoC, BoCB, DL, ED, P2Vec BoC* and *BoCB* include the GloVe and CBOW models computed on the Amazon dataset.

The constrained model for the restaurant domain additionally uses *5sS, ChN, LTD, LTD-C, P2Vec 5sS* filters only adjective and adverb, *ChN* in this case means character unigrams with frequency > 5. This model also considers separate *BoW-POS* features for groups for adverbs, nouns and verbs.

The unconstrained model for the restaurant domain uses *BoC, BoCB, DR, LDA, ND, NSh* on top of the previously listed features for the constrained model.

BoC and *BoCB* include the GloVe, CBOW, and LDA models computed on the Yelp dataset and CBOW model computed on the Opentable dataset.

Sentence-Level Target (SB1, slot 2) Similarly to [Brychcín et al., 2014], we have decided to use CRF to solve this subtask. The context for this task is defined as a five word window centred at the currently processed word. English features for this subtask employ lemmatization.

The baseline feature set consists of *A, BoB, BoW-POS, HW, LTD, LTD-C, N, POS-N, V-BoT, W, WSh*. *BoW-POS* contains adjectives, adverbs, nouns, verbs, and a union of adverbs and nouns. We consider *POS-N* with frequency > 10. *V-BoT* includes adverbs, nouns, and a union of adjectives,

adverbs, nouns, and verbs.

In the unconstrained model, we extend this with the semantic features *C, CB* (created using the CBOW model computed on the Opentable dataset) and with lexicons *DR, EDG*.

Sentence-Level Category & Target (SB1, slot 1&2) We assign targets as described above and then combine them with the best five candidates for aspect category[14]. We also add aspect categories without target. This produces too many combinations, thus we need to filter the unlikely opinions. We remove the opinions without target in a sentence where the aspect category is already present with a target. When there is only one target and one aspect category in a sentence we combine them into a single opinion.

Text-Level Category (SB2, slot 1) We used the baseline algorithm: the predicted sentence-level tuples (SB1, slot 1) are copied to text-level and duplicates are removed.

8.2.7 Phase B

Sentence-Level Sentiment Polarity (SB1, slot 3) Our sentiment polarity detection is based on the Maximum Entropy classifier, which works very well in many NLP tasks, including document-level sentiment analysis [Habernal et al., 2014].

Chinese We used identical features for both domains (*5V, 5eS, 5sS, AC, BoB, BoHW, BoW, BoW-POS, ChN, N, NSh, P, SkB, V-BoT, V-BoW*), where *5V* considers adjectives and adverbs with frequency > 5, *5eS* and *5sS* contain adjectives, adverbs, nouns, and verbs, *BoW-POS* is used separately for adjectives and adverbs, *ChN* ranges from unigram to 5-gram and *ChN* with frequency < 5 are removed, *N* ranges from unigram to 5-gram and *ChN* with frequency < 2 are removed, *V-BoT* is used separately for verbs, and a union of adjectives and adverbs, *V-BoW* is used separately for adjectives, adverbs, verbs, a union of adjectives and adverbs and a union of adjectives, adverbs, nouns, and verbs, while reducing feature space by 2 occurrences.

French We employ lemmatization for French. The first constrained model includes the following features: *AC, BoB, BoHW, BoW, BoW-POS, ChN,*

[14]We use the same settings and approach as in the sentence-level category detection (SB1 slot 1).

LTD, LTD-C, N, NSh, P, SkB, V-BoT, where *BoW-POS* is used separately for adjectives, adverbs and a union of adjectives, adverbs, nouns, and verbs, *ChN* ranges from unigram to 5-gram and *ChN* with frequency ≤ 5 are removed, *N* ranges from unigram to 5-gram and *N* with frequency < 2 are removed, *V-BoT* is used separately for verbs, and a union of adjectives and adverbs.

The second constrained model additionally uses *5V, 5eS, 5sS, AT, T-Bow, V-BoW*, where *5V* considers only adjective and adverb, *5eS, 5sS* considers adjective, adverb, noun, and verb, *T-BoW* is used for adjectives, adverbs, nouns, and verbs, *V-BoW* is used separately for adjectives, adverbs, verbs, a union of adjectives and adverbs and a union of adjectives, adverbs, nouns, and verbs, while reducing feature space by 2 occurrences.

Spanish We employ lemmatization for Spanish. We used the following features: *5V, 5eS, AC, BoB, BoHW, BoW, BoW-POS, E, ChN, LTD, LTD-C, N, NSh, P, SkB, T-Bow, V-BoT, V-BoW*, where *5V* considers only adjective and adverb, *5eS* considers adjective, adverb, noun, and verb, *BoW-POS* is used separately for adjectives, adverbs, *ChN* ranges from unigram to 5-gram and *ChN* with frequency ≤ 5 are removed, *N* ranges from unigram to 5-gram and *N* with frequency < 2 are removed, *T-BoW* is used for adjectives, adverbs, nouns, and verbs, *V-BoT* is used separately for verbs, and a union of adjectives and adverbs, *V-BoW* is used separately for adjectives, adverbs, verbs, a union of adjectives and adverbs and a union of adjectives, adverbs, nouns, and verbs, while reducing feature space by 2 occurrences.

English We use lemmatization in this subtask. Common features for all experiments in this task are *5V, 5eS, 5sS, AC, AT, BoB, BoHW, BoW, BoW-POS, E, ChN, LTD, LTD-C, N, NSh, P, SkB, V-BoT, V-BoW*, where *5V* considers adjectives and adverbs, *5eS, 5sS* consists of adjectives, adverbs, nouns, and verbs, , *BoW-POS* contains adjectives and adverbs, *N* ranges from unigram to 5-gram and *N* with frequency < 2 are removed, *V-BoT* is used separately for verbs, and a union of adjectives and adverbs, *V-BoW* is used separately for adjectives, adverbs, verbs, a union of adjectives and adverbs and a union of adjectives, adverbs, nouns, and verbs, while reducing feature space by 2 occurrences.

The unconstrained model for the Laptops domain additionally uses *BoC, BoCB, ED, S BoC* and *BoCB* include the GloVe and CBOW models computed on the Amazon dataset.

The constrained model for the restaurant domain additionally uses *T-Bow, TF-IDF*, where *T-BoW* is used for adjectives, adverbs, nouns, and verbs.

The unconstrained model for the restaurant domain uses *BoC, BoCB, ED, ND, S* on top of the previously listed features for the constrained model.

BoC and *BoCB* include the GloVe and CBOW models computed on the Yelp dataset and CBOW model computed on the Opentable dataset.

Text-Level Sentiment Polarity (SB2, slot 3) The baseline algorithm traverses the predicted sentence-level tuples of the same category and counts the respective polarity labels (positive, negative or neutral). Finally, the polarity label with the highest frequency is assigned to the text-level category. If there are not any sentence-level tuples of the same category the polarity label is determined based on all tuples regardless of the category.

Our improved algorithm contains an additional step, that assigns polarity for cases (categories) with more than one sentence-level polarity labels. The resulting polarity is determined by the following algorithm:

```
if(catPolarity == lastPolarity){
   assign lastPolarity;
}else if(catPolarity == entPolarity){
   assign entPolarity;
}else{
   assign CONFLICT;
}
```

where `catPolarity` is the polarity label with the highest frequency for the given category (E#A touple), `entPolarity` is the polarity label with the highest frequency for the entity E and `lastPolarity` is the last seen polarity label for the given category. This follows our believe that the last polarity tends to reflect the final sentiment (opinion) toward the aspect category.

8.2.8 Results and Discussion

As shown in the Table 8.8 we achieved very satisfactory results especially for the constrained experiments.

In the English sentence-level Laptops domain our constrained method was slightly better than the unconstrained one (by 0.6%).

			Constrained				Unconstrained			
Domain	**Lang**	**Subtask**	**Category**		**Sentiment**		**Category**		**Sentiment**	
			Rank	F_1[%]	Rank	ACC[%]	Rank	F_1[%]	Rank	ACC[%]
Restaurants	EN	SB1	3.	67.8	2.	81.8	8.	68.2	9.	81.7
Laptops	EN	SB1	1.	47.9	3.	73.8	7.	47.3	10.	73.8
Restaurants	EN	SB2	1.	81.0	1.	80.9	3.	80.2	1.	81.9
Laptops	EN	SB2	1.	60.5	1.	74.5	2.	59.7	1. - 2.	75.0
Restaurants	FR	SB1	–	–	2.	75.3	–	–	–	–
Cameras	CH	SB1	1.	36.3	3.	77.8	–	–	–	–
Phones	CH	SB1	1.	22.5	3.	72.0	–	–	–	–
Restaurants	SP	SB1	3.	62.0	2.	81.3	–	–	–	–
Restaurants	SP	SB2	3.	73.7	1.	77.2	–	–	–	–
			Target		**Category & Target**		**Target**		**Category & Target**	
			Rank	F_1[%]	Rank	F_1[%]	Rank	F_1[%]	Rank	F_1[%]
Restaurants	EN	SB1	1.	66.9	4.	41.1	3.	67.1	6.	41.1

Table 8.8: Achieved ranks and results (in %) by UWB for all submitted systems.

The baseline algorithm for text-level category (SB2, slot 1)[15] achieves an F_1 score of 96.1% on the Laptops domain and 97.1% on the Restaurants domain for English. When we add the corresponding general class for the given domain (e.g. `RESTAURANT#GENERAL` and `LAPTOP#GENERAL`) the algorithm achieves an F_1 score of 96.9% on the Laptops domain and 99.8% on the Restaurants domain for English.

The baseline algorithm for text-level sentiment polarity (SB2, slot 3)[15] achieves an Accuracy of 86.8% on the Laptops domain and 89.6% on the Restaurants domain for English, while the improved algorithm achieves an Accuracy of 94.5% on the Laptops domain and 97.3% on the Restaurants domain for English.

8.2.9 Conclusion

We competed in 19 constrained experiments and achieved state-of-the-art results in 9 of them. In the other 10 cases we have reached at worst the 4th place. Our unconstrained systems participated in 10 experiments and achieved 5 ranks ranging from the 1st to 3rd place. In other words, we achieved state-of-the-art results in 2 experiments among the unconstrained systems.

[15]Using the sentence-level gold test data.

9 Neural Networks for Sentiment Analysis

This chapter presents the first attempt at using neural networks for sentiment analysis in Czech [Lenc and Hercig, 2016].

We first perform experiments on two English corpora to allow comparability with the existing state-of-the-art methods for sentiment analysis in English. Then we explore the effectiveness of using neural networks on four Czech corpora.

9.1 Introduction

The current approaches to sentiment analysis in English explore various neural network architectures (e.g. [Kim, 2014, Socher et al., 2013, dos Santos and Gatti, 2014]). We try to replicate the results shown in [Kim, 2014] and adapt the proposed architecture to the sentiment analysis task in Czech – a highly inflectional Slavic language. To the best of our knowledge, neural networks have not been used for the sentiment analysis task in Czech.

In this work we will focus on the sentiment polarity task on aspect-level and document-level[1] for Czech and English. In terms of the SemEval 2014 task it is the *Aspect Term Polarity* and *Aspect Category Polarity* (TP and CP) subtasks. In terms of the SemEval 2016 task it is the *Sentence-level Sentiment Polarity* subtask.

Our main goal is to measure the difference between the previous results and the new results achieved by neural network architectures.

[1]For the English RT dataset and Czech Facebook dataset it can be also called the sentence-level.

9.2 Data

In this work we use two types of corpora:

- Aspect-level for the ABSA task and
- Document-level for the sentiment polarity task.

The properties of these corpora are shown in Table 9.1. The English Aspect-level datasets come from the SemEval ABSA tasks. Although we show properties of the datasets from previous years, we report results only on the latest datasets from the SemEval 2016.

We do not use the Czech IT product datasets [Tamchyna et al., 2015] because of its small size and because no results for the sentiment polarity task have been reported using these datasets so far. The Czech Facebook dataset has a label for bipolar sentiment which we discard, similarly to the original publication.

For all experiments we use 10-fold cross validation in cases where there are no designated test and train data splits.

