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

Distributional Semantics Using Neural Networks

Ing. Lukáš Svoboda

Doctoral Thesis

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

Supervisor: prof. Ing. Václav Matoušek, CSc.
Consulting Specialist: Ing. Tomáš Brychcín, Ph.D.

Pilsen, 2019

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

Distribuční sémantika s využitím neuronových sítí

Ing. Lukáš Svoboda

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

Školitel: prof. Ing. Václav Matoušek, CSc.
Konzultant-specialista: Ing. Tomáš Brychcín, Ph.D.

Plzeň, 2019

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 August 1, 2019

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 1. srpna 2019

Ing. Lukáš Svoboda

Acknowledgment

Firstly, I would like to thank my supervisor prof. Ing. Václav Matoušek, CSc. and consulting specialist Ing. Tomáš Brychcín, Ph.D. for their guidance and support.

Secondly, I need to thank my university colleagues for a friendly atmospehere and valuable advice. Especially, I would like to thank my wife Ing. Alena Svobodová for her love, support and patience not only during writing my thesis.

Last but not the least, I would like to express my deepest gratitude to my mother who unfortunately is no longer with us. She gave me strenght, determination, sense of purpose, her wisdom and not only those skills which helped me to finish this thesis. I wish she could have read this thesis.

Abstract

During recent years, neural network-based methods are showing crucial improvements in catching semantic and syntactical properties of words or sentences. Much has been investigated about word embeddings of English words and phrases, but little attention has been dedicated to other languages.

At the level of words, we explore the behavior of state-of-the-art word embedding methods on Czech and Croatian, which are representatives of Slavic languages characterized by rich word morphology. We build the first corpora for testing word embedding accuracy on similarity and analogy tasks of Czech and Croatian language.

For understanding semantics on the sentence level, we show how to deal with these languages on some of the currently most discused tasks such as aspect-based sentiment analysis (ABSA) and semantic textual similarity (STS). Most of the community work here is also dedicated to English language. Free word order of Czech and Croatian complicates learning of current state-of-the-art methods. We build first corpora and state-of-the-art models for understanding sentence semantics adapted on highly inflectional language for dealing with STS and ABSA task.

Finally, we develop a new approach for learning word embeddings enriched with global information extracted from Wikipedia. We evaluate our new approach based on the Continuous Bag-of-Words and Skip-gram models enriched with global context information on highly inflectional language and compare it with English. The results of the model shows, that our approach can help to create word embeddings that perform better with smaller corpora and improve performance on highly inflected languages.

Our research helps the community to continue with improving the state-of-the-art methods with focus on highly inflectioned languages. The thesis also focuses on further use of neural networks (NN) in Natural Language Processing (NLP) tasks. Basic machine learning algorithms for NLP are described as well as the commonly used algorithms for extracting word embeddings. A brief overview of distributional semantics methods is presented. We emphasize the analysis of models' behaviour in the highly inflected language environment.

Abstrakt

V posledních letech vykazují metody založené na neuronových sítích zásadní zlepšení v zachycení sémantiky a syntaxe slov nebo vět. Mnoho bylo vyzkoumáno o vnoření anglických slov a frází, ale jen malá pozornost byla věnována jiným jazykům.

Na úrovni slov zkoumáme chování nejmodernějších metod pro tvorbu vnořených slov na češtině a chorvatštině, což jsou zástupci slovanských jazyků charakterizovaných bohatou morfologií slov. Tvoříme první korpusy pro testování kvality číselné reprezentace (vnoření) slov na podobnost a tzv. úlohu slovních analogií českého a chorvatského jazyka.

Pro pochopení významu vět ukážeme, jak s těmito jazyky pracovat při řešení aktuálně jedněch z nejdiskutovanějších úloh jako je sémantická textová analýza a analýza sentimentu založená na aspektech. Většina prací komunity v počítačovém zpracování přirozeného jazyka věnující se těmto úlohám se také zaměřuje výlučně na anglický jazyk. Nejen volný slovosled českého a chorvatského jazyka komplikuje učení současných nejmodernějších metod. Představíme první korpusy a modely, které dokáží pochopit sémantiku vět k řešení těchto úloh pro flektivní jazyky.

Na závěr představíme nový přístup k učení číselné reprezentace slov obohacený o globální informace získané z Wikipedie. Pro náš nový přístup vycházíme z modelů Continuous Bag-of-Words a Skip-gram vylepšených o globální kontextové informace. Provedeme analýzu chování výsledného modelu na flektivním jazyku a porovnáváme je s výsledky v angličtině. Výsledky tohoto modelu ukazují, že náš přístup může pomoci vytvořit číselné preprezentace slov, které lépe fungují s menšími korpusy a zlepšují výkonnost ve vysoce flektivních jazycích.

Náš výzkum pomáhá komunitě pokračovat ve zdokonalování nejmodernějších metod s důrazem na flektivní jazyky. Práce se také zaměřuje na využití neuronových sítí mezi úlohami v počítačovém zpracování přirozeného jazyka. Jsou popsány základní algoritmy strojového učení a jejich použití při zpracování přírozeného jazyka a nejčastěji využívané algoritmy pro extrakci číselné reprezentace slov. Je uveden stručný přehled metod distribuční sémantiky.

Contents

1 Introduction

Understanding semantics of the text is crucial in many of Natural Language Processing (NLP) tasks. Each improvement in semantic understanding of text may also improve the particular application, where the model is used. Its impact can be seen in sub-fields of NLP areas, such as sentiment analysis, machine translation, natural language understanding, named entity recognition (NER), word sense disambiguation and many others.

Research on distributional semantics has been evolving more than 20 years. Most of the techniques for modeling semantics have been outperformed by neural network based models and deep learning during recent years. We believe that distributional semantics models (DSMs) are essential to understand the meaning of text.

Semantics is the meaning of a text and if we understand the meaning, we will likely benefit in many NLP tasks. The extraction of the meaning from a text became the backbone research area in NLP. It led to impressive results on English. However, during our research we experienced significantly lower performance with most of state-of-the-art models dealing with tasks such as (aspect-based) sentiment analysis or semantic textual similarity (STS) when applied to Czech.

The fundamental question that we raised was: 'What if the problem is already in the basic extraction of word meaning?' We could not immediately answer our question, because there were no word analogy corpora to test the quality of word embeddings on Czech for example. Czech has not yet been thoroughly targeted by the research community. Czech as a representative of an inflective language is an ideal environment for the study of various aspects distributional sematics for inflectional languages. It is challenging because of its very flexible word order and many different word forms.

The lack of data is always issue in NLP, especially with small languages. There are many researchers trying to surpass the latest best results or achieve the state-of-the-art results on a variety of NLP tasks in English. The research is then usually adapted to other languages, but models usually do not perform as well as on English.

We conceive this thesis to deal with several aspects of distributional semantics. The breadth of this thesis can lead to more general view and better understanding of meaning the text. We can reveal and overcome unexpected obstacles, create necessary evaluation datasets and even come up with new creative solutions to better extract the meaning of textual data.

Therefore, the aim of this doctoral thesis is to study various aspects of distributional semantics with the emphasis on the Czech language.

1.1 Overall Aims of the PhD Thesis

The goal of this doctoral thesis is to explore models for distributional semantics using neural networks for improving performance of semantic representation with special emphasis on highly inflected languages. The work will be focused on the following research tasks:

- Study the influence of rich morphology on the quality of meaning representation.
- Propose novel approaches based on neural networks for improving the meaning representation of inflectional languages.
- Use distributional semantic models for improving NLP tasks.

1.2 Outline

The thesis is organized as follows:

The state-of-the-art architectures for Distributional Semantics are discussed in Chapter 2. Chapter 3 discuss problem of standard Machine Learning approaches dealing with NLP problems and describes neural networks architectures that currently play the key role in modeling semantics.

Semantic models based on distributional semantics can be used as additional sources of information for aspect-based sentiment analysis (ABSA), machine translation, named entity recognition, semantic textual similarity and many other tasks of NLP.

The related work and testing DSMs with highly inflected languages is presented in Chapter 4. Further, the unique and state-of-the-art model for STS task is presented in Chapter 5, the model is adapted and tested on Czech language in Section 5.3. The ABSA model and corpora with focus on Czech language are presented in Chapter 6.

In Chapter 7 we show our new approach based on the state-of-the-art distributional semantic models enriched with global context information and evaluate with highly inflected Czech language.

We make a summary and conclude in Chapter 8 and show potential further work in Section 8.4. Chapter 8.3 gives an overview of fulfilment of individual research tasks defined in this chapter.

2 Distributional Semantics

Distributional semantics is a research area that develops and studies theories and methods for quantifying and categorizing semantic similarities between linguistic items based on their distributional properties in large samples of language data. The basic idea of distributional semantics can be summed up in the so-called Distributional Hypothesis: *"linguistic items with similar distributions have similar meanings"*.

The idea that "you shall know a word by the company it keeps" was popularized by Firth Firth [1957], followed by other researchers; "words with similar meanings will occur with similar neighbors if enough text material is available" [Schutze and O. Pedersen, 1996]; "a representation that captures much of how words are used in natural context will capture much of what we mean by meaning" [Landauer and Dumais, 1997]; and "words that occur in the same contexts tend to have similar meanings" [Pantel, 2005].

The claim has theoretical bases in psychology, linguistics, and lexicography [Charles, 2000]. During last years it has become a popular. The models based upon Distributional Hypothesis are often referred to as the DSMs, see Section 2.2 for further information.

2.1 Model Types

2.1.1 Distributional Model Structure

Distributional models of words reflects the basic distributional hypothesis. The idea behind the Distributional Hypothesis is clear: there is a correlation between distributional similarity and meaning similarity. In other words: the word meaning is related to the context where it usually occurs and therefore it is possible to compare the meanings of two words by statistical comparisons of their contexts. This implication was confirmed by empirical tests carried out on human groups in [Rubenstein and Goodenough, 1965, Charles, 2000].

Distributional profile of words is based on which other words surround them. The DSMs typically represent the word meaning as a vector, where the vector reflects the contextual information of a word across the training corpus. Each word $w \in W$ (where W denotes the word vocabulary) is associated with a vector of real numbers $\boldsymbol{w} \in \mathbb{R}^k$. 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.

2.1.2 Bag-of-words Model Structure

In this model, a text (such as a sentence or a document) is represented as the bag of its words, disregarding grammar and even word order. The term *bag* means a *set* where the order has no role, however, the duplicates are allowed (the bags a, a, a, b, b, c and c, a, b, a, b, a are equivalent). Bag-of-words model is mainly used as a tool of feature extraction for NLP tasks. After transforming the text into a "bag of words", we can calculate various measures to characterize the text. The most common type of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text.

An early reference can be found in [Harris, 1954], but the first practical application was arguably in information retrieval. In work of [Salton et al., 1975], the documents were represented as bags-of-words and the frequencies of words in a document indicated the relevance of the document to a query. The implication is that two documents tend to be similar if they have similar distribution of similar words, no matter what is their order. This is supported by the intuition that the topic of a document will probabilistically influence the author's choice of words when writing the document.

Similarly, the words can be found related in meaning if they occur in similar documents (where document represents the word context). Thus, both hypotheses (Bag-of-words Hypothesis and Distributional Hypothesis) are related.

This intuition later expanded into many often used models for meaning extraction, such as latent semantic analysis (LSA) [Deerwester et al., 1990], probabilistic latent semantic analysis (PLSA) [Hofmann, 1999], latent Dirichlet allocation (LDA) [Blei et al., 2003], and others.

2.2 Distributional Semantic Models

DSMs learn contextual patterns from huge amount of textual data. They typically represent the meaning as a vector which reflects the contextual (distributional) information across the texts [Turney and Pantel, 2010]. The words $w \in W$ are associated with a vector of real numbers $\boldsymbol{w} \in \mathbb{R}^k$. Represented geometrically, the meaning is a point in a k-dimensional space. The words that are closely related in meaning tend to be closer in the space. This architecture is sometimes referred to as the *Semantic Space*. The vector representation allows us to measure similarity between the meanings, most often by the cosine of the angle between the corresponding vectors. Approaches that extract such vectors are often called Word Embedding methods.

In last years, the extraction of meaning from a text became the fundamental research area in NLP. Word-based semantic spaces provide impressive performance in a variety of NLP tasks, such as language modeling [Brychcín and Konopík, 2015], NER [Konkol et al., 2015a], sentiment analysis [Hercig et al., 2016a], and many others (see Section 3.8).

In this thesis we focus on Czech, which belongs to the West Slavic family and Croatian, from the South Slavic family language. Czech has seven cases and three genders. Croatian language has also seven cases and three genders. Many properties of both languages are very similar because of historical similarities and mutual interaction. Both languages have a relatively free word order (from the purely syntactic point of view): words in a sentence can usually be ordered in several ways which carry a slightly different meaning. These properties of Czech and Croatian language complicate the distributional semantics modeling. High number of word forms and more sequences of words that are possible in the language lead to a higher number of n-grams. Free word order, according our opinion, complicates the fundamental use of Distributional Hypothesis.

2.2.1 Context Types

Different types of context induce different kinds of semantic space models. [Riordan and Jones, 2011] and [McNamara, 2011] distinguish *context-word* and *context-region* approaches to the meaning extraction. In this thesis we use the notion *local context* and *global context*, respectively, because we think this notion describes the principle of the meaning extraction better.

Global context

The models that use the global context are usually based upon bag-of-word hypothesis, assuming that the words are semantically similar if they occur in similar documents, and that the word order has no meaning. The document can be a sentence, a paragraph, or an entire text. These models are able to register long-range dependencies among words. For example, if the document is about *hockey*, it is likely to contain words like *hockey-stick* or *skates*, and these words are found to be related in meaning.

Local context

The local context models are those that collect short contexts around the word using a moving window to model its semantics. These methods do not require text that is naturally divided into documents or pieces of text. Thanks to the short context, these models can take the word order into account, thus they usually model semantic as well as syntactic relations among words. In contrast to the global semantics models, these models are able to find mutually substitutable words in the given context. Given the sentence *The dog is an animal*, the word *dog* can be for example replaced by *cat*.

2.2.2 Model Architectures

There are several architectures that have been successfully used to extract meaning from raw text. In our opinion, the following four architectures are the most important for our work (see other architectures in [Svoboda, 2016]):

Co-occurrence Matrix

The frequencies of co-occurring words (often taken as an argument of some weighting function, e.g. *term frequency – inverse document frequency* (TF-IDF) [Ramos et al., 2003], mutual information [E. Shannon, 1948], etc.) are recorded into a matrix. The dimension of such matrix is sometimes big, and thus the *singular value decomposition* (SVD) or different algorithm can be used for dimensionality reduction.

Formally, the co-occurrence matrix of a textual corpus is a square matrix of unique words with dimensions $N \times N$. A cell m_{ij} contains the number of times word w_i co-occurs with word w_j within a specific context. Context can be either a natural unit such as a sentence or a certain window of m words (where m is an application-dependent parameter). The upper and lower triangles of the matrix are identical since co-occurrence is a symmetric relation.

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

Topic Model

The group of methods based upon the bag-of-word hypothesis that try to discover latent (hidden) topics in the text are called *topic models*. They usually represent the meaning of the text as a vector of topics but it is also possible to use them for representing the meaning of a word. The number of topics in the text is usually set in advance.

It is assumed that documents may vary in domain, topic and styles, which means that they also differ in the probability distribution of n-grams. This assumption is used for adapting language models to the long context (domain, topic, style of particular documents). LSA (or similar methods) [Choi et al., 2001] aim to partition a document into blocks, such that each segment is coherent and consecutive segments are about different topics. This long context information is added to standard n-gram models to improve their performance. A very effective group of models (sometimes called topic-based language models) work with this idea for the benefit of language modeling.

In [Bellegarda, 2000] a significant reduction in perplexity[1] (down to 33%)

[1] A measurement of how well a probability distribution or probability model predicts a sample

and WER[2] (down to 16%) in the WSJ[3] corpus was shown. Many other authors have obtained good results with PLSA [Gildea and Hofmann, 1999, Wang et al., 2003] and LDA [Tam and Schultz, 2005, 2006] approaches.

Neural Network

In the last years, these models have become very popular. It is the human brain that defines semantics, so it is natural to use a neural network for the meaning extraction. The principles of the meaning extraction differ with the architecture of a neural network. Much work on improving the learning of word representations with Neural Networks has been done, from feed-forward networks [Bengio et al., 2003] to hierarchical models [Morin and Bengio, 2005, Mnih and Hinton, 2009] and recently recurrent neural networks [Mikolov et al., 2010].

In [Mikolov et al., 2013a,c] Mikolov examined existing word embeddings and showed that these representations already captured meaningful syntactic and semantic regularities such as the singular and plural relation that vectors $\boldsymbol{orange} - \boldsymbol{oranges} = \boldsymbol{plane} - \boldsymbol{planes}$. Read more in section 3.8

2.3 Language Models

Language models are crucial in NLP, and the backbone principle of language modeling is often used in DSMs. The goal of a language model is very simple, to estimate probability of any word sequence possible in the language. Even though the task looks very easy, a satisfactory solution for natural language is very complicated.

2.4 Statistical Language Models

A statistical language model is a probability distribution over sequences of words. Given such a sequence, say of length m, it assigns a probability $P(w_1, \ldots, w_m)$ to the whole sequence. Let W denote the word vocabulary.

[2]The Word Error Rate (WER) measure is often used in Speech recognition
[3]Wall Street Journal (WSJ) [Paul and Baker, 1992]

The $W^{\mathbb{N}}$ is the set of all combination of word sequences of length N which it is possible to create from the vocabulary W. Let

$$\mathcal{L} \subseteq W^{\mathbb{N}} \tag{2.1}$$

be a set of all possible word sequences in a language.

The sequence of words (i.e. sentence) can be expressed as

$$S = w_1, \cdots, w_m, \qquad S \in \mathcal{L}. \tag{2.2}$$

The language model tries to capture the regularities of a natural language by giving constraints on sequences S. These constraints can be either *deterministic* (some sequences are possible, some not) or *probabilistic* (some sequences are more probable than others).

The reason we are talking about Language modeling is simple: the better the models represent language, the better results we usually achieve solving our NLP problem (such as semantic understanding). Currently, there is a massive research invested in language modeling, but this time invested into creating the new representation is being outperformed by simple n-gram model and recently by simple recurrent neural network models [Mikolov et al., 2010]. In Chapter 3 we show that standard n-grams and many other language models with strong mathematical background can be outperformed by Recursive Neural Network with memory.

2.5 N-gram Language Models

There is no way to process all possible histories of words with all possible lengths k. The number of training parameters needed to be estimated rises exponentially with extending the history.

Truncating the word history is done to decrease the number of training parameters. It means, that the probability of word w_i is estimated only by $n-1$ preceding words, not by the complete history.

$$P(S) = P(w_1^m) \approx \prod_{i=1}^{m} \tilde{P}\left(w_i | w_{i-n+1}^{i-1}\right). \tag{2.3}$$

These models are referred to as the *n-gram language models*. N-gram language models have been the most often used architecture for language modeling for a long time. N-grams, where $n = 1$, are called *unigrams*. The most often used are, however, *bigrams* ($n = 2$) and *trigrams* ($n = 3$).

2.6 Clustering (word classes)

The goal of clustering is simple; to find an optimal grouping in a set of unlabeled data. That is to say, similar words should share parameters which leads to generalization [Brychcín and Konopík, 2011].

Example:

$$\begin{aligned} Class_1 &= \{black, white, blue, red\} \\ Class_2 &= \{Czech, German, French, Italian\} \end{aligned} \tag{2.4}$$

There are many ways of how to compute the classes – usually, it is assumed that similar words appear in similar context. However, there are two problems. Firstly, the optimality criterion must be defined. This criterion depends on the task that is being solved. The second problem is the complexity of the problem. The number of possible partitioning rises exponentially[4] with the number of elements in the set. It is therefore impossible to examine every possible partitioning of even a decently large set. The task is then to find a computationally feasible algorithm that would be as close to the optimal partitioning as possible. Combination of word- and class-based language models gives promising results [Maltese et al., 2001].

In [Brown et al., 1992] the MMI[5] clustering algorithm was introduced.

[4]To be exact, the number of possible partitioning of a n-element set is given by the Bell number, which is defined recursively: $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$.

[5]Maximum Mutual Information

The algorithm is based upon the principle of merging a pair of words into one class according to the minimal mutual information loss principle.

The algorithm gives very satisfactory results and it is completely unsupervised. This method of word clustering is possible only on very small corpora and is not suitable for large vocabulary applications. The authors in [Yokoyama et al., 2003] used the MMI algorithm to build class-based language models.

3 Neural Networks

Neural networks is the name of a biologically-inspired programming paradigm which enables a computer to learn from observational data.

The simplest definition of a neural network, respective 'artificial' neural network (ANN), is provided by the inventor of one of the first neurocomputers, Robert Hecht-Nielsen. In [Hecht-Nielsen, 1990] he defines a neural network as:

"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."

3.1 Introduction

The architecture of neural networks is composed from neurons, layers and connections. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning. Either the *sigmoid* or *tanh* function is commonly used as an activation function that converts a neuron's weighted input to its output activation, similarly to logistic regression (see Section 3.2). More information about neurons (respective perceptrons) and neural network architectures can be found in our technical report [Svoboda, 2016].

The main motivation is to simply come up with more precise way how to represent and model words, documents and language than the basic machine learning approaches. Like other machine learning methods – systems that learn from data – neural networks have been used to solve a wide variety of tasks, in this thesis we will however focus on NLP problems. There is nothing that neural networks can do in NLP that the basic machine learning techniques completely fail at, but in general neural networks and deep learn-

ing currently provide the best solutions to many problems in NLP. We can benefit from those gains and see it as an evolution in machine learning.

3.2 Machine Learning

This section describes a brief indroduction into Machine learning and basic classifiers. For more detailed description including derivations for the math can be found in our report [Svoboda, 2016], for most of our implementations we used *Bainy* library presented in [Konkol, 2014].

Machine learning explores the study and construction of algorithms that can learn from input data and make predictions on data. Such algorithms operate by building a model from the example data during a training phase. New inputs is given to the resulting model in order to make data-driven predictions or decisions expressed as outputs. This is achieved by observing the properties from labeled training data, this learning technique is called supervised learning. Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Creating a manually annotated dataset is generally a hard and time-consuming task. However most of current NLP problems are being solved based on annotated data sets which have been annotated by humans. Often such datasets small and speciaized and (together with features developed for NLP task) tend to be over-tuned for the specific data set and fail to generalise to new examples.

With supervised learning, the acquired knowledge is later applied to determine the best category for the unseen testing dataset. For unsupervised learning there is no error or reward signal to evaluate potential solution, the goal is to model input data. Commonly used unsupervised learning algorithms are artificial neural network models, about which we will talk more in Section 3.3.

Machine learning techniques applied to NLP often use n-gram language models, word clustering and basic bag-of-words representations as basic feature representation and further infer more complicated features.

One basic machine learning technique to perform classification is logistic regression that is described in next section (also commonly referred as Maximum Entropy classifier). Later, we describe Naive Bayes Classifier and SVM Classifier.

3.2.1 Logistic Regression Classifier

The Logistic Regression Classifier is based on the maximum entropy principle. The principle says that we are looking for a model which will satisfy all our constraints and at the same time resembles uniform distribution as much as possible. Logistic regression is a probabilistic model for binomial cases, that is, the input is a vector of features, output is usually one – binary classification. A logistic classifier can be trained by stochastic gradient descent. The Maximum Entropy (MaxEnt) generalizes the same principle for multinomial cases.

We want a conditional probability:

$$p(y|\mathbf{x}), \tag{3.1}$$

where y is the target class and x is vector of features.

Logistic regression follows the binomial distribution. Thus, we can write following probability mass function:

$$p(y, \mathbf{x}) = \begin{cases} h_\Theta(\mathbf{x}), & \text{if y = 1,} \\ 1 - h_\Theta(\mathbf{x}), & \text{if y= 0.,} \end{cases} \tag{3.2}$$

where Θ is the vector of parameters, and $h_\Theta(x)$ is the hypothesis:

$$h_\Theta(\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\Theta}^T \mathbf{x})} \tag{3.3}$$

The probability mass function can be rewritten as follows:

$$p(y|\mathbf{x}) = (h_\Theta(\mathbf{x}))^y (1 - h_\Theta(\mathbf{x}))^{1-y} \tag{3.4}$$

We use maximum log-likelihood for N observations to estimate parameters:

$$\begin{aligned} l(\Theta) &= log \left[\prod_{n=1}^{N} (h_\Theta(\mathbf{x}_n))^{y_n} (1 - h_\Theta(\mathbf{x}_n))^{1-y_n} \right] \\ &= \sum_{n=1}^{N} \left[y_n \; log \; h_\Theta(\mathbf{x}_n) + (1 - y_n) \; log \; (1 - h_\Theta(\mathbf{x}_n)) \right] \end{aligned} \tag{3.5}$$

3.2.2 Naive Bayes Classifier

Naive Bayes (NB) classifier is a simple classifier commonly used as a baseline for many tasks. The model computes the posterior probability of a class based on the distribution of words in the given document as shown in equation 3.6, where s is the output label and x is the given document.

$$P(s|x) = \frac{P(\mathbf{x}|s)P(s)}{P(\mathbf{x})} \tag{3.6}$$

$$\hat{s} = \arg\max_{s \in S} P(s) \prod_{x=1}^{n} P(x_i|s) \tag{3.7}$$

The NB classifier is described by equation 3.7, where $\hat{s}$ is the assigned output label. The NB classifier makes the decision based on the maximum a posteriori rule. In other words it picks the label that is the most probable.

3.2.3 SVM Classifier

The support vector machine was one of the most used classifiers until very recently. It is very similar to logistic regression. It is a vector space based machine learning method 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 basic principle. 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.

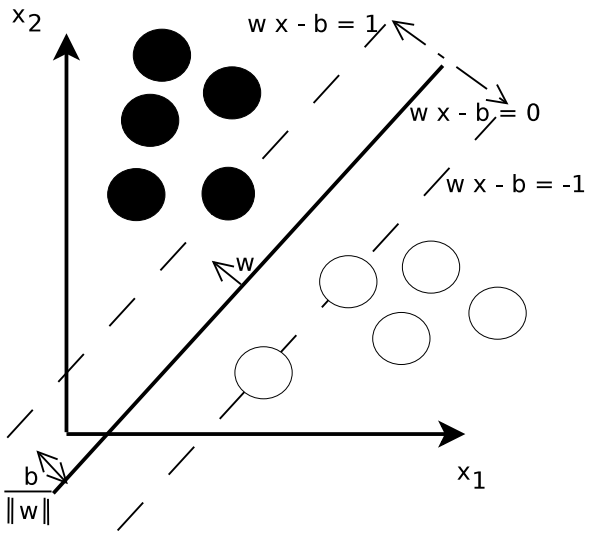


Figure 3.1: Optimal (and suboptimal) hyperplane.

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

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

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

With an SVM we find 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 on 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. Mathematically written as (3.13).

$$label(\mathbf{x}) = \text{sign}(\mathbf{w}_0 \cdot \mathbf{x} + b_0) \tag{3.13}$$

3.3 Training of Neural Networks

The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. There are many ways how to train neural networks [Scalero and Tepedelenlioglu, 1992, Hagan and Menhaj, 1994, Montana and Davis, 1989]. However, the most widely used and successful in practice is stochastic gradient descent (SGD) [Rumelhart et al., 1988].

Training of neural networks involves two stages, the first called the *forward pass* (also called forward propagation)

3.3.1 Forward Pass

- Input vector are presented at first in input layer.
- Forward propagation of a training takes input feature vector through the neural network in order to generate the propagation's output activations. The target vector presents the desired output vector.
- While training we change weights that in another cycle, where the same input vector is presented, the output vector will be closer to the target vector.

The second stage is called *backpropagation* (or also "backward propagation of errors"). Backpropagation [Hecht-Nielsen, 1989] takes output activations through the neural network using the training pattern target in order to generate the deltas (the difference between the targeted and actual output values) of all output and hidden neurons (see at picture 3.2).

3.3.2 Backpropagation

The backpropagation algorithm was originally introduced in the 1970s [Kelley, 1960], but its importance was not fully appreciated for use in artificial neural networks until 1986 [Rumelhart et al., 1988]. That paper describes neural networks where backpropagation works far faster than earlier approaches to learning and makes it possible to use artificial neural networks to solve problems which were not solvable before.

Read our technical report [Svoboda, 2016] to dive into more details about Backpropagation.