9.3 System

The proposed sentiment classification system can be divided into two modules. The first one serves for data preprocessing and creates the data representation while the second one performs the classification. The classification module utilizes three different neural network architectures. All networks use the same preprocessing.

9.3.1 Data Preprocessing and Representation

The importance of data preprocessing has been proven in many NLP tasks. The first step in our preprocessing chain is removing the accents similarly to [Habernal et al., 2014] and converting the text to lower case. This process may lead to loss of some information but we include it due to the fact that the data we use are collected from the Internet and therefore it may contain grammatical errors, misspellings and could be written either with or without accents. Finally, all numbers are replaced with one common token. We also perform stemming utilizing the High Precision Stemmer [Brychcín and Konopík, 2015].

Aspect-level Sentiment Dataset	Sentences	Average Length	Positive	Negative	Neutral
English 2016 Laptops train + test	3.3k	14	2.1k	1.4k	0.2k
English 2016 Restaurants train + test	2.7k	13	2.3k	1k	0.1k
English 2015 Restaurants train + test	2k	13	1.7k	0.7k	0.1k
Czech Restaurant reviews	2.15k	14	2.6k	2.5k	1.2k
Czech IT product reviews short	2k	6	1k	1k	–
Czech IT product reviews long	0.2k	144	0.1k	0.1k	–
Document-level Sentiment Dataset	**Sentences**	**Average Length**	**Positive**	**Negative**	**Neutral**	**Bipolar**
English RT Movie reviews	10.7k	21	5.3k	5.3k	–	–
Czech CSFD Movie reviews	91.4k	51	30.9k	29.7k	30.8k	–
Czech MALL Product reviews	145.3k	19	103k	10.4k	31.9k	–
Czech Facebook posts	10k	11	2.6k	2k	5.2k	0.2k

Table 9.1: Properties of the aspect-level and document-level corpora in terms of the number of *sentences*, *average length* of sentences (number of words), and numbers of *positive*, *negative*, *neutral* and *bipolar* labels.

The input feature of the neural networks is a sequence of words in the document represented using the one hot encoding. A dictionary is first created from the training set. It contains a specified number of most frequent words. The words are then represented by their indexes in the dictionary. The words that are not in the dictionary are assigned a reserved "Out of dictionary" index. An important issue is the variable length of classified sentences. Therefore, we cut the longer ones and pad the shorter ones to a fixed length. The padding token has also a reserved index. We use dictionary size 20,000 in all experiments. The sentence length was set to 50 in all experiments with document-level sentiment. We set the sequence length to 11 in the aspect-level sentiment experiments.

9.3.2 CNN 1

This network was proposed by Kim [2014]. It is a modification of the architecture proposed in [Collobert et al., 2011]. The first layer is the embedding one. It learns a word vector of fixed length k for each word. We use $k = 300$ in all experiments. It uses one convolutional layer which is composed of a set of filters of size $n \times k$ which means that it is applied on sequence of n words and the whole word vector (k is the length of the word vector). The application of such filters results in a set of feature maps (results after applying the convolutional filters to the input matrix). Kim proposes to use multiple filter sizes ($n = 3, 4, 5$) and utilizes 100 filters of each size. Rectified linear units (Relu) are used as activation function, drop-out rate is set to 0.5 and the mini-batch size is 50. After this step, a max-over-time pooling is applied on each feature map and thus the most significant features are extracted. The selection of one most important feature from each feature map is supposed to ensure invariance to the sentence length. The max pooling layer is followed by a fully connected softmax layer which outputs the probability distribution over labels. There are four approaches to the training of the embedding layer:

- **1)** Word vectors trained from scratch (randomly initialized)
- **2)** Static Word2Vec [Mikolov et al., 2013a] vectors
- **3)** Non-static vectors (initialized by Word2Vec and then fine tuned)
- **4)** Multichannel (both random initialized and pre-trained by Word2Vec).

The hyper-parameters of the network was set on the development part of the SST dataset[2]. We use identical configuration in our experiments to allow

[2]Stanford Sentiment Treebank with neutral reviews removed and binary labels.

comparability. We implemented only the basic – randomly initialized version of word embeddings. Figure 9.1 depicts the architecture of the network.

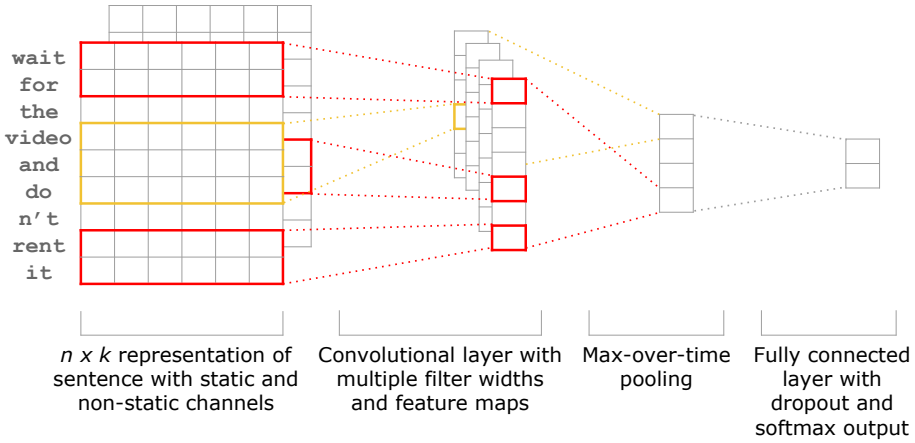


Figure 9.1: Architecture of the convolutional network CNN1

9.3.3 CNN 2

The architecture of this network was designed according to [Lenc and Král, 2016] where it is successfully used for multi-label document classification.

Contrary to the work of Kim [2014] this network uses just one size of the convolutional kernels and not the combination of several sizes. The kernels have only 1 dimension (1D) while Kim have used larger 2 dimensional kernels. It was proven on the document classification task that the simple 1D kernels give better results than the 2D ones.

The input of the network is a vector of word indexes as described in Section 9.3.1. The first layer is an embedding layer which represents each input word as a vector of a given length. The document is thus represented as a matrix with l rows and k columns where k is the length of embedding vectors. The embedding length is set to 300. The next layer is the convolutional one. We use n_c convolution kernels of the size $l_k \times 1$ which means we do 1D convolution over one position in the embedding vector over l_k input words. The size k is set to 3 (aspect-level sentiment) and 5 (document-level sentiment) in our experiments and we use $n_c = 32$ kernels. The following layer performs max pooling over the length $l - l_k + 1$ resulting in n_n $1 \times k$ vectors. The output of this layer is then flattened and connected with the output layer containing either 2 or 3 nodes (number of sentiment labels). Figure 9.2 shows the architecture of the network.

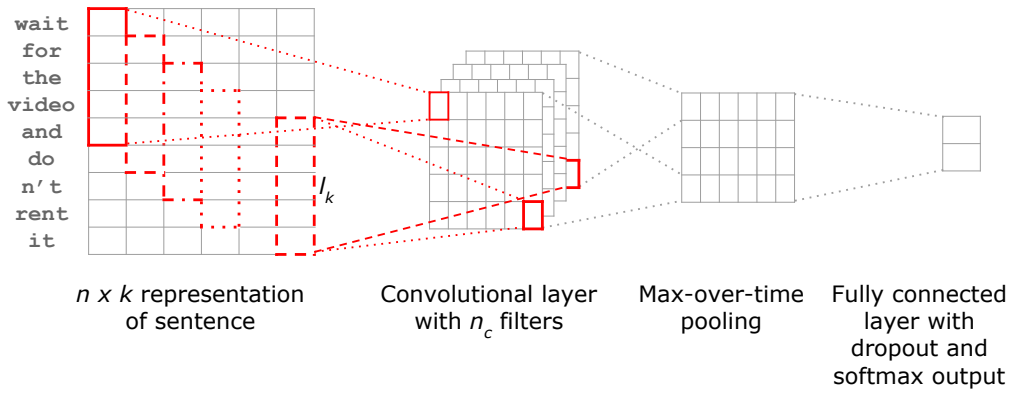


Figure 9.2: Architecture of the convolutional network CNN2

9.3.4 LSTM

The word sequence is the input to an embedding layer same as for the CNNs. We use the embedding length of 300 in all experiments. The word embeddings are then fed to the recurrent LSTM layer with 128 hidden neurons. Dropout rate of 0.5 is then applied and the final state of the LSTM layer is connected with the softmax output layer. The network architecture is depicted in Figure 9.3.

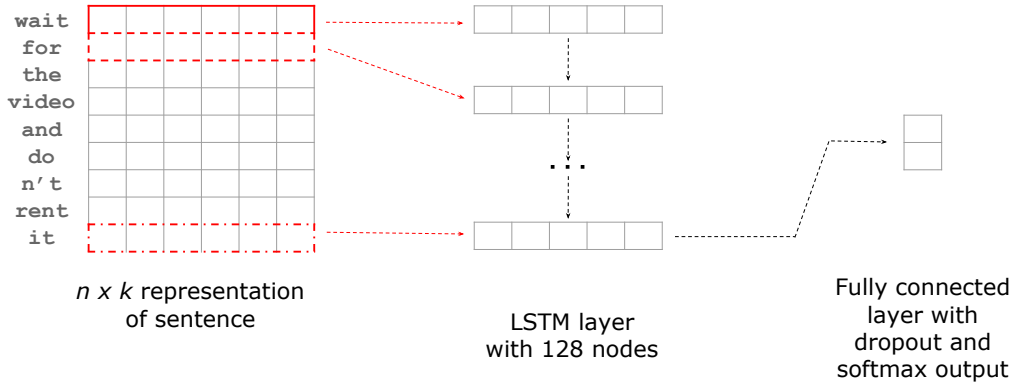


Figure 9.3: Architecture of the LSTM network

9.3.5 Tools

We used Keras [Chollet, 2015] for implementation of all above mentioned neural networks. It is based on the Theano deep learning library [Bergstra et al., 2010]. It has been chosen mainly because of good performance and our previous experience with this tool. All experiments were computed on GPU to achieve reasonable computation times.

9.4 Experiments

Results on RT movie dataset [Pang and Lee, 2005] (10662 sentences, 2 classes) confirm that our implementation works similarly to the original (see Table 9.2).

Description	Results
Kim [2014] randomly initialized	76.1
Kim [2014] best result	**81.5**
CNN1	77.1
CNN2	76.2
LSTM	61.7
Confidence Interval	±0.8

Table 9.2: Accuracy on the English RT movie reviews dataset in %.

We further performed evaluation on the current SemEval 2016 ABSA dataset to allow comparison with the current state-of-the-art methods. These results (see Table 9.3) show that the used neural network architectures are still quite far from the finely tuned state-of-the-art results. However we need to remind the reader that our goal was not to achieve the state-of-the-art results, but to replicate network architectures that are used for sentiment analysis in English as well as some networks utilized for other tasks in Czech.

Description	Restaurants	Laptops
SemEval 2016 best result	**88**	**82**
SemEval 2016 best constrained	**88**	75
CNN1	78	68
CNN2	78	71
LSTM	72	68
Confidence Interval	±3	±3

Table 9.3: Accuracy on the English SemEval 2016 ABSA datasets in %.

Results on the Czech document-level datasets are shown in Table 9.4. For the CSFD movie dataset, results are much worse than the previous work. We believe that this is due to the number of words used for representation. We used 50 words in all experiments and it may not suffice to fully understand the review. This is supported by the fact that the global target context [Brychcín and Habernal, 2013] helps to improve the results by 1.5%.

We applied three types of neural networks to the term polarity (TP) and class polarity (CP) tasks and evaluated them on the Czech aspect-level

restaurant reviews dataset. The results in Table 9.5 show markedly inferior results compared to the state-of-the-art results 72.5% for the TP and 75.2% for the CP tasks in [Hercig et al., 2016b]. Best results are achieved using the combination of words and stemms as input.

Description	CSFD Movies	MALL Products	Facebook Posts
Supervised Machine Learning[3]	78.5	75.3	**69.4**
Semantic Spaces[4]	**80**	**78**	-
Global Target Context[5]	**81.5**	-	-
CNN1 stemmed	70.8	74.4	**68.9**
CNN2 stemmed	71.0	75.5	**69.4**
LSTM stemmed	70.2	73.5	67.6
Confidence Interval	±0.3	±0.2	±1.0

Table 9.4: F-measure on the Czech document-level datasets in %.

The inputs of the networks are one-hot vectors created from words in the context window of the given aspect term. We used five words in each direction of the searched aspect term resulting in window size 11. We do not use any weighting to give more importance to the closest words as in [Hercig et al., 2016b].

For statistical significance testing, we report confidence intervals at α 0.05.

CNN1 and CNN2 present similar results although the average best performance is achieved by the CNN2 architecture. The LSTM architecture consistently underperforms, we believe that this is due to the basic architecture model.

9.5 Conclusion and Future Work

In this work we have presented the first attempts to classify sentiment of Czech sentences using a neural network. We evaluated three architectures.

We first performed experiments on two English corpora mainly to allow comparability with existing work for sentiment analysis in English.

We have further experimented with three Czech corpora for document-level sentiment analysis and one corpus for aspect-based sentiment analysis.

[3] [Habernal et al., 2013]
[4] [Habernal and Brychcín, 2013]
[5] [Brychcín and Habernal, 2013]

	Term Polarity			Class Polarity		
Description \ Features	W	S	W+S	W	S	W+S
CNN1	65	66	**67**	65	66	68
CNN2	64	65	66	67	68	**69**
LSTM	61	62	62	65	65	64
Confidence Interval	±2	±2	±2	±2	±2	±2

Table 9.5: Accuracy on the Czech aspect-level restaurant reviews dataset in %. *W* denotes words, *S* stemms and *W+S* the combination of these inputs.

The experiments proved that the tested networks don't achieve as good results as the state-of-the-art approaches. The most promising results were obtained when using the CNN2 architecture. However, regarding the confidence intervals, we can consider the performance of the architectures rather comparable.

The results show that Czech is much more complicated to handle when determining sentiment polarity. This can be caused by various properties of Czech language that differ from English (e.g. double negative, sentence length, comparative and superlative adjectives, or free word order). Double or multiple negatives are grammatically correct ways to express negation in Czech while in English double negative is not acceptable in formal situations or in writing. Thus the semantic meaning of sentences with double or multiple negatives is hard to determine. In English comparative and superlative forms of adjectives are created by adding suffixes[6] while in Czech suffixes and prefixes are used. Informal texts can contain mixed irregular adjectives with prefixes and/or suffixes, thus making it harder to determine the semantic meaning of these texts. The free word order can also cause difficulties to train the models because the same thing may be expressed differently.

However, it must be noted that the compared approaches utilize much richer information than our basic features fed to the neural networks. The neural networks were also not fine-tuned for the task. Therefore we believe that there is much room for further improvement and that neural networks can reach or even outperform the state-of-the-art results.

We consider this paper to be the initial work on sentiment analysis using neural networks in Czech.

[6]excluding irregular and long adjectives

10 Sarcasm Detection

This chapter presents our machine learning approach to sarcasm detection on Twitter in two languages – English and Czech. Although there has been some research in sarcasm detection in languages other than English (e.g. Dutch, Italian, and Brazilian Portuguese), our work [Ptáček et al., 2014] is the first attempt at sarcasm detection in the Czech language.

10.1 Introduction

Sentiment analysis on social media has been one of the most targeted research topics in NLP in the past decade, as shown in several surveys [Liu and Zhang, 2012, Tsytsarau and Palpanas, 2012]. Since the goal of sentiment analysis is to automatically detect the polarity of a document, misinterpreting irony and sarcasm represents a big challenge [Davidov et al., 2010].

As there is only a weak boundary in meaning between irony, sarcasm and satire [Reyes et al., 2012], we will use only the term sarcasm in this chapter. Bosco et al. [2013] claim that *"even if there is no agreement on a formal definition of irony, psychological experiments have delivered evidence that humans can reliably identify ironic text utterances from an early age in life."* We have thus decided to rely on the ability of our human annotators to manually label sarcastic tweets to train our classifiers. Sarcasm generally reverses the polarity of an utterance from positive or negative into its opposite, which deteriorates the results of a given NLP task. Therefore, correct identification of sarcasm can improve the performance.

The issue of automatic sarcasm detection has been addressed mostly in English, although there has been some research in other languages, such as Dutch [Liebrecht et al., 2013], Italian [Bosco et al., 2013], or Brazilian Portuguese [Vanin et al., 2013]. To the best of our knowledge, no research has been conducted in Czech or other Slavic languages. These languages are challenging for many NLP tasks because of their rich morphology and

syntax. This has motivated us to focus our current research on both English and Czech.

Majority of the existing state-of-the-art techniques are language dependent, which rely on language-specific lexical resources. Since no such resources are available for Czech, we adapt some language-independent methods and also apply various preprocessing steps for sentiment analysis proposed by Habernal et al. [2013].

Research Questions

(1) To what extent can the language-independent approach compete with methods based on lexical language-dependent resources? **(2)** Is it possible to reach good agreement on annotating sarcasm and what typical text properties on Twitter are important for sarcasm detection? **(3)** What is the best preprocessing pipeline that can boost performance on highly-flective Czech language and what types of features and classifiers yield the best results?

10.2 Our Approach

This chapter presents the first attempt at sarcasm detection in the Czech language, in which we focus on supervised machine learning approaches and evaluate their performance. We selected various n-grams, including unigrams, bigrams, trigrams with frequency greater than three [Liebrecht et al., 2013], and a set of language-independent features, including punctuation marks, emoticons, quotes, capitalized words, character n-grams and skip-grams [Reyes et al., 2013] as our baselines.

10.2.1 Classification

Our evaluation was performed using the Maximum Entropy (MaxEnt) and SVM classifiers. We used *Brainy* – a Java framework for machine learning [Konkol, 2014] – with default settings (the linear kernel for SVM). All experiments were conducted in the 5-fold cross validation manner similar to [Davidov et al., 2010, González-Ibáñez et al., 2011]. Our motivation to test multiple classifiers stemmed also from related works which mostly test more than one classifier. On the other hand, the choice between state-of-the-art linear classifiers might not be much of importance, as the most important is the feature engineering.

10.2.2 Features

For our evaluation we used the most promising language-independent features from the related work and POS related features. Feature sets used in our evaluation are described below.

N-gram

Character n-gram We used character n-gram features [Blamey et al., 2012]. We set the minimum occurrence of a particular character n-gram to either 5 or 50, in order to prune the feature space. Our character feature set contains 3-grams to 6-grams.

N-gram We used word unigrams, bigrams and trigrams as binary features. The feature space is pruned by the minimum n-gram occurrence set to 3 [Liebrecht et al., 2013].

Skip-bigram Instead of using sequences of adjacent words (n-grams) we used skip-grams [Guthrie et al., 2006], which skip over arbitrary gaps. Reyes et al. [2013] consider skip-bigrams with 2 or 3 word skips and remove skip-grams with a frequency ≤ 20.

Pattern

Pattern Patterns composed of high frequency words (HFWs)[1] and content words (CWs)[2] used by Davidov et al. [2010]. Pattern must contain at least one high frequency word. The patterns contain 2-6 HFWs and 1-6 CWs. We set the minimum occurrence of a particular pattern to 5.

Word-shape pattern We tried to improve pattern features by using word-shape classes for content words. We assign words into one of 24 classes[3] similar to the function specified in [Bikel et al., 1997].

Part Of Speech

POS characteristics We implemented various POS features, e.g. the number of nouns, verbs, and adjectives [Ahkter and Soria, 2010], the ratio of nouns to adjectives and verbs to adverbs [Kouloumpis et al., 2011], and number of negative verbs obtained from POS tags.

[1] A word whose corpus frequency is more than 1000 words per million plus all punctuation characters.

[2] A word whose corpus frequency is less than 1000 words per million.

[3] We use edu.stanford.nlp.process.WordShapeClassifier with the WORDSHAPE-CHRIS1 setting.

POS word-shape Unigram feature consisting of POS and word-shape (see Word-shape pattern). The feature space is pruned by the minimum occurrence set to 5.

POS n-gram Direct use of POS n-grams has not shown any significant improvement in sentiment analysis but it may improve the results of sarcasm detection. We experimented with 3-grams to 6-grams with the minimum n-gram occurrence set to 5.

Others

Emoticons We used two lists of positive and negative emoticons [Montejo-Ráez et al., 2012]. The feature captures the number of occurrences of each class of emoticons within the text.

Punctuation-based We adapted punctuation-based features proposed by Davidov et al. [2010]. This feature set consists of number of words, exclamation marks, question marks, quotation marks and capitalized words normalized by dividing them by the maximal observed value multiplied by the averaged maximal value of the other feature groups.

Pointedness-based Pointedness was used by Reyes et al. [2013] to distinguish irony. It focuses on explicit marks which should reflect a sharp distinction in the information that is transmitted. The presence of punctuation marks, emoticons, quotes and capitalized words has been considered.

Extended Pointedness This feature captures the number of occurrences of punctuation marks and emoticons as well as the number of words, exclamation marks, question marks, quotation marks and capitalized words normalized by maximal observed value.

Word-case We implemented various word-case features that include e.g. the number of upper cased words, number of words with first letter capital normalized by number of words and number of upper cased characters normalized by number of words.

10.3 Evaluation Datasets

We collected datasets using *Twitter Search API* and *Java Language Detector*[4]. We collected 140,000 Czech and 780,000 English tweets, respectively.

[4] <http://code.google.com/p/jlangdetect/>

Due to lack of support for the Czech language on Twitter, we used the *Twitter Search API* parameter *geocode* to acquire tweets posted near Prague. For the English dataset we also collected tweets with the #sarcasm hashtag. Czech users generally don't use the sarcasm ("`#sarkasmus`") or irony ("`#ironie`") hashtag variants[5], thus we had to annotate the Czech dataset manually. The final label distribution in datasets is shown in Table 10.3.

10.3.1 Filtering and Normalization

All user, URL and hashtag references in tweets have been replaced by *"user"*, *"link"* and *"hashtag"* respectively. We also removed all tweets starting with "RT" because they refer to previous tweets and tweets containing just combinations of user, link, *"RT"* and hashtags without any additional words.

Tokenization of tweets requires proper handling of emoticons and other special character sequences typical on Twitter. The *Ark-tweet-nlp tool* [Gimpel et al., 2011] offers precisely that and although it was developed and tested in English, it yields satisfactory results in Czech as well.

Czech is a highly flective language and uses a lot of diacritics. However some Czech users type only the unaccented characters.[6] Preliminary experiments showed that removing diacritics yields better results, thus we removed diacritics from all tweets.

10.3.2 Czech Dataset Annotation

Firstly we conducted an experiment to determine whether to annotate the original data or the normalized data. We selected two sample sets of 50 tweets containing Czech sarcasm (`#sarkasmus`) and irony (`#ironie`) hashtags and other tweets. One annotator obtained the original data while the other got the normalized data from the first sample set. We then tried to give both annotators the original data from the first sample set and finally we gave them both the normalized data from the second sample set. Table 10.1 shows the difficulty of sarcasm identification without the knowledge hidden in hashtags, user and links. The most promising results come from the annotation of the original data, thus the rest of the data are annotated in this manner.

[5]We found only 10 tweets with sarcasm hashtag ("`#sarkasmus`") and 100 tweets with irony hashtag ("`#ironie`") in 140,000 collected tweets.

[6]Approximately 10% of collected tweets were without any diacritics.

Normalized	Normalized		
	Tag	**n**	**s**
	n	35	10
	s	0	5
Cohen's κ:			0.412

Original	Normalized		
	Tag	**n**	**s**
	n	19	10
	s	5	16
Cohen's κ:			0.404

Original	Original		
	Tag	**n**	**s**
	n	25	4
	s	3	18
Cohen's κ:			0.715

Table 10.1: Confusion matrices and annotation agreement (Cohen's κ) between two annotators using original or normalized data.