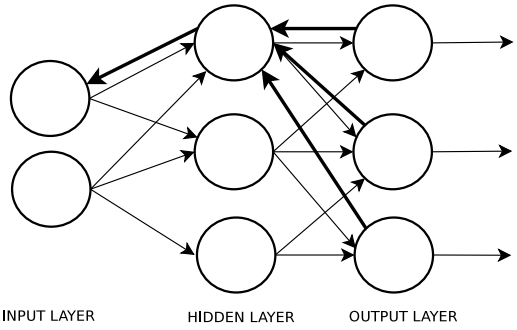


Figure 3.2: Backpropagation

3.3.3 Regularization

While a network is being trained, it often overfits the training data, so it has good performance during training, but fails to generalize on *Test-data*. In Section 3.3.2 we briefly talked about *Held-out data*, but we did not say, why to use them. Simple answer is that we are using them to setup the hyper-parameters – such as α, regularization parameters, cache for RNN and others. To understand why, consider that when setting hyper-parameters we are going to try many different choices for the hyper-parameters. If we set the hyper-parameters based on evaluations of the *Test-data* it is possible we will end up overfitting our hyper-parameters to the *Test-data*. That is, we may end up finding hyper-parameters which fit particular *Test-data*, but where the performance of the network will not generalize to other data sets. We guard against that by figuring out the hyper-parameters using the *Held-out data* data.

The network itself "memorizes" the training data, after training is finished, it will contain high weights that are used to model only some small subset of data.

We can try to force the weights to stay small during training to reduce this problem.

3.4 Feed-forward Neural Networks

A feedforward neural network is biologically inspired classification algorithm. It consist of a (possibly large) number of simple neuron-like processing *units*, organized in *layers*. It is an artificial neural network where connections between the units do not form a cycle and the network can be seen on figure 3.3. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal: each connection may have a different strength or *weight*. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes.

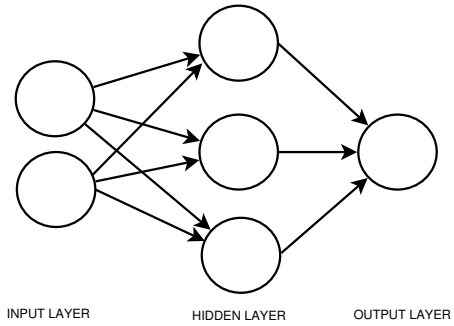


Figure 3.3: Feed-forward Neural Network

Data enters at the inputs and passes through the network, layer-by-layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feedforward neural networks. This is different from recurrent neural networks introduced in following Section 3.6.

Any layer that is not an output layer is a *hidden layer*. The network presented in figure 3.3 has one hidden layer and one output layer. When we have more than one hidden-layer, we talk about Deep-feed-forward Neural Network (see more about Deep-learning in Section 3.7).

3.5 Convolutional Neural Networks

When we hear about Convolutional Neural Network (CNN), we typically think of Computer Vision. CNNs were responsible for major breakthroughs in Image Classification and are the core of most Computer Vision systems [Krizhevsky et al., 2012, Lawrence et al., 1997] today, from Facebook's automated photo tagging [Farfade et al., 2015] to self-driving cars [Bengio, 2009].

More recently NLP community has also started to apply CNNs and gotten some interesting results. A good start is [Zhang and Wallace, 2015] where authors evaluate different hyper parameter settings on various NLP problem. Article [Kim, 2014] evaluates CNNs on various classification NLP problems. In [Johnson and Zhang, 2014] they train CNN from scratch, without need for pre-trained word embeddings. Another use case of CNNs in NLP from Microsoft Research lab can be found in [Gao et al., 2015] and [Shen et al., 2014]. They describe how to learn semantically meaningful representations of sentences that can be used for Information Retrieval.

A detailed overview of CNN networks and their use in NLP is presented in technical report [Svoboda, 2016].

3.6 Recursive Neural Networks

Recursive Neural Networks (RNN) are popular in NLP due to their capability for processing arbitrary length sequences. The idea behind RNNs is to make use of sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But for many tasks that is not ideal, especially in NLP tasks. If you want to predict the next word in a sentence you better know which words came before it. RNNs operate with each element of the sequence being presented to the input nodes of the RNN in turn. They are called recurrent because values computed from each element is carried over to the computation for the next element. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far. In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps, because it is also often claimed that learning long-term dependencies by stochastic gradient descent can be difficult [Bengio et al., 1994].

For Language Modeling [Mikolov et al., 2010] a so called *simple recurrent neural network* (see figure 3.4) or Elman network [Elman, 1990] is being used.

3.6.1 RNNs with Long Short-Term Memory

Long Short-Term Memory (LSTM) units [Hochreiter and Schmidhuber, 1997] have re-emerged as a popular architecture due to their representational power and effectiveness at capturing long-term dependencies. LSTMs do not have a fundamentally different architecture from RNNs, but they use a different function to compute the hidden state. There are many LSTM architectures, some evaluation of different architectures has been done in [Jozefowicz et al., 2015].

The memory in LSTMs are called cells and they take as input the previous state hs_{t-1} and current input x_t. Internaly these cells decide what to keep in (and what to erase from) memory. They then combine the previous state, the current memory, and the input.

In a traditional recurrent neural network, during the gradient phase of back-propagation, the gradient signal can end up being multiplied a large number of times (as many as the number of timesteps) by the weight matrix associated with the connections between the neurons of the recurrent hidden layer. This means that, the magnitude of weights in the transition matrix can have a strong impact on the learning process.

When the weights in this matrix are small (if the leading eigenvalue of the weight matrix is smaller than 1), it can lead to a situation called vanishing gradients [Bengio et al., 1994] where the gradient signal gets so small that learning either becomes very slow or stops working altogether. It can also make more difficult the task of learning long-term dependencies in the data. Conversely, if the weights in this matrix are large (or, again, more formally, if the leading eigenvalue of the weight matrix is larger than 1), it can lead to a situation where the gradient signal is so large that it can cause learning to diverge. This is often referred to as exploding gradients.

These issues are the main motivation behind the LSTM model which introduces a new structure called a *memory cell* (fig. 3.5). Cells take as input the previous state h_{t-1} and current input x_t. Internally these cells decide what to keep in (and what to erase from) memory. They then combine the previous state, the current memory, and the input.

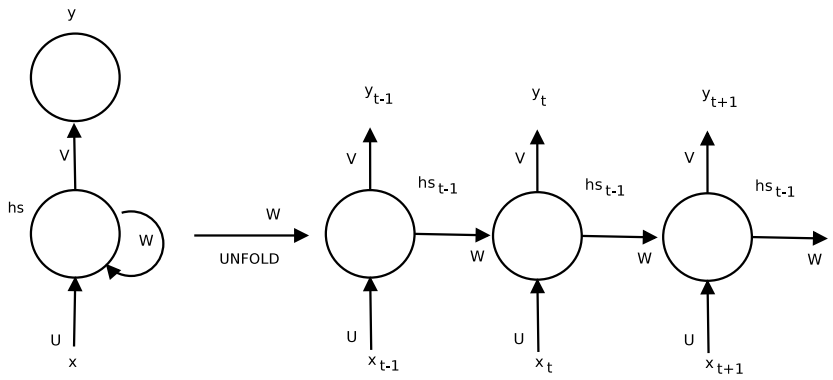


Figure 3.4: Picture shows a RNN being unrolled (or unfolded) into a full network. By unrolling we simply mean that we write out copies of the network for the complete sequence. For example, if the sequence we care about is a sentence of 5 words, the network would be unrolled into a 5-stage neural network, one stage for each word. On the picture we see:

- $\boldsymbol{x_t}$ is the input at time step t. For example for language modeling, $\boldsymbol{x_1}$ could be seen as a vector corresponding to the second word of a sentence.
- $\boldsymbol{hs_t}$ is the hidden state at time step t. It is the networks "memory" (captures information about what happened in all the previous time steps0) and it is calculated based on the previous hidden state and the input at the current step: $hs_t = f(Ux_t + Whs_{t-1})$, where the f is usually our well known nonlinearity function such as tanh. hs_{-1}, which is required to calculate the first hidden state, is typically initialized to all zeroes.
- $\boldsymbol{y_t}$ is the output at step t. For example, if we wanted to predict the next word in a sentence, it would be a vector of probabilities across our vocabulary, $\boldsymbol{y_t} = softmax(Vhs_t)$. Output is calculated based on the memory at time t, but it is more complicated in practice, because hs_t can not capture information from too many time steps ago (explained in Section 3.6.1). $Softmax$ regression is a probabilistic method with function similar to the Logistic regresion, we use the softmax function to map inputs to the the predictions (can be multinomial).
- U and W are parameters of RNN that are shared across the whole network and are not different at each layer as it is for example in Feed-forward Neural Networks and its weight parameters.

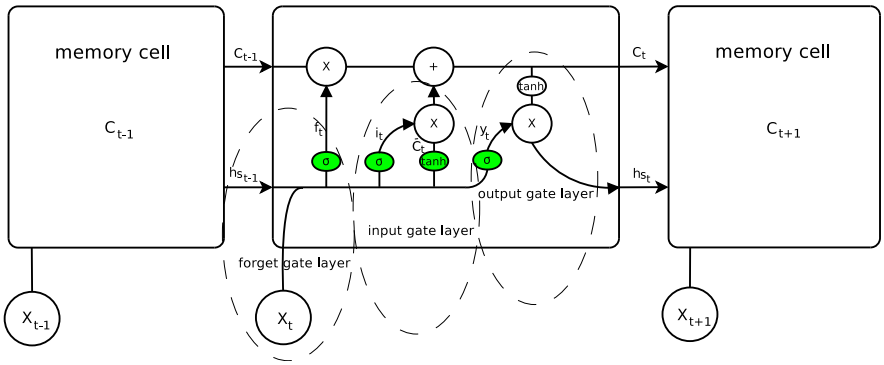


Figure 3.5: LSTM memory cell. Green boxes represent learned neural network layers, while circles inside a cel represents pointwise operations.

The forget gate is one of the most important features of the LSTM network [Greff et al., 2015]. It makes the decision what information we are going to throw away from the cell state. The input gate layer decides which values we will update (which information we keep). It has turned out that these types of units are very efficient at capturing long-term dependencies.

Mathematical background of LSTM and further information has been presented in our technical report [Svoboda, 2016].

3.7 Deep Learning

Deep learning algorithms attempt to learn multiple levels of representation of increasing complexity (or abstraction of the problem) [LeCun et al., 2015]. Most current machine learning techniques require human-designed representations and input features. Machine learning then just optimizes the weights to produce the best final prediction. Machine Learning methods thus are heavily dependent on quality of input features created by humans.

Deep Belief Networks (DBNs), Markov Random Fields with multiple layers, various types of multiple-layer neural networks are techniques which has more than one hidden layer and are able to model complex non-linear problems. Deep architectures can, in principle, represent certain families of functions more efficiently (and with better scaling properties) than shallow ones, but the associated loss functions are almost always non convex. Deep learning is practically putting back together representation learning with machine

learning. It tries to learn good features, across multiple levels of increasing complexity and abstraction (hidden layers) of the problem [Bengio et al., 2007].

The hidden layers represent learned non-linear combination of input features. With hidden layers, we can solve non-linear problems (such as XOR):

- Some neurons in the hidden layer will activate only for some combination of input features.
- The output layer can represent combination of the activations of the hidden neurons.

A neural network with one hidden layer is a *universal approximator*. The *universal approximator* theorem for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer. However, not all functions can be represented efficiently with a single hidden layer – thus deep learning architectures can achieve better accuracy for complex problems.

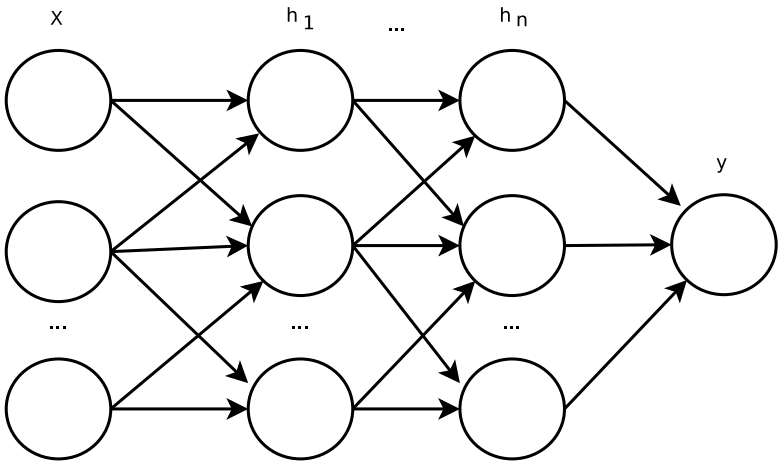


Figure 3.6: Deep neural network. X represents the input layer, $h_1, h_2, \ldots h_n$ represents hidden layers and y denotes to output layer

In recent work, deep LSTM networks are being used, often bidirectional deep recurrent (LSTM) networks [Tai et al., 2015a]. Bidirectional RNNs are based on the idea that the output at time t may not only depend on the

previous elements in the sequence, but also future elements. For example, to predict a missing word in a sequence you want to look at both the left and the right context. Bidirectional RNNs are quite simple. They are just two RNNs stacked on top of each other. The output is then computed based on the hidden state of both RNNs. Deep (Bidirectional) RNNs are similar to Bidirectional RNNs, only that we now have multiple layers per time step. In practice this gives us the higher learning capacity already mentioned (but we also need a lot of training data).

3.7.1 Representations and Features Learning Process

Developing good features is a hard and time-consuming process. Features are eventually over-specified and incomplete anyway. In NLP research we usually after some time can find and tune features for a manually annotated corpus dealing with some NLP problem. However, we will often find that developed features were over specified for the concrete corpus and fail in generalization for a real application.

If machine learning could learn features automatically, the learning process could be automated more easily and more tasks could be solved. Deep learning provides one way of automating the feature learning process. Usually, we need big datasets for deep learning to avoid over-fitting. Deep neural networks have many parameters, therefore if they don't have enough data, they tend to memorize the training set and perform poorly on the test set.

3.8 Distributional Semantics Models Based on Neural Networks

Many models in NLP are based on counts over words, for example, Probabilistic Context Free Grammars (PCFG) [Manning et al., 1999]. In those approaches it can hurt generalization performance when specific words during testing were not present in the training set. Because an index vector over a large vocabulary is very sparse, models tends to overfit to the training data. The classical solutions to the problem is the already mentioned time consuming manual engineering of complex features. Deep Learning models of language usually use distributed representation (see 2.1.1). These are methods for learning word representations in which meaning of words or phrases

is represented by vectors of real numbers, where the vector reflects the contextual information of a word across the training corpus.