We randomly selected 7,000 tweets from the collected data for annotation. The annotators were given just simple instructions without an explicit sarcasm definition (see Section 10.1): *"A tweet is considered sarcastic when its content is intended ironically / sarcastically without anticipating further information. Offensive utterances, jokes and ironic situations are not considered ironic / sarcastic."*

The complete dataset of 7,000 tweets was independently annotated by two annotators. The inter-annotator agreement (Cohen's κ) between the two annotators is 0.54. They disagreed on 403 tweets. To resolve these conflicts we used a third annotator.

The third annotator has been instructed the same way as the other two. The final κ agreement was measured between the first two annotators, thus it was not affected by the third annotator. Kappa agreements measured on the conflicted states (403 tweets) were 0.4 (annotator 1 vs. annotator 3) and 0.6 (annotator 2 vs. annotator 3).

Preprocessing

Preprocessing steps for handling social media texts in Czech were explored in [Habernal et al., 2013]. The preprocessing diagram and its variants is depicted in Table 10.2. Overall, there are various possible preprocessing *"pipe"* configurations including *"Basic"* pipeline consisting of tokenizing and POS-tagging only. We adapted all their preprocessing pipelines. However, as the number of combinations would be too large, we report only the settings with better performance.

10.3.3 English Dataset

We collected 780,000 (130,000 sarcastic and 650,000 non-sarcastic) tweets in English. The `#sarcasm` hashtag was used as an indicator of sarcastic tweets.

***"Basic"* pipe**	**Pipe 2**	**Pipe 3**
Tokenizing: ArkTweetNLP		
POS tagging: PDT		
–	**Stem:** no (Sn) / light (Sl) / HPS (Sh)	
–	**Stopwords removal**	
–	–	**Phonetic:** eSpeak (Pe)

Table 10.2: The preprocessing pipes for Czech (top-down). Combinations of methods are denoted using the appropriate labels, e.g. *"Sn"* means *1. tokenizing, 2. POS-tagging, 3. no stemming and 4. removing stopwords.* eSpeak stands for a phonetic transcription to International Phonetic Alphabet, which should reduce the effects of grammar mistakes and misspellings.

From this corpus we created two distributional scenarios based on the work of Reyes et al. [2013]. Refer to Table 10.3 for the final statistics of the dataset. Part of speech tagging was done using the *Ark-tweet-nlp tool* [Gimpel et al., 2011].

Dataset \ Tweets	Sarcastic	Non-sarcastic
Czech	325	6,675
English Balanced	50,000	50,000
English Imbalanced	25,000	75,000

Table 10.3: The tweet distributions in datasets.

10.4 Results

For each preprocessing pipeline (refer to table 10.2) we assembled various sets of features and employed two classifiers. Accuracy (micro F-measure) tends to prefer performance on dominant classes in highly unbalanced datasets [Manning et al., 2008], thus we chose macro F-measure as the evaluation metric [Forman and Scholz, 2010], as it allows us to compare classification results on different datasets. For statistical significance testing, we report confidence intervals at α 0.05. Another applicable methods would be i.e. two-matched-samples t Test or McNemar's test [Japkowicz and Shah, 2011].

Feature Set \ Pipeline	Basic	Sh	ShPe	Sl	SlPe	Sn	SnPe
Baseline 1 (B1): n-gram	54.8	55.3	55.2	55.0	55.0	54.4	55.3
B1 + pattern	55.1	54.4	54.7	55.1	54.8	54.2	54.5
B1 + word-shape pattern	54.6	54.8	55.2	54.4	55.0	54.8	55.1
B1 + punctuation-based	54.7	48.8	48.8	48.8	48.8	53.8	55.5
B1 + pointedness	55.0	54.7	54.7	55.0	55.9	54.8	54.9
B1 + extended pointedness	54.5	48.8	48.8	48.8	48.8	54.7	54.6
B1 + POS n-gram	53.4	54.1	54.2	55.3	55.1	54.2	53.9
B1 + POS word-shape	55.0	55.6	55.2	54.8	54.6	55.8	54.4
B1 + skip-bigram	54.2	54.8	54.2	54.7	**56.0**	54.6	54.4
B1 + POS char. + emot.	55.5	54.7	55.6	55.2	55.4	55.2	53.9
B1 + POS char. + emot. + word-case	53.8	**56.4**	55.5	54.6	55.3	55.9	55.3
Character n-gram (3-6, min. occ. > 5)	53.0	52.7	53.2	53.9	54.7	52.0	53.2
Baseline 2 (B2)	55.0	55.2	55.4	**56.8**	**56.2**	54.7	54.0
B2 + FS1	52.3	48.8	48.8	48.8	48.8	52.0	52.9
B2 + FS1 + FS2	53.0	48.8	48.8	48.8	48.8	52.2	53.6
B2 + pattern	55.3	55.4	55.7	**56.9**	**56.6**	54.4	53.6
B2 + POS word-shape	55.5	55.8	55.4	**57.0**	**56.3**	55.3	54.7
B2 + POS char. + emot. + word-case	**56.1**	55.7	55.7	**56.9**	**56.1**	55.0	54.3

Table 10.4: Results on the Czech dataset with the MaxEnt classifier. Macro F-measure, 95% confidence interval $\approx \pm 1.2$. Best results are in bold. **B2**: character n-gram (3-5, min. occurrence > 50) + skip-bigram + pointedness; **FS1**: character n-gram (3-6, min. occurrence > 5) + extended pointedness; **FS2**: POS word-shape + pattern + POS characteristics + emoticons + word-case.

10.4.1 Czech

Tables 10.4 and 10.5 show the results on the Czech dataset. The best result (F-measure 0.582) was achieved by the SVM classifier and a feature set enriched with patterns, utilizing stopwords removal and phonetic transcription in the preprocessing step.

The importance of the appropriate preprocessing techniques for Czech is evident from the improvement of results for various feature sets, e.g. the best result for *"Basic"* pipeline (see line *"B2 + pattern"*). Both baselines show improvement on most preprocessing pipelines. The most significant difference is visible on the second baseline with the MaxEnt classifier and the *"Sl"* pipeline where the F-measure is 0.018 higher than the *"Basic"* pipeline with no addi-

Feature Set \ Pipeline	Basic	Sh	ShPe	Sl	SlPe	Sn	SnPe
Baseline 1 (B1): n-gram	55.8	54.6	54.5	54.6	55.5	56.0	53.9
B1 + pattern	55.6	54.0	54.3	54.6	55.7	55.4	55.6
B1 + word-shape pattern	54.9	55.0	53.8	55.2	55.1	55.4	55.3
B1 + punctuation-based	55.8	48.8	48.8	48.8	48.8	55.7	53.7
B1 + pointedness	55.9	54.5	53.1	54.6	54.3	55.4	54.6
B1 + extended pointedness	56.5	48.8	48.8	48.8	48.8	55.8	56.9
B1 + POS n-gram	54.0	54.1	54.0	54.7	53.4	54.5	53.9
B1 + POS word-shape	55.2	56.4	55.9	55.1	56.0	56.1	55.0
B1 + skip-bigram	55.9	55.3	54.8	55.4	55.0	56.1	55.2
B1 + POS char. + emot.	55.9	54.5	54.1	54.6	54.2	56.7	55.8
B1 + POS char. + emot. + word-case	55.6	54.5	54.3	55.1	55.5	56.3	56.4
Character n-gram (3-6, min. occ. > 5)	54.6	53.6	53.3	55.2	53.6	53.4	54.9
Baseline 2 (B2)	55.9	56.4	56.3	**57.0**	56.2	**57.1**	55.8
B2 + FS1	52.2	48.8	48.8	48.8	48.8	53.1	52.7
B2 + FS1 + FS2	54.0	48.8	48.8	48.8	48.8	54.4	54.3
B2 + pattern	56.8	**57.0**	56.7	56.5	**57.5**	**57.1**	**58.2**
B2 + POS word-shape	56.5	56.3	**57.2**	56.4	56.1	56.3	**57.8**
B2 + POS char. + emot. + word-case	56.2	55.7	55.8	56.0	56.0	**57.0**	56.0

Table 10.5: Results on the Czech dataset with the SVM classifier. Macro F-measure, 95% confidence interval $\approx \pm 1.2$. Best results are in bold. **B2**: character n-gram (3-5, min. occurrence > 50) + skip-bigram + pointedness; **FS1**: character n-gram (3-6, min. occurrence > 5) + extended pointedness; **FS2**: POS word-shape + pattern + POS characteristics + emoticons + word-case.

tional preprocessing. The n-gram baseline was significantly outperformed by the SVM classifier with feature sets *"B1 + POS characteristics + Emoticons + Word-case"* and *"B1 + extended pointedness"* on the *"SnPe"* pipeline.

Error Analysis

To get a better understanding of the limitations of our approach, we inspected 100 random tweets from the Czech dataset, which were wrongly classified by the SVM classifier with the best feature combination. We found 48 false positives and 52 false negatives. The annotators disagreed upon 10% of these tweets.

Non-sarcastic tweets were often about news, reviews, general information

and user status updates. In most of the difficult cases of true negatives, the tweet contains a question, insult, opinion or wordplay.

Understanding sarcasm in some tweets was often bound with broader common knowledge (e.g. about news or celebrities), the context known only to the author or authors opinion. Another difficulty poses subtle or sophisticated expression of sarcasm such as *"I'm not sure whether you didn't overdo a bit the first part of the renovation - the demolition. :)"* [7] or *"Conservatism, once something is in the school rules, it must be followed, forever, otherwise anarchy will break out and traditional values will die."* [8]

10.4.2 English

The results on both balanced and imbalanced English datasets are presented in Table 10.6. In most cases the MaxEnt classifier significantly outperforms the SVM classifier. The combination of majority of features (*"B2 + FS1 + FS2"*) with the MaxEnt classifier yields the best results for both balanced and imbalanced dataset distributions. This suggests that these features are coherent. While no single feature captures the essence of sarcasm, all features together provide useful linguistic information for detecting sarcasm at textual level.

Balanced Distribution Both baselines were surpassed by various combinations of feature sets with the MaxEnt classifier, although in some cases very narrowly (*"B1 + punctuation-based"* and *"B1 + pointedness"* feature sets). Although the SVM classifier has slightly worse results, it still performs reasonably, and we even recorded significant improvement over the baseline for *"B1 + POS word-shape"*. The best results were achieved using the MaxEnt classifier with *"B2 + FS1 + FS2"* (F-measure 0.947) and *"B1 + word-shape pattern"* (F-measure 0.943) feature sets.

Imbalanced Distribution However, data in the real world do not necessarily resemble the balanced distribution. Therefore we have also performed the evaluation on an imbalanced distribution. The MaxEnt classifier clearly achieves the best results. This experiment indicates that combinations of features *"B2 + FS1 + FS2"* (F-measure 0.924) and *"B1, word-shape pattern"* (F-measure 0.920) yields the best results for both balanced and imbalanced dataset distribution.

[7] *"Jestli jste tu první část rekonstrukce - demolici - trochu nepřehnali . :)"*

[8] *"Konzervatismus , když je to jednou ve školním řádu , tak se to musí dodržovat , a to navždy , jinak vypukne anarchie a tradiční hodnoty zemřou ."*

Dataset	Balanced				Imbalanced			
Classifier	MaxEnt		SVM		MaxEnt		SVM	
Feature set \ Results	Fm	CI	Fm	CI	Fm	CI	Fm	CI
Baseline 1 (B1): n-gram	**93.28**	0.16	92.86	0.16	90.76	0.18	90.44	0.18
B1 + pattern	**94.25**	0.14	**93.13**	0.16	91.86	0.17	90.22	0.18
B1 + word-shape pattern	**94.33**	0.14	**93.17**	0.16	**92.01**	0.17	90.35	0.18
B1 + punctuation-based	**93.32**	0.15	92.84	0.16	90.72	0.18	90.43	0.18
B1 + pointedness	**93.29**	0.16	92.99	0.16	91.00	0.18	90.07	0.19
B1 + extended pointedness	**93.68**	0.15	92.61	0.16	91.07	0.18	89.89	0.19
B1 + POS n-gram	**93.66**	0.15	92.64	0.16	91.20	0.18	89.85	0.19
B1 + POS word-shape	**93.96**	0.15	**93.19**	0.16	91.41	0.17	90.51	0.18
B1 + skip-bigram	**93.63**	0.15	**93.17**	0.16	90.99	0.18	90.48	0.18
B1 + POS char. + emot.	**93.97**	0.15	91.66	0.17	91.69	0.17	89.39	0.19
B1 + POS char. + emot.+word-case	**93.96**	0.15	91.54	0.17	91.61	0.17	88.89	0.19
Character n-gram (3-6, min. occ. > 5)	**93.01**	0.16	91.73	0.17	90.36	0.18	88.81	0.20
Baseline 2 (B2)	92.81	0.16	91.67	0.17	90.65	0.18	88.70	0.20
B2 + FS1	**93.82**	0.15	91.56	0.17	91.21	0.18	88.73	0.20
B2 + FS1 + FS2	**94.66**	0.14	91.39	0.17	**92.37**	0.16	88.62	0.20
B2 + pattern	**93.60**	0.15	91.66	0.17	90.86	0.18	88.82	0.20
B2 + POS word-shape	**93.20**	0.16	91.65	0.17	90.82	0.18	88.74	0.20
B2 + POS char. + emot. + word-case	**93.21**	0.16	91.07	0.18	89.98	0.19	88.40	0.20

Table 10.6: Results on the English dataset with the MaxEnt and SVM classifiers. Macro F-measure (Fm) and 95% confidence interval (CI) are in %. Best results are in bold. **B2**: character n-gram (3-5, min. occurrence > 50) + skip-bigram + pointedness; **FS1**: character n-gram (3-6, min. occurrence > 5) + extended pointedness; **FS2**: POS word-shape + pattern + POS characteristics + emoticons + word-case.

10.4.3 Discussion

To explain the huge difference in the performance between English and Czech, we conducted an additional experiment in English. We sampled the *"big-data"* English corpus (100k Tweets) to obtain the same distribution as on the *"small-data"* Czech corpus (325 sarcastic and 6,675 non-sarcastic Tweets). Feature combination *"B2 + FS1 + FS2"* achieves an F-measure of 0.734 ± 0.01 (MaxEnt classifier) and 0.729 ± 0.01 (SVM). This performance drop shows that the amount of training data plays a key role (≈ 0.92 on *"big-data"* vs. ≈ 0.73 on *"small-data"*). However, these results are still significantly better than in Czech (≈ 0.58). This demonstrates that Czech is a challenging language in sarcasm detection, as in other NLP tasks.

In addition, we also experimented with the Naive Bayes classifier and with delta TF-IDF feature variants [Martineau and Finin, 2009, Paltoglou and Thelwall, 2010] in both languages. However, the performance was not satisfactory in comparison with the reported results.

10.5 Conclusions

We investigated supervised machine learning methods for sarcasm detection on Twitter. As a pilot study for sarcasm detection in the Czech language, we provide a large human-annotated Czech Twitter dataset containing 7,000 tweets with inter-annotator agreement $\kappa = 0.54$.

We created a large English Twitter corpus of 780k automatically-labeled tweets. The dataset consists of a balanced distribution and an imbalanced distribution, each containing 100,000 tweets, where the hashtag `#sarcasm` was used as an indicator of sarcastic tweets.

The novel contributions of our work include the extensive evaluation of two classifiers with various combinations of feature sets on both the Czech and English datasets as well as a comparison of different preprocessing techniques for the Czech dataset.

Our approaches significantly outperformed both baselines adapted from related work[9] in English and achieved F-measure of 0.947 and 0.924 on the balanced and imbalanced datasets, respectively.[10]

The best result on the Czech dataset was achieved by the SVM classifier with the feature set enriched with patterns yielding F-measure 0.582.

The whole project and the datasets are available at `<http://liks.fav.zcu.cz/sarcasm/>` under GPL and Creative Commons BY-NC-SA license.

[9]Word unigrams, bigrams, trigrams [Liebrecht et al., 2013] and a set of language-independent features (punctuation marks, emoticons, quotes, capitalized words, character n-grams, and skip-grams) [Reyes et al., 2013].

[10]Note that the best result (F-measure 0.715 on the balanced distribution and F-measure 0.533 on the imbalanced distribution) from the related work was achieved by Reyes et al. [2013] using decision trees classifier.

11 Sentiment Analysis of Figurative Language

Figurative language such as irony, sarcasm, and metaphor is considered a significant challenge in sentiment analysis. These figurative devices can sculpt the affect of an utterance and test the limits of sentiment analysis of supposedly literal texts. In this chapter we explore the effect of figurative language on sentiment analysis presented in [Hercig and Lenc, 2017].

11.1 Introduction

Recently there have been several experiments with sarcasm detection e.g. [Ptáček et al., 2014, Ghosh and Veale, 2016, Zhang et al., 2016, Poria et al., 2016]. Although these works succeeded in their goal to detect variations of sarcasm, one final step is still missing – the evaluation of sentiment analysis with and without additional sarcasm indicators. There have been attempts at investigating the impact of sarcasm on sentiment analysis [Maynard and Greenwood, 2014] or thorough analysis of hashtags indicating sarcastic tweets [Sulis et al., 2016]. However, the impact of figurative language (including sarcasm) on sentiment analysis has not yet been studied in depth.

11.2 Datasets

We use the dataset from SemEval-2015 Task 11 [Ghosh et al., 2015] for training and evaluation. Table 11.1 shows the mean polarity and the original estimated tweet distributions[1]. The category type labels refer to the authors' expectations of tweet category types in each segment of the dataset. To ensure the validity of the task, the authors added the category *other* to the test dataset.

[1] In the original publication there were some typos, we show the recalculated statistics.

Table 11.2 contains the same statistics for our collected datasets[2]. We separated data into the category types by using the harvesting criteria for the datasets' collection (e.g. the *#irony* hashtag)[3]. Table 11.3 shows the detailed sentiment polarity distributions. The training data were provided with rounded integer values and floating point values. However when we rounded the real-valued scores we got different counts for individual polarity values. This issue corresponds to the *Train* data columns *int* and *rounded*. In our experiments we use *rounded* values wherever it is possible.

To compensate for the missing *other* category in the training data of SemEval-2015 Task 11, we use the dataset from SemEval-2015 Task 10B [Rosenthal et al., 2015] as additional training data. We were able to download approximately 75.7% of the training data and 78.6% of the test data (see Table 11.4).

For the SemEval-2015 Task 10B we evaluate on the test data and the sarcasm dataset[4] from the same task in SemEval-2014.

Type	Train		Test		Trial	
	Mean Polarity	# Tweets	Mean Polarity	# Tweets	Mean Polarity	# Tweets
Sarcasm	-2.25	5000	-2.02	1200	-1.94	746
Irony	-1.70	1000	-1.87	800	-1.35	81
Metaphor	-1.49	2000	-0.77	800	-0.34	198
Other	–	–	-0.26	1200	–	–
Overall	-1.99	8000	-1.21	4000	-1.89	1025

Table 11.1: The tweet distributions and mean polarity in SemEval-2015 Task 11 datasets.

11.3 Convolutional Neural Network

The architecture of the proposed CNN is depicted in Figure 11.1. We use similar architecture to the one proposed by Brychcín and Král [2014]. The input layer of the network receives a sequence of word indices from a dictionary. The input vector must be of a fixed length. We solve this issue by padding the input sequence to the maximum tweet length denoted M. A special *"PADDING"* token is used for this purpose. The embedding layer maps

[2]Note that we were unable to download the whole Trial dataset due to perishability of tweets.

[3]Separating tweets into category types is a rule-based approach.

[4]We were not able to download sufficient amount of tweets for the sarcasm dataset from SemEval-2015 Task 10B.

Type	Train		Test		Trial	
	Mean Polarity	# Tweets	Mean Polarity	# Tweets	Mean Polarity	# Tweets
Sarcasm	-2.25	4895	-2.05	1107	-2.00	612
Irony	-1.70	1424	-1.85	763	-1.98	23
Metaphor	-1.49	1681	-0.85	878	-0.67	91
Other	–	–	-0.33	1252	–	–
Overall	-1.99	8000	-1.21	4000	-1.83	726

Table 11.2: The tweet distributions and mean polarity in SemEval-2015 Task 11 datasets by hashtags.

Value	Test	Train (int)	Train (rounded)	Trial orig.	Trial downl.
-5	4	0	6	6	4
-4	100	361	364	90	56
-3	737	2954	2971	403	282
-2	1541	2911	2934	255	180
-1	680	909	861	87	67
0	298	347	345	50	40
1	169	164	165	51	39
2	155	197	197	41	29
3	201	106	106	32	23
4	111	49	49	9	6
5	4	2	2	1	0
SUM	4000	8000	8000	1025	726

Table 11.3: The tweet sentiment polarity distributions in SemEval-2015 Task 11.

the word indices to the real-valued embedding vectors of length L. The convolutional layer consists of N_C kernels containing $k \times 1$ units and uses rectified linear unit (ReLU) activation function. The convolutioanl layer is followed by a max-pooling layer and dropout for regularization. The max-pooling layer takes maxima from patches with dimensions $(M - k + 1) \times 1$. The output of the max-pooling layer is fed into a fully-connected layer. The fully connected layer is optionally concatenated with the additional `category-type-binary-input` layer that adds the information about hashtags used in the tweet. The output layer is connected to this layer and has just one neuron serving as a regressor.

In our experimental setup we use the embedding dimension $L = 300$

Corpus	Positive	Negative	Neutral	Total	Downloaded
Twitter2015-train	3,640	1,458	4,586	9,684	7,326 (76%)
Twitter2015-test	1,038	365	987	2,390	1,878 (79%)
Twitter2014-sarcasm	33	40	13	86	86 (100%)

Table 11.4: The tweet polarity distributions in SemEval-2015 Task 10B.

and $N_C = 40$ convolutional kernels with 5×1 units. The penultimate fully-connected layer contains 256 neurons connected with the optional `category-type-binary input` with 4 neurons. We train the network using adaptive moment estimation optimization algorithm [Kingma and Ba, 2014]. Mean square error is used as loss function.

11.4 Experiments

We perform regression experiments on the 11-point scale (-5, .., 0, .., 5) for the SemEval-2015 task 11 and classification into positive, negative, and neutral classes for SemEval-2015 task 10B.

11.4.1 Preprocessing

The same preprocessing has been done for all datasets. We use UDPipe [Straka et al., 2016] with English Universal Dependencies 1.2 models for POS tagging and lemmatization. Tokenization has been done by TweetNLP tokenizer [Owoputi et al., 2013]. Preliminary experiments have shown that lower-casing the data achieves slightly better results, thus all the experiments are performed with lower-cased data. We further replace all user mentions with the token *"@USER"* and all links with the token *"$LINK"*.

11.4.2 Regression

Regression has been done using CNN (Section 11.3) and Weka 3.6.6 [Hall et al., 2009] with the M5P decision tree regression. We use the SemEval-2015 task 11 scorer to evaluate our results. Used features are unigrams with more than two occurrences. We map the additional training data from SemEval-2015 task 10B (-1 negative, 0 neutral, 1 positive) to the 11-point scale by using multiplier 4 (-4 negative, 0 neutral, 4 positive). This corresponds to our intuition that the positive and negative class should contain strong polarity values.