These word vectors can significantly improve and simplify many NLP applications [Collobert and Weston, 2008, Collobert et al., 2011]. There are also NLP applications, where word embeddings do not help much [Andreas and Klein, 2014].

Recent studies have introduced several methods based on the feed-forward NNLP (Neural Network Language Model). One of the Neural Network based models for word vector representation which outperforms previous methods on word similarity tasks was introduced in [Huang et al., 2012]. The word representations computed using NNLP are interesting, because trained vectors encode many linguistic properties and those properties can be expressed as linear combinations of such vectors.

Nowadays, word embedding methods Word2Vec [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014] significantly outperform other methods for word embeddings. Word representations made by these methods have been successfully adapted on variety of core NLP task such as named entity recognition [Siencnik, 2015, Demir and Ozgur, 2014], Part-of-speech Tagging [Al-Rfou et al., 2013], sentiment Analysis [Pontiki et al., 2015], and others.

There are also neural translation-based models for word embeddings [Cho et al., 2014, Bahdanau et al., 2014] that generate an appropriate sentence in the target language given a sentence in the source language, while they learn distinct sets of embeddings for the vocabularies in both languages. Comparison between monolingual and translation-based models can be found in [Hill et al., 2014].

In following sections, we will introduce current state-of-the-art Word Embedding methods called Word2Vec [Mikolov et al., 2013a] and other methods for sentence representation.

3.8.1 Vector Similarity Metrics

The distance (similarity) between two words can be calculated by a vector similarity function. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ denote the two vectors to be compared and $S(\boldsymbol{a}, \boldsymbol{ab})$ denote their similarity measure.

Such a metric needs to be symmetric: $S(\boldsymbol{a}, \boldsymbol{b}) = S(\boldsymbol{b}, \boldsymbol{a})$.

There are many methods to compare two vectors in a multi-dimensional vector space. Probably the simplest vector similarity metrics are the familiar Euclidean ($r = 2$) and city-block ($r = 1$) metrics

$$S_{mink}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt[r]{\sum |a_i - b_i|^r}, \tag{3.14}$$

that come from the Minkowski family of distance metrics.

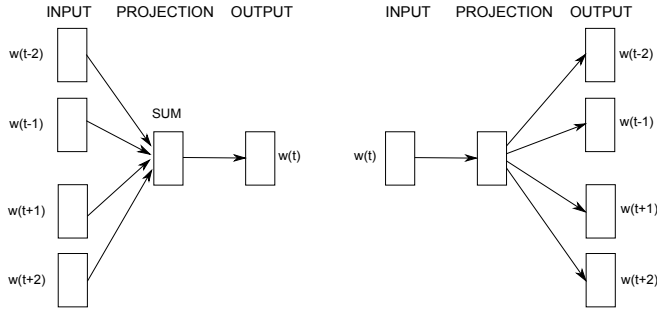
Another often used metric characterizes the similarity between two vectors as the cosine of the angle between them. Cosine similarity is probably the most used similarity metric for words embedding methods:

$$S_{\cos}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2 \sum b_i^2}}, \tag{3.15}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two vectors we try to compare. The cosine similarity is used in all cases where we want to find the most similar word (or top n most similar words) for a given type of analogy.

3.8.2 CBOW

The CBOW (Continuous Bag-of-Words) [Mikolov et al., 2013a] architecture forfinding word embeddings tries to predict the current word from a small context window around the word. The architecture is similar to the feed-forward NNLM (Neural Network Language Model) which was proposed in paper [Bengio et al., 2006]. The NNLM is computationally expensive between the projection and the hidden layer. Thus, in the CBOW architecture, the (non-linear) hidden layer is removed (or in reality is just linear) and projection layer is shared between all words. The word order in the context does not influence the projection (see Figure 3.7a). This architecture proved to have low computational complexity.



(a) CBOW (b) Skip-gram

Figure 3.7: Nerual network based architectures, $w(t-1)$ represents previous word, $w(t)$ current word and $w(t+1)$ next word.

3.8.3 Skip-gram

The Skip-gram architecture is similar to CBOW, although 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., 2013c]. 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 3.7b). Skip-gram model estimates the syntactic properties of words slightly worse than the CBOW model, but it is much better for modeling the word semantics on an English test set [Mikolov et al., 2013a,c]. Training of the Skipgram model does not involve dense matrix multiplications 3.7b and that makes training also efficient [Mikolov et al., 2013c], but generaly slower than CBOW architecture.

3.8.4 Fast-Text

FastText tool introduced in [Bojanowski et al., 2017] combines concepts of CBOW (resp. Skip-Gram) architectures introduced earlier in Section 3.8.2 and 3.8.3. In additon to representing contexts with bag of words, it also considers them as a bag of n-grams, thus using subword information, and shares information across classes through a hidden representation.

3.8.5 GloVe

The GloVe (Global Vectors) [Pennington et al., 2014] model is not based on neural network architecture and focuses more on the global statistics of the training data. This approach analyses log-bilinear regression models that effectively capture global statistics and also captures word analogies. Authors propose a weighted least squares regression model that trains on global word-word co-occurrence counts. The main concept of this model is the observation that ratios of word-word co-occurrence probabilities have the potential for encoding meaning of words.

3.8.6 Paragraph Vectors

Paragraph vectors were proposed in [Le and Mikolov, 2014] as an unsupervised method of learning text representation. The article shows how to compute vectors for whole paragraphs, documents or sentences. The resulting feature vector has fixed dimension while the input text can be of any length. The paragraph vectors and word vectors are concatenated to predict the next word in a context. The paragraph token acts as a memory that remembers what information is missing from the current context.

The sentence representations can be further used in classifiers (logistic regression, SVM or NN).

3.8.7 Tree-based LSTM

Tree-structured input for LSTM was presented in [Tai et al., 2015a], where a tree model represents the sentence structure. Dependency parsing is being used as a typical sentence-tree structure representation [De Marneffe et al., 2006]. The LSTM processes input sentences of variable length via recursively apply the hidden state of child nodes to a head node, rather than following the sequential order of words in a sentence, as is common in LSTMs. The model was tested for sentiment analysis and sentence semantic similarity, achieving state-of-the-art results on both tasks.

4 Word Embeddings of Inflected Languages

Word embedding methods have been proven to be very useful in many NLP tasks. Much has been investigated about word embeddings of English words and phrases, but only a little attention has been dedicated to other languages. Our goal in this chapter is to explore the behavior of state-of-the-art word embedding methods on Czech and Croatian, two languages that are characterized by rich morphology. We introduce a new corpus for the word analogy task that inspects syntactic, morphosyntactic and semantic properties of Czech and Croatian words and phrases. We experiment with Word2Vec, Fasttext and GloVe algorithms and discuss the results on this corpus. We added some of the specific linguistic aspects from Czech and Croatian language to our word analogy corpora. All corpora are available for the research community.

In [Svoboda and Brychcín, 2016] we explore the behavior of state-of-the-art word embedding methods on Czech, which is a representative of the Slavic language family (Indo-European languages) with rich word morphology. These languages are highly inflected and have a relatively free word order. Czech has seven cases and three genders. The word order is very variable from the syntactic point of view: words in a sentence can usually be ordered in several ways, each carrying a slightly different meaning. All these properties complicate the learning of word embeddings. We introduced a new corpus for the word analogy task that inspects syntactic, morphosyntactic and semantic properties of Czech words and phrases. We experimented with Word2Vec and GloVe algorithms and discussed the results on this corpus. We showed that while current methods can capture semantics on English in a similar corpus with 76% of accuracy, there is still room for improvement of current methods on highly inflected languages where the models work on less than 38%, respectively 58% for single tokens without phrases (CBOW architecture) presented later in [Svoboda and Brychcín, 2018a].

In [Svoboda and Beliga, 2018] we explore the behavior of state-of-the-art word embedding methods on Croatian that is another highly inflected lan-

guage from the Slavic family. Next, we created Croatian WordSim353 and RG65 corpora for a basic evaluation of word similarities. We compared created corpora on two popular word representation models, based on *Word2Vec* tool and *fastText* tool.
Models were trained on a 1.37 billion tokens training data corpus and tested on a new robust Croatian word analogy corpus. Results show that the models are able to create meaningful word representation. This research has shown that free word order and the higher morphological complexity of Croatian language significantly influences the quality of resulting word embeddings. We showed that there is similarly to Czech language room for improvement of current DSMs as well and proves our theory about highly inflected languages.

The word-analogy-based evaluation is one of the most common tools to evaluate linguistic relationships encoded in monolingual meaning representations. In [Brychcín et al., 2019], we go beyond monolingual representations and generalize the word analogy task across languages to provide a new intrinsic evaluation tool for cross-lingual semantic spaces. Our approach allows examining cross-lingual projections and their impact on different aspects of meaning. It helps to discover potential weaknesses or advantages of cross-lingual methods before they are incorporated into different intelligent systems. Furthermore, we generalize the word analogy task across languages, to provide a new intrinsic evaluation method for cross-lingual semantic spaces. We experiment with six languages within different language families, including English, German, Spanish, Italian, Czech, and Croatian. State-of-the-art monolingual semantic spaces are transformed into a shared space using dictionaries of word translations. We compare several linear transformations and rank them for experiments with monolingual (no transformation), bilingual (one semantic space is transformed to another), and multilingual (all semantic spaces are transformed onto English space) versions of semantic spaces. We show that tested linear transformations preserve relationships between words (word analogies) and lead to impressive results. We achieve average accuracy of 51.1%, 43.1%, and 38.2% for monolingual, bilingual, and multilingual semantic spaces, respectively.

The structure of this chapter is following. Section 4.1 puts our work into the context of the state of the art. In Section 4.2 we present the first Czech analogy word corpus, Section 4.3 presents Croatian analogy and similarity corpora. The experimental results are presented and discussed in Sections 4.2.1,4.2.2 for Czech and in Sections 4.3.3,4.3.4 for Croatian language. We conclude in Section 4.5 and offer some directions for future work.

4.1 Introduction

Word representation based on Distributional Hypothesis (see Chapter 2) represent words as vectors of real numbers from high-dimensional space. The goal of such representations is to capture the syntactic and semantic relationship between words.

It was shown that the word vectors can be sucessfully used in order to improve and/or simplify many NLP applications [Collobert and Weston, 2008, Collobert et al., 2011]. There are also NLP tasks, where word embeddings do not help much [Andreas and Klein, 2014].

Most of the work is focused on English. Recently the community has realized that the research should focus on other languages with rich morphology and different syntax [Berardi et al., 2015, Elrazzaz et al., 2017, Köper et al., 2015], but there is still little attention to languages from Slavic family. These languages are highly inflected and have a relatively free word order. Since there are open questions related to the embeddings in the Slavic language family, we will focus mainly on Czech and Croatian word embeddings, from the Slavic language family. With the aim of expanding existing findings about Czech and Croatian word embeddings, we will:

1. Compare different word embeddings methods on Czech/Croatian language that is not deeply explored highly inflected language.

2. For the purposes of the word embeddings experiments, we created three new Croatian datasets and two Czech word analogy datasets. Two basic word similarity corpora based on original WordSim353 [Finkelstein et al., 2002] and RG65 [Rubenstein and Goodenough, 1965] translated to Croatian. Except the similarity between words, we would like to explore other semantic and syntactic properties hidden in word embeddings. A new evaluation scheme based on word analogies were presented in [Mikolov et al., 2013a]. Based on this popular evaluation scheme, we have created a Croatian and Czech version (with and without phrasal words) of original Word2Vec analogy corpus in order to qualitatively compare the performance of different models.

3. Empirically compare the results obtained from the Czech/Croatian language to the results obtained from English – the most commonly studied language.

Nowadays, word embeddings are typically obtained as a product of training feed-forward NNLP (Neural Network Language Models). One of the first architectures was presented in [Huang et al., 2012]. The word representations computed using NNLP are interesting, because trained vectors encode many linguistic properties and those properties can be expressed as linear combinations of such vectors. Language modeling is a classical NLP task of predicting the probability distribution over the "next" word (see Section 2.3). In these models a word embedding is a vector in $\mathbb{R}^n$, with the value of each dimension being a feature that weights the relation of the word with a "latent" aspect of the language. These features are jointly learned from plain unannotated text data. This principle is known as the *Distributional Hypothesis* [Harris, 1954](see Chapter 2).

There is a variety of datasets for evaluating semantic relatedness between English words, such as:

- *WordSimilarity-353* [Finkelstein et al., 2002],
- *Rubenstein and Goodenough (RG)* [Rubenstein and Goodenough, 1965],
- *Rare-words* [Luong et al., 2013],
- *Word pair similarity in context* [Huang et al., 2012],
- and many others.

[Mikolov et al., 2013a] reported that word vectors trained with a simplified neural language model [Bengio et al., 2006] encode syntactic and semantic properties of language, which can be recovered directly from the vector space through linear translations, to solve analogies such as: $\vec{king} - \vec{man} = \vec{queen} - \vec{woman}$. This evaluation scheme based on word analogies was presented in [Mikolov et al., 2013a].

To the best of our knowledge, only a small portion of recent studies attempted evaluating Croatian and Czech word embeddings. In [Zuanovic et al., 2014] the authors translated small portion from the English analogy corpus to Croatian to evaluate their neural network based model. However, this translation was only made for a total of 350 questions.

Many methods have been proposed to learn such word vector representations. One of the neural network based models for word vector representation which outperforms previous methods on word similarity tasks was

introduced in [Huang et al., 2012]. Word embeddings methods implemented in tool *Word2Vec* [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014] significantly outperform other methods for word embeddings. Word vector representations made by these methods have been successfully adapted on variety of core NLP tasks. The recent library *FastText* [Bojanowski et al., 2017] tool is derived from Word2Vec and enriches word embeddings vectors with subword information.

In this work we will focus on CBOW, Skip-gram and Glove monolingual models (see Sections 3.8.2, 3.8.3 and 3.8.5) that produce high quality word embeddings. In general, given a single word in the corpus, these models predict which other words should serve as a substitution for this word.