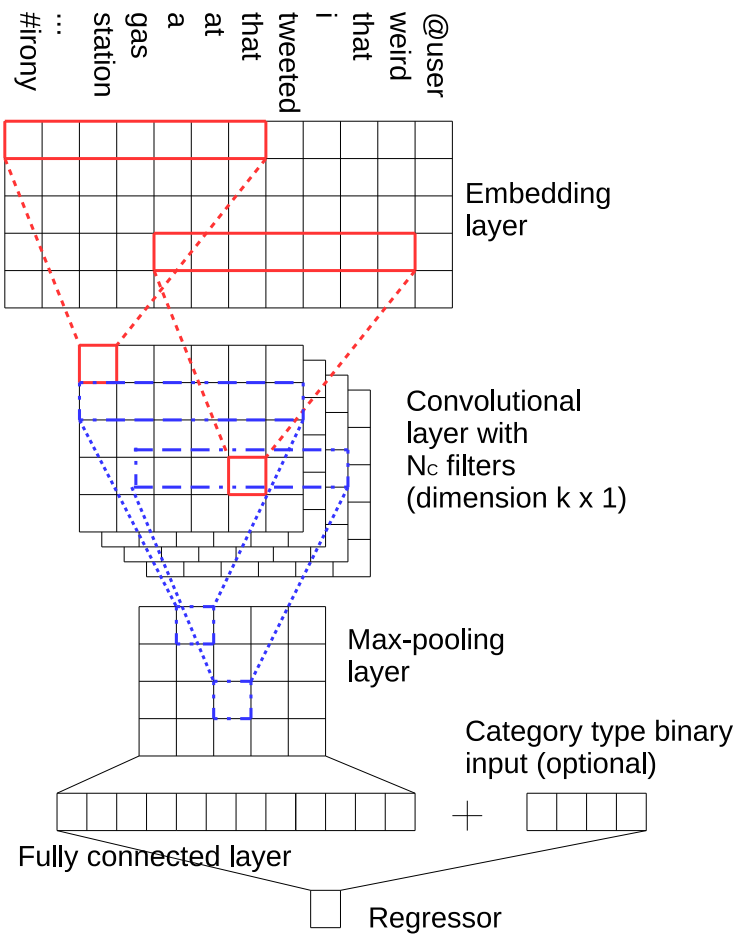


Figure 11.1: Neural network architecture.

We incorporate the figurative language indicators into the sentiment analysis process and compare the results with and without the additional information about them. We use additional dataset to examine the results achieved with extra training data and to compensate for the missing *other* category in the training data.

First we use the preprocessed dataset. Then we remove the category types harvesting criteria (e.g. the #irony hashtag) from the entire dataset. Finally we add binary features indicating category types to the second experiment.

Table 11.5 shows the regression results, where system description "`-nohash`" indicates removing the category types and "`-nohash + #`" signifies the same

Train	System	Sarcasm		Irony		Metaphor		Other		Overall	
Data	Description	Cosine	MSE	Cosine	MSE	Cosine	MSE	Cosine	MSE	Cosine	MSE
T11	Best	**0.904**	**0.934**	**0.918**	**0.673**	**0.655**	**3.155**	**0.612**	**3.411**	**0.758**	**2.117**
T11	CLaC	0.892	1.023	0.904	0.779	**0.655**	**3.155**	0.584	**3.411**	**0.758**	**2.117**
T11	CPH	0.897	0.971	0.886	0.774	0.325	5.014	0.218	5.429	0.625	3.078
T11	PRHLT	0.891	1.028	0.901	0.784	0.167	5.446	0.218	4.888	0.623	3.023
T11	CNN	**0.908**	**0.893**	0.863	1.049	0.402	4.641	0.361	4.408	0.652	2.846
T11	-nohash	0.901	0.942	**0.886**	**0.897**	0.420	4.554	0.236	5.822	0.606	3.254
T11	-nohash + #	0.899	0.995	0.879	0.928	0.277	5.134	0.291	4.772	0.620	3.073
T10+T11	CNN	0.900	0.957	0.880	0.924	**0.620**	**3.401**	**0.633**	**2.966**	**0.755**	**2.116**
T10+T11	-nohash	0.851	1.523	0.860	1.163	0.547	3.876	0.518	3.786	0.691	2.679
T10+T11	-nohash + #	0.880	1.269	0.876	0.976	0.573	3.759	0.591	3.219	0.724	2.370
T11	M5P	0.908	0.888	**0.903**	**0.802**	0.291	5.040	0.277	4.588	0.636	2.941
T11	-nohash	0.910	0.874	0.876	0.962	0.378	4.921	0.190	4.917	0.625	3.045
T11	-nohash + #	0.909	0.893	0.891	0.845	0.357	4.825	0.274	4.599	0.640	2.907
T10+T11	M5P	0.834	1.720	0.863	1.140	**0.525**	**3.986**	**0.410**	**4.121**	**0.658**	**2.858**
T10+T11	-nohash	0.816	1.678	0.832	1.295	0.468	4.341	0.388	4.469	0.623	3.063
T10+T11	-nohash + #	**0.912**	**0.858**	0.877	0.958	0.397	4.639	0.381	4.549	0.654	2.862

Table 11.5: Results on the SemEval-2015 Task 11. Training data T11 and T10 denote the respective tasks' datasets used for training. System description "`-nohash`" indicates removing the category types harvesting criteria (e.g. the #irony hashtag), "`-nohash + #`" signifies the same plus binary features indicating category types.

plus binary features indicating category types.[5]

Removing the category type indicators deteriorates the results for most cases, except for the category *Metaphor* without additional training data, where the results are actually better. We believe this is due to the removal of words that results in less uncertainty for the model. A similar case is the CNN model for *Irony* without additional training data.

Restoration of the category types using binary features again improves the results in most cases, with the exception of the category *Metaphor*. This suggests that figurative language does matter and information about the given figurative language helps improve sentiment analysis.

Metaphor seems to be very hard to correctly assign sentiment polarity.

[5]Note that the category results are not directly comparable to the SemEval-2015 task 11 results.

We believe this is caused by the datasets' composition, because the training dataset does not contain the category *Other*, thus the tweets that do not belong into the *Irony* or *Sarcasm* categories must belong to the *Metaphor* category. This claim presumes that the *Other* category is not present in the training dataset. We believe this is the reason why the *Metaphor* category is suffering in the "`-nohash + #`" setting. Moreover, tweets from training data in this category such as *"@USER we're the proverbial frog getting slowly boiled in the pot of water."* may not contain words that can be removed as figurative language indicators.

Additional training data directly improves results for *Metaphor* and *Other*, however the results for *Sarcasm* and *Irony* are worse. This effect is diminished in the "`-nohash + #`" setting. The results for the "`-nohash`" setting are consistently worse for all category types.

The best results are achieved with additional training data and basic setting with best results for the category types *Metaphor* and *Other*, which confirms the claim by Ghosh et al. [2015] i.e. there is a strong correlation between the overall performance and performance on the category *Metaphor* and *Other*.

Regardless of the categories, the *Overall* column in Table 11.5 is directly comparable to the SemEval-2015 Task 11 results. We can see that removing the figurative language indicators always deteriorates the results and their restoration by the binary figurative language features again improves the results for all cases. This supports our hypothesis that figurative language affects sentiment analysis.

11.4.3 Classification

The classification experiment in Table 11.6 was performed using the Maximum Entropy classifier from Brainy [Konkol, 2014]. This experiment shows that even small in-domain (sarcasm) training data can help improve results. Used features are unigrams and bigrams with more than five occurrences. We train the Maximum Entropy classifier on the SemEval-2015 Task 10B training data (Twitter2013-train cleansed) and test on Twitter2015-test data and the Twitter2014-sarcasm data.

The F1 score for test data changes just slightly with additional training data (tweets containing sarcasm from SemEval-2015 Task 11 trial data[6]).

[6]We mark tweets as positive for polarity ≥ 1 and negative for polarity ≤ -1.

The additional training data cause slight improvement on the test data and greatly improve the results on the sarcasm dataset. Our simple solution is competitive on the sarcasm dataset with the best results achieved with lexicons, classifier ensembles, and various dictionaries.

Description	Test F1	Sarcasm F1
Best Result	0.648	0.591
CLaC	0.620	0.514
MaxEnt	0.527	0.457
MaxEnt + trial	0.533	0.547

Table 11.6: Results on the SemEval-2015 Task 10B.

11.5 Conclusion

We have shown that figurative language can affect sentiment analysis. In our regression experiments removing the figurative language indicators deteriorates the results and their restoration by the figurative language features again improves the results on the whole dataset. The classification experiment shows that even small in-domain (sarcasm) training data can help improve results.

Our approach is simple without fine-tuned features and lexicons. We only use extra training data, which was allowed for this task. We evaluate on the SemEval-2015 Task 11 data and outperform the first team with our CNN model and additional training data in terms of MSE and we follow closely behind the first place in terms of cosine similarity. Our CNN model without additional training data would have achieved the fourth place in terms of MSE and the seventh place in terms of cosine similarity.

12 Contributions Summary

We present our contributions to the sentiment analysis task.

Automatic sentiment analysis in the Czech environment has not been thoroughly targeted by the research community. Therefore it was necessary to create publicly available labeled datasets as well as to evaluate the current state-of-the-art methods.

Contributions

- We created a large-scale labeled corpora (10k Facebook posts, 90k movie reviews, and 130k product reviews) [Habernal et al., 2013].
- We have done an in-depth research on machine learning methods and preprocessing for sentiment analysis of Czech social media [Habernal et al., 2013, 2014].
- We evaluated and compared feature selection algorithms and we investigated the influence of named entity recognition on sentiment analysis [Habernal et al., 2014].
- We created two new Czech corpora within the restaurant domain for the aspect-based sentiment analysis task: one labeled for supervised training (2.15k sentences), and the other unlabeled for unsupervised training (514k sentences) [Hercig et al., 2016b].
- We achieved state-of-the-art results in Czech aspect-based sentiment analysis with word clusters created using semantic models [Hercig et al., 2016b].
- We achieved state-of-the-art results in the aspect-based sentiment analysis task of SemEval 2016 in nine experiments among the constrained systems and in two experiments among the unconstrained systems [Hercig et al., 2016a].

- We were the first to use neural networks for sentiment analysis in Czech [Lenc and Hercig, 2016].
- We have done the first automatic detection of sarcasm in Czech and outperformed state-of-the-art methods in English [Ptáček et al., 2014].
- We created Czech sarcasm corpus consisting of 7k manually-labeled tweets and a large English corpus consisting of 780k automatically-labeled tweets [Ptáček et al., 2014].
- We confirmed that figurative language affects sentiment analysis and that the use of figurative language indicators improves results [Hercig and Lenc, 2017].
- We achieved state-of-the-art results on the Task 11 SemEval-2015 [Hercig and Lenc, 2017].

Additional research

- We explored the word order freedom of languages [Kubon et al., 2016].
- We explored stance detection [Hercig et al., 2017] in Czech.
- We explored flame detection [Steinberger et al., 2017] in Czech.
- We proposed new evaluation measure for word embeddings in [Konkol et al., 2017].

12.1 Fulfilment of the Thesis Goals

In the following paragraphs, we summarize our contributions according to the thesis goals.

- **Deal with specific properties of Czech language in the sentiment analysis environment.**

  We explored different preprocessing techniques including social-media specific tokenization, POS tagging, stemming, lemmatization, stopwords removal, lower-casing, and phonetic transcription in [Habernal et al., 2013] and [Habernal et al., 2014]. The most promising preprocessing pipeline includes stemming and either lower-casing or phonetic transcription.

We used similar preprocessing pipelines to the previous approach in [Ptáček et al., 2014] only with less variations. Phonetic transcription proved most effective.

The remaining publications dealing with Czech language [Hercig et al., 2016b, Lenc and Hercig, 2016, Hercig et al., 2017, Steinberger et al., 2017] simply used one preprocessing pipeline. We usually employed lower-casing and depending on preliminary experiments, we also removed diacritics.

Semantic features in [Hercig et al., 2016b] also improved the results by e.g. clustering different word forms with similar meaning into one cluster, thus reducing the data sparsity problem in Czech.

- **Use additional semantic and/or syntactic information to improve sentiment analysis.**

  We used syntactic features in [Habernal et al., 2013, 2014]. We significantly outperformed the baseline and achieved F-measure 0.69 using a combination of features including POS tags.

  Semantic features helped to improve results in [Hercig et al., 2016b]. We used word clusters from semantic models. Especially CBOW and GloVe models proved to be very useful.

  We achieved state-of-the art results in [Hercig et al., 2016a] with various combinations of both semantic and syntactic features (including POS tags, word dependencies from parse tree, and sentence structure).

- **Explore the influence of figurative language (e.g. sarcasm) on sentiment analysis.**

  We first explored sarcasm detection in both Czech and English in [Ptáček et al., 2014]. We created Czech sarcasm corpus consisting of 7k manually-labeled tweets and a large English corpus consisting of 780k automatically-labeled tweets. We set the state of the art in Czech and outperformed state-of-the-art methods in English.