4.2 Czech Word Analogy Corpus

In this section we present a new Czech word analogy corpus for testing word embeddings. Inspiration was taken from English corpus revealed in [Mikolov et al., 2013a]. We follow the observation that the state-of-the-art models for word embeddings can capture different types of similarities between words. Given two pairs of words with the same relationship as a question: Which word is related to *export* in the same sense as *minimum* is related to *maximum*? Correct answer should be *import*.

Such a question can be answered with a simple algebraic operation with the vector representation of words:

$$\boldsymbol{x} = \text{vector}(\text{“maximum”}) - \text{vector}(\text{“minimum”}) + \text{vector}(\text{“export”}) \quad (4.1)$$

The difference between vector(“maximum”) and vector(“minimum”)
should be similar to difference between vector(“export”) and vector(“import”). For resulting vector $\boldsymbol{x}$ we search in the vector space for the most similar word. When the model works well and is properly trained, we will find that the closest vector representing correct answer for our question is the vector for the word *import*.

If the model has sufficient data, it is able to learn also more complicated semantic relationships between words, such as the main city *Prague* to the

state *Czech Republic* is in the same relation as *Paris* is to *France*, or capturing the presidents of individual states, already mentioned antonyms, plural versus singular words, gradation of adjectives, and other word relationships.

To measure quality of word vectors, we designed test set containing 8,705 semantic and 13,552 syntactic questions, together than 22,257 combinations of questions. The dataset contains only frequent-enough words from the Czech Wikipedia. We split the dataset into several categories. Each category usually contains about 35–40 pairs of words with same relationship. The questions are built from all combinations of word pairs in the same category.

There is a majority of word-to-word relationships, but *Presidents and states* category contains also bigram-to-word (word-to-bigram) relationships such as *Prague* vs. *Czech Republic*.

Semantic questions are represented in categories:

- `Presidents-states-cities:` Consists of 34 pairs of states in Europe and their main cities combining 1,122 questions. There is also 1,122 questions for state with corresponding current president.
- `Antonyms:` This category compounds of three subcategories. In first subcategory we have 38 noun antonym pairs that is resulting in 1406 questions combined. Example of such question is: *anode, cathode* versus *export, import*. Similarly we have 42 adjectives pairs (such as *big, small*) and 34 verb pairs – *buy, sell* versus *give, take*.
- `Family-relations (man-woman):` In this category we have 19 pairs of family representatives with man-woman relation as *brother, sister* versus *husband, wife*.

Syntactic questions are represented in categories:

- `Adjectives-gradation:` In this category we have two antonym pairs with three degrees of adjectives in positive, comparative, and superlative form: *big, bigger* vs *small, smaller*.
- `Nationalities (woman/man):` This category is specific for Czech language, which distinguish between masculine and feminine word relations. Every nationality has its corresponding masculine and feminine word form. For example, English word *Japan* has in Czech masculine form *Japonec* and feminine form *Japonka*. We have 35 such pairs.

- `Nouns-plural:` We find here 37 pairs of nouns and their plural forms.
- `Jobs:` Category with 35 pairs of professions with masculine-feminine word relations.
- `Verb-past:` This category consists of verbs in present form versus verbs in past tense form, such as *play, played* versus *see, saw*.
- `Pronouns:` Last category consists of pairs of pronouns in singular versus plural form.

4.2.1 Experiments

In our experiments, we used unsupervised learning of word-level embeddings using Word2Vec [Mikolov et al., 2013a] and GloVe tool [Pennington et al., 2014]. We used the January 2015 snapshot of the Czech Wikipedia as a source of unlabeled data. The Wikipedia corpus has been preprocessed with the following steps:

1. Removed special characters such as *#$&%*, HTML tags and others.
2. Filtering XML dumps, removed tables, links converted to normal text. We lowercase all words. We have also removed sentences with less than 5 tokens.

The resulting training corpus contains about 2,6 billion words. For our purpose, it is useful to have vector representation of word phrases, i.e. for bigram representing state *Czech Republic*, it is desirable to have one vector representing those two words. This was achieved by preprocessing the training data set to form the phrases using the *Word2Phrase* tool [Mikolov et al., 2013c]. We have to note that due to the preprocessing of the corpus using *Word2Phrase* tool, we have lost a lot of usefull single-token words, or those words were not obtained in sufficient frequency to train a robust word embeddings. Therefore, we have lower score than in other articles mentioned in further chapters, where we have used only our non-phrasal corpora for testing word-analogies. Word2Phrase tool would have to be additionaly tuned for Slavic family languages to make better phrasal word representations.

We evaluate the word embedding models on our corpus by accuracy that is defined as

$$\text{Acc} = \frac{\text{NC}}{\text{NT}}\,[\%]\,, \tag{4.2}$$

where NC is the number of correctly answered questions for a category and NT is total number of questions in category.

In our experiments, we use *cosine similarity* (see 3.15) as a measure of similarity between two word vectors.

Models settings

During the training of Word2Vec (resp. GloVe) models, we limited the size of the vocabulary to 400,000 most frequent single token words and about 800,000 most frequent bigrams. OOV (Out-of-vocabulary) word rate was 6%. That means that out of 22,257 questions about 1,300 questions had at least one word not seen in the vocabulary.

To train word embedding methods we use context window of size 10. We also explore results with different vector dimension (set to 100, 300, and 500). We choose to compare three training epochs as in [Mikolov et al., 2013a] for similarly sized training corpus versus ten training epochs for Word2Vec tool. For GloVe tool we choose 10 and 25 iterations, because algorithms cannot be simply compared with the same settings [Pennington et al., 2014]. Other Word2Vec and GloVe settings were on their default values.

Results

In this section we present the accuracies for all tested models (CBOW, Skip-gram, and GloVe) on our word analogy corpus. In all tables below we present results for different vector dimension ranging between 50 and 500, except for Skip-gram model with dimension 500 and 10 training epochs, where the time of computation was much higher than with other methods. The model did not finish after 4 days of training and results of 500 dimension vector does not adequately reward such long training time. We use notation n_D in the tables, n means that the correct word must be between n most similar words for a given analogy. D denote the dimension of vectors. Accuracies are expressed in percents.

In Table 4.1 we present the results for CBOW model. There is a significant improvement between 3 and 10 training epochs. Interesting is also fact that 300-dimensional vectors perform better than 500-dimensional vectors on most categories. Similarly, the results for Skip-gram model are in Table 4.2. This model performs significantly worse on most categories in comparison with CBOW model. There is also significant improvement between 50-dimensional and 100-dimensional vector, but less significant between 100 and 300. Table 4.3 shows result for GloVe model. This model gives on Czech the worst results compared to both Word2Vec models.

Categories, where the models gives best results are *Verb-past*, *Noun-plural*, and *State-city*. In general, all models gives better results on tasks exploring syntactic information. Poor accuracy was in categories *State-presidents* and category *Nationality*.

Type	3 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	1.35	4.84	5.55	5.69	3.98	10.88	13.16	10.95	5.69	13.44	16.00	13.30
Anton. (adj.)	4.82	8.86	11.79	13.24	10.63	14.29	18.70	19.16	13.24	17.31	23.64	22.76
Anton. (verbs)	0.20	1.88	2.68	1.25	2.77	3.13	6.25	3.57	2.94	3.84	7.77	4.38
State-president	0.00	0.00	0.18	0.09	0.18	0.00	0.98	0.18	0.45	0.27	1.43	0.71
State-city	14.62	14.8	16.22	8.47	29.77	30.93	32.89	23.26	35.92	39.57	42.96	31.82
Family	6.42	9.01	11.60	9.26	12.10	17.28	21.85	18.64	14.44	21.11	25.80	23.95
Noun-plural	34.46	42.42	41.74	44.60	45.95	53.60	54.35	54.35	50.45	57.43	57.43	57.81
Jobs	2.95	3.87	3.37	2.78	6.57	10.52	10.00	8.92	9.18	14.05	13.80	12.37
Verb-past	14.83	24.29	42.52	34.91	29.94	40.91	60.61	52.00	36.66	48.31	66.50	58.80
Pronouns	1.59	3.84	5.95	3.57	3.97	8.07	12.70	10.05	5.69	9.66	16.00	13.10
Adj.-gradation	12.50	20.00	22.50	15.00	20.00	22.50	22.50	27.50	20.00	27.50	25.00	27.50
Nationality	0.08	0.42	0.33	0.16	0.84	0.92	0.84	1.10	1.26	1.26	1.26	2.01
	10 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	3.84	7.82	8.53	7.40	8.39	15.93	18.49	16.07	10.38	19.42	22.76	20.55
Anton. (adj.)	7.26	11.90	15.45	15.04	13.53	19.63	25.49	23.58	16.49	23.05	30.26	28.92
Anton. (verbs)	0.89	1.88	2.86	3.12	4.01	5.98	6.43	6.07	5.09	6.70	7.59	7.41
State-president	0.18	0.35	0.09	0.09	0.71	0.98	0.62	0.71	1.16	1.60	1.33	1.16
State-city	16.58	27.99	25.94	18.63	37.07	50.62	52.05	39.13	43.49	58.47	61.41	50.71
Family	11.85	15.43	15.68	15.93	19.75	25.55	30.99	29.13	25.56	30.12	38.02	36.42
Noun-plural	50.23	56.68	60.56	57.96	63.21	68.92	70.35	66.52	67.87	72.97	74.02	69.14
Jobs	6.73	10.52	6.82	4.04	14.39	19.78	17.68	13.30	17.59	24.24	23.06	19.36
Verb-past	25.87	38.71	48.53	48.71	46.92	58.95	69.34	68.78	55.10	66.75	76.00	74.94
Pronouns	5.03	6.22	7.80	7.14	10.71	12.17	15.61	15.48	13.76	16.53	19.31	19.84
Adj.-gradation	25.00	25.00	20.00	17.50	25.00	25.00	27.50	25.00	25.00	30.00	32.50	27.50
Nationality	0.67	1.26	0.34	0.42	2.35	2.60	1.68	2.35	3.03	3.19	3.27	2.77

Table 4.1: Results for CBOW.

4.2.2 Discussion

How to achieve better accuracy? It was shown in [Mikolov et al., 2013c] that sub-sampling of the frequent words and choosing larger Negative Sampling

Type	3 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	0.85	1.71	3.34	5.55	2.20	3.84	8.04	10.74	2.92	5.41	9.67	14.08
Anton. (adj.)	2.26	3.02	5.23	8.48	4.59	5.69	9.00	12.37	6.21	7.14	11.32	14.81
Anton. (verbs)	0.18	0.36	0.36	0.98	0.27	1.61	0.45	2.05	0.89	1.79	0.89	2.68
State-president	0.18	0.18	0.09	0.09	0.53	0.71	0.36	0.62	0.62	1.16	0.71	0.80
State-city	6.60	14.26	8.20	3.48	17.20	27.27	18.89	12.75	22.99	33.69	25.94	21.93
Family	1.98	2.72	2.59	6.79	3.70	6.30	9.01	12.59	6.30	8.52	12.72	16.42
Noun-plural	8.11	14.04	19.14	18.77	15.17	24.62	27.25	36.41	18.17	29.05	31.23	44.59
Jobs	1.77	1.26	1.09	1.01	5.05	3.96	3.45	3.53	6.40	5.81	4.88	5.39
Verb-past	1.72	4.36	4.14	6.08	4.20	8.28	7.67	12.74	6.04	10.62	9.90	19.97
Pronouns	0.79	1.06	0.66	0.40	2.78	2.25	1.72	1.72	3.97	4.23	2.65	2.78
Adj.-gradation	2.50	5.00	5.00	10.00	5.00	7.50	12.50	17.50	5.00	12.50	12.50	25.00
Nationality	0.17	0.08	0.08	0.00	0.84	0.67	0.17	0.42	1.26	1.01	0.25	0.92
	10 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	1.35	2.63	6.19	x	3.27	5.83	10.24	x	4.41	7.25	12.23	x
Anton. (adj.)	1.74	4.82	5.69	x	4.53	9.12	10.05	x	5.57	11.85	12.54	x
Anton. (verbs)	0.36	0.00	0.18	x	0.98	1.96	0.36	x	1.52	2.95	0.62	x
State-president	0.27	0.09	0.27	x	1.07	0.36	0.80	x	1.52	0.62	1.60	x
State-city	4.55	15.15	9.98	x	14.26	31.73	25.85	x	19.88	39.48	35.29	x
Family	3.09	3.70	6.67	x	6.30	9.14	13.46	x	10.37	12.22	16.54	x
Noun-plural	19.22	29.95	23.95	x	31.91	43.92	37.91	x	37.39	47.75	44.59	x
Jobs	2.53	3.03	2.53	x	6.99	7.58	4.88	x	9.93	10.44	7.58	x
Verb-past	2.93	8.25	8.77	x	7.41	15.15	16.69	x	9.73	18.72	20.84	x
Pronouns	0.66	0.66	0.79	x	2.65	2.25	3.44	x	3.84	3.44	4.76	x
Adj.-gradation	2.50	10.00	7.50	x	10.00	15.00	12.50	x	10.00	15.00	15.00	x
Nationality	0.17	0.42	0.08	x	0.50	1.26	0.34	x	0.67	1.60	0.76	x

Table 4.2: Results for Skip-gram.

Type	3 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	0.36	1.28	0.64	0.81	1.00	2.92	1.99	1.72	1.49	4.27	2.63	2.42
Anton. (adj.)	0.87	0.81	1.34	1.34	2.44	4.01	6.10	5.81	3.60	5.40	8.89	7.62
Anton. (verbs)	0.00	0.00	0.00	0.00	0.36	0.00	0.00	0.00	0.36	0.00	0.18	0.00
State-president	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
State-city	1.52	0.98	1.16	0.98	3.83	3.21	4.01	2.85	5.17	4.90	6.68	5.81
Family	3.33	4.20	0.99	1.42	6.67	6.42	4.81	3.85	8.52	8.64	7.41	4.35
Noun-plural	14.79	15.32	12.69	5.54	24.47	26.35	25.83	14.30	28.53	31.46	33.03	18.70
Jobs	0.67	0.25	0.00	0.00	1.43	0.76	0.08	0.00	1.68	1.09	0.17	0.00
Verb-past	5.39	6.96	3.15	0.82	11.59	13.71	7.72	2.78	15.11	17.70	10.80	4.71
Pronouns	0.79	0.66	0.00	0.00	1.59	1.32	1.46	0.00	2.12	1.72	2.38	0.00
Adj.-gradation	7.50	7.50	5.00	0.00	10.00	12.50	7.50	7.50	10.00	12.50	10.00	7.50
Nationality	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.08	0.17	0.00	0.00
	25 training epochs											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
Anton. (nouns)	0.50	0.85	1.14	1.42	1.28	2.70	4.69	4.05	1.71	4.34	6.33	5.62
Anton. (adj.)	1.68	2.67	1.34	1.34	3.83	6.68	6.56	6.21	5.28	7.96	9.87	8.65
Anton. (verbs)	0.18	0.00	0.00	0.00	0.36	0.18	0.09	0.18	0.89	0.18	0.45	0.36
State-president	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
State-city	0.98	1.07	0.98	0.45	3.39	4.19	4.01	2.85	4.99	5.97	7.66	6.51
Family	2.35	3.70	2.10	2.22	5.43	5.80	6.05	4.20	7.04	7.65	8.52	5.56
Noun-plural	28.00	30.56	15.32	6.98	39.79	43.84	29.20	18.02	43.47	48.35	38.44	28.23
Jobs	0.17	0.00	0.00	0.00	0.59	0.42	0.00	0.00	0.76	0.76	1.18	0.51
Verb-past	7.86	10.78	3.98	1.13	16.53	19.25	10.07	4.19	20.82	23.64	14.12	6.81
Pronouns	1.32	1.32	0.26	0.00	3.44	2.25	1.06	0.00	4.76	3.57	1.72	0.00
Adj.-gradation	5.00	5.00	5.00	0.00	7.50	10.00	12.50	7.50	15.00	12.50	12.50	7.50
Nationality	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.3: Results for GloVe.

Type	3 training epochs for CBOW and Skip-gram, 10 training epochs for GloVe.											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
CBOW – semantics	4.77	6.57	8.00	6.33	9.90	12.75	15.64	12.63	12.11	15.92	19.6	16.15
Skip-gram – semantics	2.00	4.75	6.66	x	3.71	7.57	9.62	x	3.30	7.62	10.21	x
GloVe – semantics	1.01	1.21	0.69	0.78	2.38	2.76	2.82	2.56	3.19	3.87	4.30	3.63
CBOW – syntactics	11.06	15.81	19.40	16.84	17.85	22.76	26.84	25.65	20.48	26.37	30.00	28.60
Skip-gram – syntactics	2.51	5.51	6.81	x	4.30	7.88	10.54	x	5.02	8.79	10.24	x
GloVe – syntactics	4.86	5.11	3.50	0.98	8.20	9.11	7.10	3.72	9.59	10.77	9.40	5.26
	10 training epochs for CBOW and Skip-gram, 25 training epochs for GloVe.											
	1_50	**1_100**	**1_300**	**1_500**	**5_50**	**5_100**	**5_300**	**5_500**	**10_50**	**10_100**	**10_300**	**10_500**
CBOW – semantics	6.77	10.90	11.42	10.03	13.91	19.78	22.35	19.12	17.02	23.23	26.90	24.20
Skip-gram – semantics	1.89	4.40	4.83	4.23	5.07	9.69	10.13	8.52	7.21	12.40	13.14	11.79
GloVe – semantics	0.95	1.38	0.93	0.90	2.38	3.26	3.57	2.92	3.32	4.35	5.47	4.45
CBOW – syntax	18.92	23.07	24.01	22.63	27.10	31.24	33.69	31.9	30.34	35.61	38.03	35.59
Skip-gram – syntax	4.67	8.72	7.27	6.04	9.91	14.19	12.63	12.05	11.93	16.16	15.59	15.94
GloVe – syntax	7.06	7.94	4.09	1.35	11.31	12.63	8.81	4.95	14.14	14.80	11.33	7.17

Table 4.4: Accuracy on semantic and syntactic part of corpus.

window helps to improve performance. Also, adding much more text with information related to particular categories would help (see [Pennington et al., 2014]), especially for class *State-presidents*.

In paper [Svoboda and Brychcín, 2016], we focused more on how number of training epochs influences overall performance in respect to the reasonable time of training and how vector embeddings hold semantics and syntactic information of individual Czech words (with respect to dimension of vector). We have a relatively large corpus for training so we choose 10 iterations (respectively 25 for GloVe) as maximum to compare. To train such models can take more than 3 days with Core i7-3960X, especially for Skip-gram model and vector dimension set to 500. We also do not expect much improvement with more iterations on our corpus, however, we recommend to do more training epochs than is set by default.

As we already mentioned in Section 4.2.1, phrases of Czech language complicates learning and the automatic phrase extraction tool that comes together with Word2Vec merged a lot of word tokens. Therefore, the frequency of single word tokens is much lower and the robustness of word-embeddings representation is not high as in our newer articles [Svoboda and Brychcín, 2019], or look at Table 5.6 where we use the corpora without Czech phrasal words for testing and tuning the word-embeddings accuracy for particular task.

Our goal here was not to achieve maximal overall score, but rather to analyze the behavior of word embedding models on Czech language and to build a first word-analogy corpus to do so. In following text, we discuss, how well these models hold semantic and syntactic information. From results on semantic versus syntactic accuracy (see Table 4.4) we can say that for

Czech the CBOW approach that predicts the current word according to the context window is better, than predicting a word's context based on the word itself as in Skip-gram approach. The results are highly affected by phrasal-words, because Skip-gram approach is usually considered better. We have proven that Skip-gram is also better than CBOW in Czech in our later research [Svoboda and Brychcín, 2019, 2018a].

Accuracy on category *State-president* is very low with all models. We expected to achieve similar results as with category State-city. However, such low score was caused by few simple facts. Firstly, we are missing data, this is supported by argument that this category has 27% OOV of questions, then the probability that resulting word will also be missing in vocabulary is going to be high. Second thing is that even if the correct word for a question is not missing in vocabulary, we have more often different corresponding candidates mentioned as presidents of Czech Republic in training data. For example for a question: *"What is a similar word to Czech as is Belarus Alexandr Lukasenko"* we are expecting word *Milos Zeman*, who is our current president. However the models tells us that the most similar word is a word *president*, which is good answer, but we would rather like to see actual name. When we explore other most similar word, we will find *Vaclav Klaus*, who was our former president, fourth similar word was the word *Vaclav Havel*, our first and famous president of Czech Republic after 1992. Based on those statements we can say that we had lack of data corresponding to current presidents in our training corpus.

Czech language has a lot of synonyms. That is why there is overall better improvement in considering more similar words – TOP 10, rather than just comparing again one word with the highest similarity – TOP 1. Therefore there is a bigger improvement in TOP 1 versus TOP 10 similar words on semantics than there is on syntactic tasks.

The most interesting results are however for a category Nationality, where we compare nationalities in masculine and feminine form. Complete category is covered in vocabulary. However, answers for questions are completely out of topic. For a question which should return feminine form of resident of America, the closest word which model returns is *Oscar Wilde*, respective just his last name, second word is *peacefully philosophy* and another name showing up is *Louise Lasser*. Similar task to category Nationalities with masculine-feminine word form is category Jobs, all models there also perform poorly. This specific task for Czech language seems to be difficult for current state-of-the-art word embeddings methods.

The GloVe model seems to give worse results than Word2Vec models, where on the English analogy task it gives better accuracy [Pennington et al., 2014]. We could probably get better results with tuning the model's properties, but that might be achieved with either presented toolkit.

4.3 Croatian Corpora

4.3.1 Word Analogies

The original Word2Vec analogy corpus is composed of 19,558 questions divided into two tested groups : semantic and syntactic questions, e.g. king : man = queen : woman . Fourth word in question is typically the predicted one.

Our Croatian analogy corpus has 115,085 question divided in the same manner as for English into two tested groups: semantic and syntactic questions.

Semantic questions are divided into 9 categories, each is having around 20 – 100 word question pairs. Combination of question pairs gives overall 36,880 semantics questions:

- `capital-common-countries`: This group consists of 23 of the most common countries. These countries are adopted from english Word2Vec analogies and they have the highest number of occurrences in text in all languages.
- `chemical-elements`: Represents 119 pairs of chemical elements with their shortcut symbol (i.e. O – Oxygen).
- `city-state`: Gives 20 regions (states) inside Croatia and gives one of city example in such region.
- `city-state-USA`: 67 pairs of cities and corresponding states in USA. This category is adopted from original English word analogy test.
- `country-world`: 118 pairs of countries with main cities from all over the world. Translated from original Word2Vec analogies.

- `currency-shortcut`: 20 pairs of state currencies with its shortcut name (i.e. Switzerland – CHF).
- `currency`: 20 pairs of states with their currencies (i.e. Japan – yen). Translated from original EN analogy corpus.
- `eu-cities-states`: 40 word pairs of states from EU and their corresponding main city (i.e. Belgium – Brussels).
- `family`: 41 word pairs with family relation in masculine vs feminine form (i.e. brother – sister).

Syntactic part of corpus is divided into 14 categories, consisting of 78,205 questions:

- `jobs`: This category is language-specific, consist of 109 pairs of job positions in masculine× feminine form.
- `adjective-to-adverb`: 32 pairs of adjectives and related adverb forms.
- `opposite`: 29 pairs of adjectives with their opposites. This category collects words from which is easy to make the opposite usually with the prefix "un" or "in". The corresponding prefix is "ne" in Croatian (i.e. certain – uncertain). Adopted from original EN word analogies.
- `comparative`: 77 pairs of adjectives and their comparative form (i.e. good – better).
- `superlative`: 77 pairs of adjectives and their superlative form.
- `nationality-man`: 84 pairs of states and humans representing their nationalities in masculine form. (i.e. Switzerland – Swiss).
- `nationality-female`: 84 pairs of states and their nationalities in feminine form. This is language specific.
- `past-tense`: 40 pairs of verbs and their past tense form.
- `plural`: 46 pairs of nouns and their plural form.
- `nouns-antonyms`: 100 pairs of nouns and their antonyms.
- `adjectives-antonyms`: Similar category to *opposite*, it consists of 96 word pairs of adjectives and their antonyms. However, words are much more complex (i.e. good – bad).

- `verbs-antonyms`: 51 pairs of verbs and their antonyms.
- `verbs-pastToFemale`: 83 pairs of verbs and their past tense in feminine form. This category is extended from category *past-tense* and is language-specific.
- `verbs-pastToMale`: 83 pairs of verbs and their past tense masculine form. Category is same as past-tense, only its extended variation to be comparable with category *verbs-pastToFemale*.

4.3.2 Word Similarities Corpora

For basic comparison with English, we have translated state-of-the-art English word similarity data sets WordSim353 [Finkelstein et al., 2002] and RG65 [Rubenstein and Goodenough, 1965]. These corpora have 353 (respespective 65) word pairs. Each word pair is manually annotated with similarity. We kept similarities untouched. The words in WordSim353 are assessed on a scale from 0 to 10, in RG65 from 0 to 5.

4.3.3 Experiments

We experimented with state-of-the-art models used for generating word embeddings.
These were neural network based models CBOW and Skip-gram from the Word2Vec [Mikolov et al., 2013a] tool and the tool FastText that promises better score for morphologically rich languages.

Training data

We trained our models on two datasets in the Croatian language. We made the entire dump of Croatian Wikipedia – dated 08-2017 with approximately 275,000 articles. We have tokenized the text, removed nonalphanumeric tokens and extracted only sentences with at least 5 tokens. Resulting corpus has 92,446,973 tokens. We merged data from Wikipedia with the Croatian corpus presented in [Šnajder et al., 2013] that has over 1.2 billion tokens. The resulting corpus has 1.37 billion tokens and 56,623,398 sentences. The corpus has vocabulary of 955,905 words with at least 10 occurences.

For English version of data, we used Wikipedia dump from June 2016. This dump was made of 5,164,793 articles, has 2.2 billion tokens and a vocabulary of 1,759,101,849 words.

We tested analogies and similarity corpora for both languages with most frequent 300,000 words.

Results

	Vocabulary $tf > 10$	Tokens
EN corpus	3,234,907	2,201,735,114
HR corpus	955,905	1,370,836,176

Table 4.5: Properties of Croatian training data corpus.

Model	**CBOW**	**Skip-gram**	**fastText-Skip**	**fastText-CBOW**
Capital	44.17	62.5	59.58	21.25
Chemical-elements	1.02	2.25	0.74	0.41
City-state	22.11	37.89	47.63	46.32
City-state-USA	5.78	8.23	4.30	0.37
Country-world	23.93	44.49	40.15	7.31
Currency	4.68	8.19	6.43	0.58
Currency-shortcut	2.08	8.19	2.50	0.42
EU-cities-states	21.59	41.95	42.33	6.16
Family	34.83	41.82	42.72	34.76
Jobs	68.94	64.06	88.54	95.45
Adj-to-adverb	18.36	21.36	35.33	62.01
Opposite	17.34	18.05	59.03	86.10
Comparative	34.90	33.57	43.22	41.46
Superlative	33.22	27.70	40.50	51.77
Nationality-man	17.01	23.87	60.05	62.13
Nationality-female	14.38	55.66	57.77	53.98
Past-tense	67.31	61.03	66.67	78.21
Plural	37.12	44.65	44.24	35.10
Nouns-ant.	12.70	10.96	10.80	21.24
Adjectives-ant.	13.39	13.11	18.59	12.59
Verbs-antonyms	9.18	6.18	7.25	9.71
Verbs-pastFemale	60.92	19.47	71.04	80.50
Verbs-pastMale	66.68	62.89	76.04	85.04
SEMANTICS_EN	73.63	83.64	68.77	68.27
SYNTACTIC_EN	67.55	66.8	67.94	76.58
SEMANTICS_HR	16.60	28.54	25.94	7.76
SYNTACTIC_HR	37.06	35.63	49.60	54.56
ALL_HR	32.03	33.89	43.83	43.13

Table 4.6: Detailed results of Croatian word analogy corpus.

	English		
Models	WordSim353	RG65	EN-analogies
CBOW	57.94	68.69	69.98 (44.02)
Skip-gram	64.73	78.27	73.57 (46.28)
fastText-Skip	46.13	76.31	68.27 (42.94)
fastText-CBOW	44.64	73.64	76.58 (48.17)
	Croatian		
CBOW	37.61	52.01	32.03 (19.19)
Skip-gram	52.16	58.47	33.89 (20.31)
fastText-Skip	52.98	64.31	43.83 (25.79)
fastText-CBOW	30.41	51.06	43.14 (25.79)

Table 4.7: Comparison with English models. Measurement in brackets gives the results including OOV questions.