  Later in [Hercig and Lenc, 2017], we explored the influence of figurative language including sarcasm in English. Results showed that removing the figurative language indicators deteriorates the results and their restoration by the figurative language features again improved the results on the whole dataset. Our classification experiment showed that even small in-domain (sarcasm) training data can help improve results.

12.2 Future Work

Sentiment analysis is a wide area with great potential and many research directions. One direction is stance detection, which is somewhat similar to sentiment analysis. We want to supplement stance detection dataset with sentiment annotation and explore the similarities of these tasks.

Given that there are vast amounts of data not only on social networks, but on the whole world wide web, the outputs of standard approaches need to be summarized for human readers. We want to study stance and sentiment summarization which should aim at identifying the most important utterances.

Another interesting research direction, that could tell us more about the motivation for certain opinions and stances, is the field of argumentation and reasoning.

A Author's Publications

A.1 Journal Publications

1. Habernal, I., Ptáček, T., and Steinberger, J. (2014). Supervised sentiment analysis in Czech social media. *Information Processing & Management*, 50(5):693–707

2. Hercig, T., Brychcín, T., Svoboda, L., Konkol, M., and Steinberger, J. (2016b). Unsupervised methods to improve aspect-based sentiment analysis in Czech. *Computación y Sistemas*, 20(3):365–375

A.2 Conference Publications

1. Habernal, I., Ptáček, T., and Steinberger, J. (2013). 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, Atlanta, Georgia. Association for Computational Linguistics

2. Ptáček, T., Habernal, I., and Hong, J. (2014). Sarcasm Detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland. Dublin City University and Association for Computational Linguistics

3. Hercig, T., Brychcín, T., Svoboda, L., and Konkol, M. (2016a). 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. Association for Computational Linguistics

4. Lenc, L. and Hercig, T. (2016). Neural networks for sentiment analysis in czech. In Brejová, B., editor, *Proceedings of the 16th ITAT:*

Slovenskočeský NLP workshop (SloNLP 2016), volume 1649 of *CEUR Workshop Proceedings*, pages 48–55, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform

5. Kubon, V., Lopatková, M., and Hercig, T. (2016). Searching for a measure of word order freedom. In Brejová, B., editor, *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 11–17, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform

6. Hercig, T. and Lenc, L. (2017). The impact of figurative language on sentiment analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2017*, Varna, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA

7. Hercig, T., Krejzl, P., Hourová, B., Steinberger, J., and Lenc, L. (2017). Detecting stance in czech news commentaries. In Hlaváčová, J., editor, *Proceedings of the 17th ITAT: Slovenskočeský NLP workshop (SloNLP 2017)*, volume 1885 of *CEUR Workshop Proceedings*, pages 176–180, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform

8. Steinberger, J., Brychcín, T., Hercig, T., and Krejzl, P. (2017). Cross-lingual flames detection in news discussions. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2017*, Varna, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA

9. Konkol, M., Brychcín, T., Nykl, M., and Hercig, T. (2017). Geographical evaluation of word embeddings. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, Taipei, Taiwan. Asian Federation of Natural Language Processing. In press

Bibliography

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ahkter, J. K. and Soria, S. (2010). Sentiment analysis: Facebook status messages. Technical report, Stanford University. Final Project CS224N.

Álvarez López, T., Juncal-Martínez, J., Fernández-Gavilanes, M., Costa-Montenegro, E., and González-Castaño, F. J. (2016). GTI at SemEval-2016 Task 5: SVM and CRF for Aspect Detection and Unsupervised Aspect-Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 306–311, San Diego, California. Association for Computational Linguistics.

Astudillo, R., Amir, S., Ling, W., Martins, B., Silva, M. J., and Trancoso, I. (2015). INESC-ID: Sentiment Analysis without Hand-Coded Features or Linguistic Resources using Embedding Subspaces. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 652–656, Denver, Colorado. Association for Computational Linguistics.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Balahur, A. and Tanev, H. (2012). Detecting entity-related events and sentiments from tweets using multilingual resources. In *Proceedings of the 2012 Conference and Labs of the Evaluation Forum Information Access Evaluation meets Multilinguality, Multimodality, and Visual Analytics*.

Balahur, A. and Turchi, M. (2012). Multilingual sentiment analysis using machine translation? In *Proceedings of the Association for Computational Linguistics: 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, ACL-WASSA'12, pages 52–60.

Banea, C., Mihalcea, R., and Wiebe, J. (2010). Multilingual subjectivity: Are more languages better? In *Proceedings of COLING*.

Basile, V. and Nissim, M. (2013). Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX.

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics.

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

Blamey, B., Crick, T., and Oatley, G. (2012). R U : -) or : -(? character- vs. word-gram feature selection for sentiment classification of OSN corpora. In *Proceedings of AI-2012, The Thirty-second SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 207–212. Springer.

Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Blitzer, J., Dredze, M., Pereira, F., et al. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447.

Boiy, E. and Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

Bosco, C., Patti, V., and Bolioli, A. (2013). Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63.

Bottou, L. (1998). On-line learning in neural networks. chapter On-line Learning and Stochastic Approximations, pages 9–42. Cambridge University Press, New York, NY, USA.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 804–812, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brun, C., Perez, J., and Roux, C. (2016). XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281, San Diego, California. Association for Computational Linguistics.

Brun, C., Popa, D. N., and Roux, C. (2014). XRCE: Hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 838–842, Dublin, Ireland. Association for Computational Linguistics.

Brychcín, T. and Habernal, I. (2013). Unsupervised improving of sentiment analysis using global target context. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 122–128, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Brychcín, T., Konkol, M., and Steinberger, J. (2014). UWB: Machine Learning Approach to Aspect-Based Sentiment Analysis. In *Proceedings of*

the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 817–822. Association for Computational Linguistics.

Brychcín, T. and Konopík, M. (2014). Semantic spaces for improving language modeling. *Computer Speech & Language*, 28(1):192 – 209.

Brychcín, T. and Konopík, M. (2015). Hps: High precision stemmer. *Information Processing & Management*, 51(1):68 – 91.

Brychcín, T. and Král, P. (2014). Novel unsupervised features for Czech multi-label document classification. In *13th Mexican International Conference on Artificial Intelligence (MICAI 2014)*, Berlin. Springer-Verlag. accepted.

Castellucci, G., Filice, S., Croce, D., and Basili, R. (2014). UNITOR: Aspect based sentiment analysis with structured learning. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 761–767, Dublin, Ireland. Association for Computational Linguistics.

Celikyilmaz, A., Hakkani-Tür, D., and Feng, J. (2010). Probabilistic model-based sentiment analysis of twitter messages. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pages 79–84. IEEE.

Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M., and Gandini, G. (2007). Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. *Language resources and linguistic theory: Typology, second language acquisition, English linguistics*, pages 200–210.

Charles, W. G. (2000). Contextual correlates of meaning. *Applied Psycholinguistics*, 21(04):505–524.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Chernyshevich, M. (2014). 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, Dublin, Ireland. Association for Computational Linguistics.

Choi, Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 793–801. Association for Computational Linguistics.

Choi, Y. and Cardie, C. (2009). Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 590–598. Association for Computational Linguistics.

Chollet, F. (2015). Keras. <https://github.com/fchollet/keras>.

Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.

Crammer, K. and Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292.

Cruz, F. L., Troyano, J. A., Pontes, B., and Ortega, F. J. (2014). Building layered, multilingual sentiment lexicons at synset and lemma levels. *Expert Systems with Applications*, 41(13):5984–5994.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining*.

Dolamic, L. and Savoy, J. (2009). Indexing and stemming approaches for the czech language. *Information Processing & Management*, 45(6):714–720.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014,*

the 25th International Conference on Computational Linguistics: Technical Papers, pages 69–78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Fahrni, A. and Klenner, M. (2008). Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proc. of the Symposium on Affective Language in Human and Machine, AISB*, pages 60–63.

Firth, J. R. (1957). A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305.

Forman, G. and Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explor. Newsl.*, 12(1):49–57.

Ghiassi, M., Skinner, J., and Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, 40(16):6266–6282.

Ghorbel, H. and Jacot, D. (2011). Sentiment analysis of french movie reviews. In Pallotta, V., Soro, A., and Vargiu, E., editors, *Advances in Distributed Agent-Based Retrieval Tools*, volume 361 of *Studies in Computational Intelligence*, pages 97–108. Springer Berlin Heidelberg.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., and Reyes, A. (2015). SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478, Denver, Colorado. Association for Computational Linguistics.

Ghosh, A. and Veale, D. T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume*

2, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford.

González-Ibáñez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 581–586, Stroudsburg, PA, USA. Association for Computational Linguistics.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.

Graupe, D. (2013). *Principles of artificial neural networks*, volume 7. World Scientific.

Gupta, P. and Gómez, J. A. (2015). PRHLT: Combination of Deep Autoencoders with Classification and Regression Techniques for SemEval-2015 Task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 689–693, Denver, Colorado. Association for Computational Linguistics.

Guthrie, D., Allison, B., Liu, W., Guthrie, L., and Wilks, Y. (2006). A closer look at skip-gram modelling. In *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pages 1–4.

Habernal, I. and Brychcín, T. (2013). Semantic spaces for sentiment analysis. In Habernal, I. and Matoušek, V., editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 484–491. Springer Berlin Heidelberg.

Habernal, I., Ptáček, T., and Steinberger, J. (2014). Supervised sentiment analysis in Czech social media. *Information Processing & Management*, 50(5):693–707.

Habernal, I., Ptáček, T., and Steinberger, J. (2013). 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, Atlanta, Georgia. Association for Computational Linguistics.

Hajic, J., Panevová, J., Hajicová, E., Sgall, P., Pajas, P., Štepánek, J., Havelka, J., Mikulová, M., Zabokrtskỳ, Z., and Ševcıková-Razımová, M. (2006). Prague dependency treebank 2.0. *CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia*, 98.

Hajmohammadi, M. S., Ibrahim, R., and Othman, Z. A. (2012). Opinion mining and sentiment analysis: A survey. *International Journal of Computers & Technology*, 2(3).

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

Hamdan, H., Bellot, P., and Bechet, F. (2015). 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, Denver, Colorado. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

He, B., Macdonald, C., He, J., and Ounis, I. (2008). An effective statistical approach to blog post opinion retrieval. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1063–1072. ACM.

Hercig, T. (2015). Aspects of sentiment analysis. Technical Report DCSE/TR-2015-04, University of West Bohemia, Pilsen, Czech Republic.

Hercig, T., Brychcín, T., Svoboda, L., and Konkol, M. (2016a). 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. Association for Computational Linguistics.

Hercig, T., Brychcín, T., Svoboda, L., Konkol, M., and Steinberger, J. (2016b). Unsupervised methods to improve aspect-based sentiment analysis in Czech. *Computación y Sistemas*, 20(3):365–375.

Hercig, T., Krejzl, P., Hourová, B., Steinberger, J., and Lenc, L. (2017). Detecting stance in czech news commentaries. In Hlaváčová, J., editor, *Proceedings of the 17th ITAT: Slovenskočeský NLP workshop (SloNLP 2017)*, volume 1885 of *CEUR Workshop Proceedings*, pages 176–180, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform.

Hercig, T. and Lenc, L. (2017). The impact of figurative language on sentiment analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2017*, Varna, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Jiang, M., Zhang, Z., and Lan, M. (2016). ECNU at SemEval-2016 Task 5: Extracting Effective Features from Relevant Fragments in Sentence for Aspect-Based Sentiment Analysis in Reviews. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 361–366, San Diego, California. Association for Computational Linguistics.

Jo, Y. and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA. ACM.

Jurgens, D. and Stevens, K. (2010). The s-space package: An open source package for word space models. *In System Papers of the Association of Computational Linguistics*.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kamps, J., Mokken, R. J., Marx, M., and de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, volume IV, pages 1115–1118. European Language Resources Association, Paris.

Kanis, J. and Skorkovská, L. (2010). Comparison of different lemmatization approaches through the means of information retrieval performance. In Sojka, P., Horák, A., Kopeček, I., and Pala, K., editors, *Text, Speech and*

Dialogue, volume 6231 of *Lecture Notes in Computer Science*, pages 93–100. Springer Berlin Heidelberg.