4.3.4 Discussion

In total, we tested on 68,986 out of 115,085 questions, which means that almost 40% of questions had OOV words. All question containing OOV words were discarded from testing process. We tested the semantic group on 16,968 questions and the part of the corpus testing syntactic properties was measured on 52,018 questions.

Only 10 out of 353 questions were OOV for the *WordSim353* corpus and all 65 questions of *RG65* were in vocabulary. Unknown words in *WordSim353* were represented as word vector averaged from 10 least common words in vocabulary.

Semantic tests give overall poor performance on all tested models, as we can see in Table 4.6. The opposite is true for English, where semantic tests usually give similar scores as syntactic tests. This behavior we already saw on Czech corpus presented in [Svoboda and Brychcín, 2016]. It seems that free word order and other properties of highly inflected languages from the Slavic family have a big impact on the performance of current state-of-the-art word embeddings methods.
From results of *City-state* and *City-state-USA* category it can be seen that knowledge of the topic in training data has significant impact on performance of a model. We wanted to show differences between two similar categories in case we have an insufficient amount of training data covering a particu-

lar topic. Category *City-state* is showing that model is able to carry such knowledge – if the topic is sufficiently represented in a training data, the model is able to carry this type of information. This behavior is seen in regions from Croatia mentioned in many articles on Croatian Wikipedia, but this was not a case with states from USA. All questions of *City-state* were covered, but only around 50% of questions in category *City-state-USA* were in vocabulary. On categories *Country-world* and *EU-cities-states* it can be seen that there is no difference between knowledge about states and main cities from EU again state-city pairs from all over the world. Another very poor performance gives group *Currency*, but this group is usually weak across all languages and shows the weaknesses of the model.

Syntactic tests give better performance than tests oriented to semantic, but they still have significantly worse performance than on English. This part of corpus includes language-specific group of tests – such as *Verbs-pastMale/Female*, *Nationality-man/female*. Simple *Past-tense* tests gives surprisingly high score – similarly it was also with Czech language in [Svoboda and Brychcín, 2016]. We could say, that languages from Slavic family tends to have easier patterns for past tense. From language-specific groups we see that slightly better score is given in categories with word pairs in the masculine form, these results also corresponds with the fact that there are more articles written in masculine form in the training data.

4.4 Cross-lingual Word Analogies

Lately, research in Distributional Semantics is moving beyond monolingual representations. The research is motivated mainly by two factors:

1. cross-lingual semantic representation enables reasoning about word meaning in multilingual contexts, which is useful in many applications (crosslingual information retrieval, machine translation, etc.)

2. it enables transferring of knowledge between languages, especially from resourcerich to poorly-resourced languages.

In [Brychcín et al., 2019] we experiment with six languages within different language families, including English, German, Spanish, Italian, Czech,

and Croatian. State-of-the-art monolingual semantic spaces are transformed into a shared space using dictionaries of word translations. We compare several linear transformations and rank them for experiments with monolingual (no transformation), bilingual (one semantic space is transformed to another), and multilingual (all semantic spaces are transformed onto English space) versions of semantic spaces. We show that tested linear transformations preserve relationships between words (word analogies) and lead to impressive results. We achieve average accuracy of 51.1%, 43.1%, and 38.2% for monolingual, bilingual, and multilingual semantic spaces.

Several approaches for inducing cross-lingual semantic representation (i.e., unified semantic space for different languages) have been proposed in recent years, each requiring a different form of cross-lingual supervision [Upadhyay et al., 2016]. They can be roughly divided into three categories according to the level of required alignment: a) document-level alignments [Vulić and Moens, 2016], b) sentence-level alignments [Levy et al., 2017], and c) word-level alignments [Mikolov et al., 2013b].

We focus on the last case, where a common approach is to train monolingual semantic spaces independently of each other and then to use bilingual dictionaries to transform semantic spaces into a unified space. Most related works rely on linear transformations [Mikolov et al., 2013b, Faruqui and Dyer, 2014, Artetxe et al., 2016] and profit from weak supervision.

4.5 Conclusion

We made an evaluation of Croatian and Czech word embeddings. New corpora are derived from the original Word2Vec. Additionally, some of the specific linguistic aspects of the Slavic family language were added. We experimented with state-of-the-art methods of word embeddings, namely, CBOW, Skip-gram, GloVe and FastText (see Chapter 7 for Czech results). Models have been trained on a new robust Czech and Croatian analogy corpus. WordSim353 and RG65 corpora were translated from English to Croatian, in order to perform basic semantic measurements. Results show that models are able to create meaningful word representation.

However, it is important to note that paper in this chapter presents the first comparative study of word embeddings for Czech, Croatian and English, and therefore, new insights for NLP community according to the behavior

of the Czech and Croatian word embeddings. Both languages belong to the group of Slavic languages and have only preliminary and basic knowledge insights from word embeddings. In addition, another contribution of this work is certainly new data sets for the Croatian/Czech languages, which are publicly available from: <https://github.com/Svobikl/>. These are also the first parallel English-Croatian/Czech word embeddings datasets.

As the results showed, the Czech/Croatian models do not achieve such good results as for English. Following this statement, we would like to point out that future research should be focused on model improvements for Slavic languages. The difference in English and Slavic language morphology is huge. Compared to the Czech/Croatian language, English language morphology is considerably poorer. Czech/Croatian is a highly inflected language with mostly free word ordering in sentence structure, unlike English, which is inflectional language and has a strict word ordering in a sentence. These differences are reflected in the results of embeddings modeling. Models give good approximations to English, they are better tailored to the English language morphology and better match the structure of such a language.

In future research, it would be worthwhile to explore, which Slavic languages specificities could be advisable to incorporate into models, in order to achieve better modeling of complex morphological structures. On the other hand, corpora preprocessing which simplifies morphological variations, such as stemming or lemmatization procedures, could also have an effect on word embeddings and should be one of the future research directions.

One of the possible directions to achieve better performance is presented in Chapter 7.

5 Semantic Textual Similarity

In [Brychcín and Svoboda, 2016] we present our UWB[1] system for Semantic Textual Similarity (STS). Given two sentences, the system estimates the degree of their semantic similarity. In the monolingual task, our system achieve mean Pearson correlation 75.7% compared with human annotators. Our system was ranked second among 113 submitted systems. In the cross-lingual task, our system has correlation of 86.3% and is ranked first among 26 systems. It shows how well simple Tree LSTM neural network architecture and other syntax, semantic and linguistic features can perform together and represent a meaning of sentence. The system was compared with complex state-of-the-art algorithms for the meaning representation. We also experimented with Paragraphs vector models and linear combination of word vectors (CBOW model) representing the sentence.

Clustering of word vectors and Paragraphs vector models showed significant improvement in sentiment analysis at SemEval2016 competition in [Hercig et al., 2016b] and also in recent work targeted on Czech [Hercig et al., 2016a]. Neural network based Word Embedding models has helped the previous model originally developed for SemEval2014 competition [Brychcín et al., 2014] to get into first position on several tasks during the competition of the year 2016.

So far, most of the STS research has been devoted to English. In [Svoboda and Brychcín, 2018a] we present the first Czech dataset for STS. The Corpus contains 1425 manually annotated pairs. Czech is highly inflected language and is considered challenging for many NLP tasks and STS is one of the core NLP disciplines. The dataset is publicly available for the research community.
We adapt our UWB system (originally for English) and experiment with new Czech dataset. Our UWB system achieves very promising results and can serve as a strong baseline for future research.

[1]University of West Bohemia

The structure of this Chapter is following. Section 5.1 puts our work into the context of the state of the art and introduces the SemEval competition. In Section 5.2 we deal with Semantic Textual Similarity task on English language, respective Section 5.3 for Czech. We define our model features in Sections 5.2.1 and 5.2.2. The experimental results presented and discussed in Sections 5.2.5 and 5.2.6, respective Sections 5.3.4 and 5.3.5 for Czech. We conclude in Section 5.4.

5.1 Introduction

Semantic Textual Similarity (STS) is one of the core disciplines in NLP. Assume, we have two textual pairs (word phrases, sentences, paragraphs, or full documents), the goal is to estimate the degree of their semantic similarity.

STS systems are usually compared with the manually annotated data. In the case of SemEval the data consist of pairs of sentences with a score between 0 and 5 (higher number means higher semantic similarity). For example, English pair

```
Two dogs play in the grass.
Two dogs playing in the snow.
```

has a score 2.8, i.e. the sentences are not equivalent, but share some information.

This year, SemEval's STS is extended with the Spanish-English cross-lingual subtask, where e.g. the pair

```
Tuve el mismo problema que tú.
I had the same problem.
```

has a score 4.8, which means nearly equivalent.

Each year STS is one of the most popular tasks at SemEval competition. The best STS system at SemEval 2012 [Bär et al., 2012] used lexical similarity and Explicit Semantic Analysis (ESA) [Gabrilovich and Markovitch, 2007]. In SemEval 2013, the best model [Han et al., 2013] used semantic models such

as Latent Semantic Analysis (LSA) [Deerwester et al., 1990], external information sources (WordNet) and n-gram matching techniques. For SemEval 2014 and 2015 the best system comes from [Sultan et al., 2014a,b, 2015]. They introduced a new algorithm, which aligns the words between two sentences. Overview of systems participating in previous SemEval competitions can be found in [Agirre et al., 2012, 2013, 2014, 2015].

The best performing systems from previous years are based on various architectures benefiting from lexical, syntactic, and semantic information. In [Brychcín and Svoboda, 2016] we try to use the best techniques presented during last years, enhance them, and combine into a single model. Later, in [Svoboda and Brychcín, 2018a] we present the first Czech dataset for STS and adapt our model to this language as well.

5.2 Semantic Textual Similarity with English

This section describes various techniques for estimating the text similarity on English language and later bring our novel approach to do so.

5.2.1 Lexical and Syntactic Similarity

This section presents the techniques exploiting lexical and syntactic information in the text. Some of them have been successfully used by [Bär et al., 2012]. Many of the following techniques benefit from the weighing of words in a sentence using *Term Frequency – Inverse Document Frequency* (TF-IDF) [Manning et al., 1999].

- **Lemma n-gram overlaps:** We compare word n-grams in both sentences using *Jaccard Similarity Coefficient* (JSC) [Manning et al., 1999]. We do it separately for different orders $n \in \{1, 2, 3, 4\}$. *Containment Coefficient* [Broder, 1997] is used for orders $n \in \{1, 2\}$. We extend original metrics by weighing of n-grams. We define this weight as a sum of *IDF* values of words in n-gram. N-gram match is not counted as one but as the weight of this n-gram. According to our experiments, this weighing significantly improves performance.

  We also use information about the length of *Longest Common Subsequence* compared to the length of the sentences.

- **POS n-gram overlaps:** In similar way as for lemmas, we calculate *Jaccard Similarity Coefficient* and *Containment Coefficient* for n-grams of part-of-speech (POS) tags. Again, we use n-gram weighing and $n \in \{1, 2, 3, 4\}$. These features exploit syntactic similarity of the sentences.

- **Character n-gram overlaps:** Similarly to lemma or POS n-grams, we use *Jaccard Similarity Coefficient* and *Containment Coefficient* for comparing common substrings in both sentences. Here the *IDF* weights are computed on character n-gram level. We use n-gram weighing and $n \in \{2, 3, 4, 5\}$.

  We enrich these features also by *Greedy String Tiling* [Wise, 1996] allowing to deal with reordered text parts and by *Longest Common Substring* (LCS) measuring the ratio between LCS and length of the sentences.

- **TF-IDF:** For each word in a sentence we calculate *TF-IDF*. Given the word vocabulary V, the sentence is represented as a vector of dimension $|V|$ with *TF-IDF* values of words present in the sentence. The similarity between two sentences is expressed as cosine similarity between corresponding *TF-IDF* vectors.

5.2.2 Semantic similarity

In this section we describe in detail the techniques that we use in our STS model. These techniques are more semantically oriented and are based on the *Distributional Hypothesis* (see Chapter 2).

- **Semantic composition:** This approach is based on *Frege's principle of compositionality*, which states that the meaning of a complex expression is determined as a composition of its parts, i.e. words. To represent the meaning of a sentence we use simple linear combination of word vectors, where weights are represented by the *TF-IDF* values of appropriate words. We use state-of-the-art word embedding methods, namely Continuous Bag of Words (CBOW) [Mikolov et al., 2013a] and Global Vectors (GloVe) [Pennington et al., 2014]. We use cosine similarity to compare vectors.

- **Paragraph2Vec:** Paragraph vectors are described in Section 3.8.6. The paragraph token acts as a memory that remembers what information is missing from the current context. We use cosine similarity for comparing two paragraph vectors.

- **Tree LSTM:** is described in more details in Section 3.8.7. We use tree-structured representation of LSTM presented in [Tai et al., 2015a]. Tree model represents the sentence structure. RNN processes input sentences of variable length via recursive application of a transition function on a hidden state vector h_t. For each sentence pair it creates sentence representations h_L and h_R using Tree-LSTM model. Given these representations, model predicts the similarity score using a neural network considering distance and angle between vectors.

- **Word alignment:** Method presented in [Sultan et al., 2014a,b, 2015] has been very successful in last years. Given two sentences we want to compare, this method finds and aligns the words that have similar meaning and similar role in these sentences.

  Unlike the original method, we assume that not all word alignments have the same importance for the meaning of the sentences. The weight of a set of words $\boldsymbol{A}$ is a sum of word's *IDF* values $\omega(\boldsymbol{A}) = \sum_{w \in \boldsymbol{A}} \text{IDF}\,(w)$, where w is a word. Then the sentence similarity is given by

$$\text{sim}\,(\boldsymbol{S}_1, \boldsymbol{S}_2) = \frac{\omega(\boldsymbol{A}_1) + \omega(\boldsymbol{A}_2)}{\omega(\boldsymbol{S}_1) + \omega(\boldsymbol{S}_2)}, \tag{5.1}$$

  where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are input sentences (represented as sets of words). $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ denote the sets of aligned words for $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, respectively. The weighting of alignments improves our results significantly.