Karypis, G. (2003). Cluto — a clustering toolkit.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiritchenko, S., Zhu, X., Cherry, C., and Mohammad, S. (2014a). NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland. Association for Computational Linguistics.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014b). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Konkol, M. (2014). Brainy: A machine learning library. In Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L., and Zurada, J., editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer International Publishing.

Konkol, M., Brychcín, T., and Konopík, M. (2015). Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470–3479.

Konkol, M., Brychcín, T., Nykl, M., and Hercig, T. (2017). Geographical evaluation of word embeddings. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, Taipei, Taiwan. Asian Federation of Natural Language Processing. In press.

Konkol, M. and Konopík, M. (2013). Crf-based czech named entity recognizer and consolidation of czech ner research. In Habernal, I. and Matoušek, V., editors, *Text, Speech and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 153–160. Springer Berlin Heidelberg.

Kouloumpis, E., Wilson, T., and Moore, J. (2011). Twitter sentiment analysis: The good the bad and the OMG! In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*. The AAAI Press.

Kubon, V., Lopatková, M., and Hercig, T. (2016). Searching for a measure of word order freedom. In Brejová, B., editor, *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 11–17, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform.

Kumar, A., Kohail, S., Kumar, A., Ekbal, A., and Biemann, C. (2016). IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1129–1135, San Diego, California. Association for Computational Linguistics.

Laboreiro, G., Sarmento, L., Teixeira, J., and Oliveira, E. (2010). Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, pages 81–88, New York, NY, USA. ACM.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on*

Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pages 1188–1196.

LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, pages 143–155.

Lenc, L. and Hercig, T. (2016). Neural networks for sentiment analysis in czech. In Brejová, B., editor, *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 48–55, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform.

Lenc, L. and Král, P. (2016). Deep neural networks for Czech multi-label document classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, Konya, Turkey.

Liebrecht, C., Kunneman, F., and Van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia. Association for Computational Linguistics.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

Liu, B. (2012). Sentiment analysis and opinion mining. *Morgan & Claypool Publishers*.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.

Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer.

Long, C., Zhang, J., and Zhu, X. (2010). A review selection approach for accurate feature rating estimation. In *Proceedings of Coling 2010: Poster Volume*.

López, R., Tejada, J., and Thelwall, M. (2012). Spanish sentistrength as a tool for opinion mining peruvian facebook and twitter. In *Artificial Intelligence Driven Solutions to Business and Engineering Problems*, pages 82–85. ITHEA, Sofia, Bulgaria.

Lukin, S. and Walker, M. (2013). Really? Well. Apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 30–40, Atlanta, Georgia. Association for Computational Linguistics.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods Instruments and Computers*, 28(2):203–208.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Martineau, J. and Finin, T. (2009). Delta TFIDF: An improved feature space for sentiment analysis. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA*. The AAAI Press.

Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., and Montejo-Raéz, A. (2014). Sentiment analysis in twitter. *Natural Language Engineering*, 20(01):1–28.

Maynard, D. and Greenwood, M. (2014). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit.

McGillion, S., Martínez Alonso, H., and Plank, B. (2015). CPH: Sentiment analysis of Figurative Language on Twitter #easypeasy #not. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 699–703, Denver, Colorado. Association for Computational Linguistics.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of International Conference on World Wide Web*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Miller, G. and Fellbaum, C. (1998). Wordnet: An electronic lexical database.

Moghaddam, S. and Ester, M. (2010). Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceeding of the ACM conference on Information and knowledge management*.

Montejo-Ráez, A., Martínez-Cámara, E., Martín-Valdivia, M. T., and Ureña López, L. A. (2012). Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 3–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Making Sense of Microposts (#MSM2011)*, pages 93–98.

Nocedal, J. (1980). Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational

text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Özdemir, C. and Bergler, S. (2015a). A Comparative Study of Different Sentiment Lexica for Sentiment Analysis of Tweets. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 488–496, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Özdemir, C. and Bergler, S. (2015b). CLaC-SentiPipe: SemEval2015 Subtasks 10 B,E, and Task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 479–485, Denver, Colorado. Association for Computational Linguistics.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*. European Language Resources Association.

Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1386–1395, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Patočka, M. (2013). Machine learning for sentiment analysis. Master's thesis, University of West Bohemia, Plzen, Czech Republic. [In Czech].

Pavlopoulos, J. and Androutsopoulos, I. (2014). *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, chapter Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method, pages 44–52. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Platt, J. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical report, Microsoft Research.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O. D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). SemEval-2014 Task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Poria, S., Cambria, E., Hazarika, D., and Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1601–1612, Osaka, Japan. The COLING 2016 Organizing Committee.

Ptáček, T., Habernal, I., and Hong, J. (2014). Sarcasm Detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages

213–223, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Pustejovsky, J. and Stubbs, A. (2013). *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc., Sebastopol, CA 95472.

Rabiner, L. (2010). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286.

Rao, D. and Ravichandran, D. (2009). Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–682. Association for Computational Linguistics.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.

Reyes, A., Rosso, P., and Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data Knowl. Eng.*, 74:1–12.

Reyes, A., Rosso, P., and Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Rohde, D. L. T., Gonnerman, L. M., and Plaut, D. C. (2004). An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Psychology*, 7:573–605.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., and Stoyanov, V. (2015). SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado. Association for Computational Linguistics.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Saias, J. (2015). Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 767–771, Denver, Colorado. Association for Computational Linguistics.

Saif, H., He, Y., Fernandez, M., and Alani, H. (2014). Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 54–63. Springer.

San Vicente, I. n., Saralegi, X., and Agerri, R. (2015). EliXa: A Modular and Flexible ABSA Platform. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 748–752, Denver, Colorado. Association for Computational Linguistics.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

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

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642.

Steinberger, J., Brychcín, T., Hercig, T., and Krejzl, P. (2017). Cross-lingual flames detection in news discussions. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2017*, Varna, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Steinberger, J., Brychcín, T., and Konkol, M. (2014). Aspect-level sentiment analysis in czech. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Baltimore, USA. Association for Computational Linguistics.

Steinberger, J., Ebrahim, M., Ehrmann, M., Hurriyetoglu, A., Kabadjov, M. A., Lenkova, P., Steinberger, R., Tanev, H., Vázquez, S., and Zavarella, V. (2012). Creating sentiment dictionaries via triangulation. *Decision Support Systems*, 53:689—694.

Steinberger, J., Lenkova, P., Kabadjov, M. A., Steinberger, R., and der Goot, E. V. (2011). Multilingual entity-centered sentiment analysis evaluated by parallel corpora. In *Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing*, RANLP'11, pages 770–775.

Stepanov, E. and Riccardi, G. (2011). Detecting general opinions from customer surveys. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 115–122.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Paris, France. European Language Resources Association (ELRA).

Sulis, E., Farías, D. I. H., Rosso, P., Patti, V., and Ruffo, G. (2016). Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not . *Knowledge-Based Systems*, 108:132 – 143. New Avenues in Knowledge Bases for Natural Language Processing.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Tamchyna, A., Fiala, O., and Veselovská, K. (2015). Czech aspect-based sentiment analysis: A new dataset and preliminary results. *Proceedings of the 15th conference ITAT 2015*, pages 95–99.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of International Conference on World Wide Web*.

Toh, Z. and Su, J. (2015). 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. Association for Computational Linguistics.

Toh, Z. and Su, J. (2016). 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, San Diego, California. Association for Computational Linguistics.

Toh, Z. and Wang, W. (2014). DLIREC: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 235–240, Dublin, Ireland. Association for Computational Linguistics.

Tsur, O., Davidov, D., and Rappoport, A. (2010). ICWSM - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In Cohen, W. W. and Gosling, S., editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press.

Tsytsarau, M. and Palpanas, T. (2012). Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514.

Turney, P. and Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346. cited By 559.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Uchyigit, G. (2012). Experimental evaluation of feature selection methods for text classification. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 1294–1298.

Vanin, A. A., Freitas, L. A., Vieira, R., and Bochernitsan, M. (2013). Some clues on irony detection in tweets. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 635–636. International World Wide Web Conferences Steering Committee.

Veselovská, K. (2013). Czech subjectivity lexicon: A lexical resource for czech polarity classification. In Gajdošová, K. and Žáková, A., editors, *Proceedings of the Seventh International Conference Slovko 2013; Natural Language Processing, Corpus Linguistics, E-learning*, pages 279–284, Lüdenscheid, Germany. Slovak National Corpus, L'. Štúr Institute of Linguistics, Slovak Academy of Sciences, RAM-Verlag.

Veselovská, K. (2015). *On the Linguistic Structure of Emotional Meaning in Czech*. PhD thesis, Faculty of Mathematics and Physics, Charles University in Prague, Prague, Czech Republic.

Veselovská, K. and Hajič jr., J. (2013). Why words alone are not enough: Error analysis of lexicon-based polarity classifier for czech. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 1–5, Nagoya, Japan. Asian Federation of Natural Language Processing, Asian Federation of Natural Language Processing.

Veselovská, K., Hajič jr., J., and Šindlerová, J. (2012). Creating annotated resources for polarity classification in Czech. In Jancsary, J., editor, *Proceedings of KONVENS 2012*, pages 296–304. ÖGAI. PATHOS 2012 workshop.

Veselovská, K., Hajič jr., J., and Šindlerová, J. (2014). Subjectivity lexicon for czech: Implementation and improvements. *Journal for Language Technology and Computational Linguistics*, 29(1):47–61.

Vilares, D., Alonso, M. A., and Gómez-Rodríguez, C. (2015). A syntactic approach for opinion mining on spanish reviews. *Natural Language Engineering*, 21:139–163.

Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., and Tounsi, L. (2014). DCU: Aspect-based polarity classification for SemEval Task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229, Dublin, Ireland. Association for Computational Linguistics.

Wang, Y. and Witten, I. H. (1997). Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer.

Wei, C.-P., Chen, Y.-M., Yang, C.-S., and Yang, C. (2010). Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. *Information Systems and e-Business Management*, 8(2):149–167.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference*

on human language technology and empirical methods in natural language processing, pages 347–354. Association for Computational Linguistics.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xianghua, F., Guo, L., Yanyan, G., and Zhiqiang, W. (2013). Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowledge-Based Systems*, 37(0):186 – 195.

Xu, T., Peng, Q., and Cheng, Y. (2012). Identifying the semantic orientation of terms using s-hal for sentiment analysis. *Knowledge-Based Systems*, 35:279–289.

Yu, L.-C., Wu, J.-L., Chang, P.-C., and Chu, H.-S. (2013). Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. *Knowledge-Based Systems*, 41:89–97.

Zhang, D., Si, L., and Rego, V. J. (2012). Sentiment detection with auxiliary data. *Information Retrieval*, 15(3-4):373–390.

Zhang, K., Cheng, Y., Xie, Y., Honbo, D., Agrawal, A., Palsetia, D., Lee, K., keng Liao, W., and Choudhary, A. N. (2011). SES: Sentiment elicitation system for social media data. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th Conference on, Vancouver, BC, Canada, December 11, 2011*, pages 129–136. IEEE.

Zhang, M., Zhang, Y., and Fu, G. (2016). Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhang, X. and LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Zhang, Z. and Lan, M. (2015). ECNU: Extracting Effective Features from Multiple Sequential Sentences for Target-dependent Sentiment Analysis in Reviews. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 736–741, Denver, Colorado. Association for Computational Linguistics.

Zheng, X., Lin, Z., Wang, X., Lin, K.-J., and Song, M. (2014). Incorporating appraisal expression patterns into topic modeling for aspect and sentiment word identification. *Knowledge-Based Systems*, 61(0):29 – 47.

Zheng, Z., Wu, X., and Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.*, 6(1):80–89.

Zhou, S., Chen, Q., and Wang, X. (2010). Active deep networks for semi-supervised sentiment classification. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1515–1523. Association for Computational Linguistics.

Zhu, X., Kiritchenko, S., and Mohammad, S. (2014). NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447, Dublin, Ireland. Association for Computational Linguistics.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.