5.2.3 Similarity Combination

The combination of STS techniques is in fact a regression problem where the goal is to find the mapping from input space $\boldsymbol{x}_i \in \mathbb{R}^d$ of d-dimensional real-valued vectors (each value $x_{i,a}$, where $1 \leq a \leq d$ represents the single STS technique) to an output space $y_i \in \mathbb{R}$ of real-valued targets (desired semantic similarity). These mapping are learned from the training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ of size N. There exist a lot of regression methods. We experiment with several of them:

- **Linear Regression:** Linear Regression (LR) is probably the simplest regression method. It is defined as $y_i = \boldsymbol{\lambda} \boldsymbol{x}_i$, where $\boldsymbol{\lambda}$ is a vector of weights that can be estimated for example by the *least squares method*.
- **Gaussian processes regression:** Gaussian process regression (GPR) is nonparametric kernel-based probabilistic model for non-linear regression [Rasmussen and Williams, 2005].
- **SVM Regression:** We use Support Vector Machines (SVM) for regression with the radial basis functions (RBF) as a kernel. We use improved Sequential Minimal Optimization (SMO) algorithm for parameter estimation introduced in [Shevade et al., 2000].
- **Decision Trees Regression:** The output of the Decision Trees Regression (DTR) [Breiman et al., 1984] is predicted by the sequence of decisions organized in a tree.
- **Perceptron Regression:** Multilayer Perceptron (MLP) is feed-forward artificial neural network that uses back-propagation to classify instances.

5.2.4 System Description

This section describes the settings of our final STS system. For monolingual STS task we submitted two runs. The first is based on supervised learning and the second is an unsupervised system:

- **UWB sup:** Supervised system based on SVM regression with RBF kernel. We use all techniques described in 5.3.2 as features for regression. During the regression we also use this simple trick: we create a set of additional features represented as a product of each pair of features $x_{i,a} \times x_{i,b}$ for $a \neq b$. We do so to better model the dependencies between single features. Together, we have 301 STS features. The system is trained on all SemEval datasets from prior years (see Table 5.1).
- **UWB unsup:** Unsupervised system based only on weighted word alignment (Section 5.2.2).

We handled the cross-lingual STS task with Spanish-English bilingual sentence pairs in two steps. Firstly, we translated Spanish sentences to English

Corpora	Pairs
SemEval 2012 Train	2,234
SemEval 2012 Test	3,108
SemEval 2013 Test	1,500
SemEval 2014 Test	3,750
SemEval 2015 Test	3,000

Table 5.1: STS gold data from prior years.

via *Google translator*. The English sentences were left untouched. Secondly, we used the same STS systems as for monolingual task.

For preprocessing pipeline we used the Stanford CoreNLP library [Manning et al., 2014], i.e. for tokenization, lemmatization and POS tagging. Most of our STS techniques (apart from word alignment and POS n-gram overlaps) work with lemmas instead of word forms (this leads to slightly better performance). Some of our STS techniques are based on unsupervised learning and thus they need large unannotated corpora to train. We trained Paragraph2Vec, GloVe and CBOW models on *One billion word benchmark* presented in [Chelba et al., 2014]. Dimension of vectors for all these models was set to 300. TF-IDF values were also estimated on this corpus.

All regression methods mentioned in Section 5.2.3 are implemented in WEKA [Hall et al., 2009].

5.2.5 Results

This section presents the results of our systems for both English monolingual and Spanish-English cross-lingual STS task of SemEval 2016. In addition we present detailed results on the test data from SemEval 2015. As an evaluation measure we use *Pearson correlation* between system output and human annotations.

5.2.6 Discussion

In the tables we present the correlation for each individual test set. Column *Mean* represents the weighted sum of all correlations, where the weights are

Model \Corpora	**Answers-forums**	**Answers-students**	**Belief**	**Headlines**	**Images**	**Mean**
Winner of SemEval 2015	0.7390	0.7725	0.7491	0.8250	0.8644	0.8015
Linear regression – all lexical	0.7053	0.7656	0.7190	0.7887	0.8246	0.7728
Linear regression – all syntactic	0.3089	0.3165	0.4570	0.2900	0.1862	0.2939
Tf-idf	0.5629	0.6043	0.6762	0.6603	0.7530	0.6593
Tree LSTM	0.4181	0.5490	0.5863	0.7324	0.8168	0.6501
Paragraph2Vec	0.5228	0.7017	0.6643	0.6562	0.7385	0.6725
CBOW composition	0.6216	0.6846	0.7258	0.6927	0.7831	0.7085
GloVe composition	0.5820	0.6311	0.7164	0.6969	0.7972	0.6936
Weighted word alignment	0.7171	**0.7752**	0.7632	0.8179	0.8525	0.7964
Linear regression	**0.7411**	0.7589	0.7739	0.8193	0.8568	0.7982
Gaussian processes regression	0.7363	0.7701	**0.7846**	0.8393	0.8749	0.8112
Decision trees regression	0.6700	0.6991	0.7281	0.7792	0.8206	0.7495
Perceptron regression	0.7060	0.7481	0.7467	0.8093	0.8594	0.7858
SVM regression	0.7375	0.7678	**0.7846**	**0.8398**	**0.8776**	**0.8116**

Table 5.2: Pearson correlations on SemEval 2015 evaluation data and comparison with the best performing system in this year.

Model \Corpora	**Answer-answer**	**Headlines**	**Plagiarism**	**Postediting**	**Question-question**	**Mean**
UWB sup	0.6215	**0.8189**	0.8236	**0.8209**	**0.7020**	**0.7573**
UWB unsup	**0.6444**	0.7935	**0.8274**	0.8121	0.5338	0.7262

Table 5.3: Pearson correlations on monolingual STS task of SemEval 2016.

	News	**Multi-Source**	**Mean**	**RR**	**TR**
UWB sup	0.9062	**0.8190**	**0.8631**	1	1
UWB unsup	**0.9124**	0.8082	0.8609	2	1

Table 5.4: Pearson correlations on cross-lingual STS task of SemEval 2016. *RR* denote the run (system) ranking and *TR* denote our team ranking.

given by the ratio of data set length compared to the full length of all datasets together. The mean value of Pearson correlations is also used as the main evaluation measure for ranking the system submissions.

In the Table 5.2 we show the results of combined features for the test data from 2015. We trained our systems on SemEval STS data from years 2012–2014. We provide comparison of individual STS techniques as well as of different types of regressions. Clearly, the SVM regression and Gaussian processes regression perform best and with our feature set it is 1% better than the winning system of SemEval 2015. The best performing single technique is indisputably the weighed word alignment correlated by 79.6% with gold data. Note that without weighing, we achieved only 74.2% on this data. The original result from authors of this approach was, however, 79.2%. This is probably caused by some inaccuracies in our implementation. Anyway, the weighing improves the correlation even if we compare it to the original results. Note that for estimating regression parameters we use the data from all years apart from 2015 (see Table 5.1).

The results for the monolingual STS task of SemEval 2016 are shown in Table 5.3. We can see that our supervised system (SVM regression) performs approximately 3% better than the unsupervised one (weighed word alignment). On the data from SemEval 2015 this difference was not so significant (approximately 1.5%).

Finally, the results for cross-lingual STS task of SemEval 2016 are shown in Table 5.4. We achieved very high correlations. We expected much lower correlation through the fact that we use the machine translation via Google translator causing certainly some inaccuracies (at least in the syntax of the sentence). On the other hand, it proves that our model efficiently generalizes the learned patterns. Here, there is almost no difference in performance between supervised and unsupervised version of submitted systems. Our submitted runs finished first and second among 26 competing systems.

5.3 Semantic Textual Similarity with Czech

For Czech there are corpora for measuring the individual words embeddings properties, such as: RG-65 [Krcmár et al., 2011], WS-353 [Cinková, 2016] and now Czech Word analogy corpora – see Chapter 4, but no corpora for measuring sentence similarity.

We introduce new Czech dataset for STS task. The corpora have been divided into 925 training and 500 testing pairs (see Table 5.5) translated to Czech by four native speakers from previous SemEval years. In SemEval competition the data consist of pairs of sentences with a score between 0 and 5 (higher number means higher semantic similarity). For example, Czech pair:

`Černobílý pes se dívá do kamery`[2]
`Černobílý býk se dívá do kamery`[3]

has a score of 2, sharing information about camera, but it is about different animal. We kept annotated similarities unchanged.

Corpora	**Pairs**
SemEval 2014-15 Images CZ – Train	550
SemEval 2013-15 Headlines CZ – Train	375
SemEval 2014-15 Images CZ – Test	300
SemEval 2013-15 Headlines CZ – Test	200

Table 5.5: Corpora with STS gold sentences in Czech.

5.3.1 Data preprocessing

To deal with Czech rich morphology, we use lemmatization [Straková et al., 2014] and stemming [Brychcín and Konopík, 2015, Dolamic and Savoy, 2009] to preprocess the training data. Stemming and lemmatization are two related fields and are among the basic preprocessing techniques in NLP. Both methods are often used for similar purposes: to reduce the inflectional word forms in a text. Stemming usually refers to a crude heuristic process that removes the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Product of lemmatization is a lemma which is a valid linguistic unit (the base or dictionary form of a word).

[2]A black and white dog looking at the camera
[3]The black and white bull is looking at the camera

5.3.2 System Description

For estimating the text similarity on Czech we used some of techniques presented in Section 5.3.2.

We summarize these basic features as follows:

- **IDF weighted lemma n-gram overlapping**, measured with *Jaccard Similarity Coefficient* (JSC).
- **IDF weighted POS n-gram overlapping**, measured with JSC.
- **Character n-gram overlapping**, measured with JSC.
- **TF-IDF** as standalone feature.
- **String features**, such as longest common subsequence, longest common substring where similarity is computed as fraction of longest commong subsequence/substring divided by the length of both sentences.

From semantically oriented methods we use state-of-the-art word embedding methods: CBOW and SkipGram [Mikolov et al., 2013a] and compare its semantic composition properties with recently published [Bojanowski et al., 2017] method that enriches word vectors with subword information. This method promises significant improvement of word embeddings quality especially for languages with rich word morphology.

We train all three above mentioned methods on Czech Wikipedia and provide experiments on datasets for word similarity (WS-353 [Cinková, 2016] and RG-65 [Krcmár et al., 2011]) and word analogy [Svoboda and Brychcín, 2016]. Results are shown in Table 5.6.

	Word similarity		**Word**
Model	**WS-353**	**RG-65**	**analogy**
FastText-SkipGram 300d wiki	67.04	67.07	71.72
FastText-CBOW 300d wiki	40.46	58.35	73.23
CBOW 300d wiki	54.31	47.03	58.69
SkipGram 300d wiki	65.93	68.09	53.74

Table 5.6: Word similarity and word analogy results on Czech Wikipedia.

5.3.3 Czech STS model

The combination of STS techniques mentioned in Sections 5.2.1 and 5.2.2 is a regression problem that is already described in Section 5.2.3. We experiment similarly to English with three regression methods:

- Linear Regression (LR),
- Gaussian Process (GP),
- Support Vector Machines (SVM) with Sequential Minimal Optimization (SMO) algorithm [Platt, 1998].

The system was trained on 925 pairs and further tested on 500 pairs (see Table 5.5).

We use algorithms for the meaning representation in the same manner as we have used for English at SemEval 2016 (see Section 5.2). Methods benefit from various sources of information, such as lexical, syntactic, and semantic.

This section describes all measured settings and their reasons. The former is a traditional STS task with paired monolingual sentences originally translated from English data sources to Czech followed by cross-lingual test. Gold data were evaluated:

- **Lexical, syntactic and semantic features:** We evaluated each feature from three categories individually in the same manner as with English to see influence of particular feature (see Table 5.9).
- **Preprocessing tests:** Most of our STS techniques (apart from word alignment and POS n-gram overlaps) work with lemmas instead of word forms (this leads to better performance). We tested all features with three techniques of representing individual tokens in sentence – word, stemming and lemma (see Table 5.10).
- **Crosslingual test:** Cross-lingual STS involves assessing paired English and Czech sentences. Cross-lingual STS measure enables an alternative way to comparing text. Due to lack of the supervised training data in the particular language, cross-lingual task is getting still higher attention during last years.

We handled with the cross-lingual STS task with Czech-English bilingual sentence pairs in two steps. Firstly, we translated original Czech sentences to English via *Google translator*. We did not use original-matching EN sentences, we did not want to involve potential manual processing of translation for cross-lingual evaluation and that was also in most cases the way, how cross-lingual task was evaluated on SemEval2016. However, the situation is changing with new bilingual word embeddings methods coming up in recent years (see our research presented in Section 4.4). The Czech sentences were left untouched. Secondly, we used the same STS system as for monolingual task. Because we have much bigger training set for English sentences, we wanted to see if such data-set will help us in performance on Czech, results can be seen in Table 5.7.

Some of our STS techniques are based on unsupervised learning and thus they need large unannotated corpora to train. We trained CBOW, Skipgram and FastText models on Czech Wikipedia. Wikipedia dump comes from 05/10/2016 with 847 million tokens, resulting models has vocabulary of size 773,952. This dump has been cleaned from any Wiki Markup tags and from HTML tags. Dimension of vectors for all these models was set to 300.

Model \Corpora	**Headlines**	**Images**
Monolingual test	0.7999	**0.7887**
Czech-English crossling. (850 pairs)	0.8060	0.7583
Czech-English crossling. (3000 pairs)	**0.8198**	0.7649

Table 5.7: Comparison of Pearson correlations on monolingual STS task versus crosslingual STS task with automatic translation to English. Cross-lingual model is trained on data from SemEval 2014 and 2015.

5.3.4 Results

Based on the learning curve (see Figure 5.1), the system needs at least 170 pairs to set weights of individual features, therefore we can state that our system has reasonable amount of training data for learning – this theory is also supported by larger amount of training data thanks to cross-lingual test (see Table 5.7).

We have achieved the best score of 78.87% on short *Images labels* with simple Linear regression. With such short sentences we will not benefit from

a larger dataset, as can be seen in Table 5.7 from our evaluation of cross-lingual test with the much larger dataset base (3000 pairs) that we have for English. We benefit from larger corpora on longer *Headlines* sentences, where we have achieved a score of 81.98%.

Model \Corpora	**Headlines**	**Images**
Our best at SemEval 2016 (EN)	0.8398	0.8776
Linear regression	0.7918	**0.7887**
Gaussian processes regression	0.7986	0.7829
SVM regression	**0.7999**	0.7856

Table 5.8: Pearson correlations on Czech evaluation data and comparison with the second best system from SemEval 2016 on English data.

Model \Corpora	**Images**	**Headlines**
Longest Common Subsequence	0.6586	0.6993
Longest Common Substring	0.4998	0.5886
Greedy String Tiling	0.7005	**0.7983**
all string features	**0.7379**	0.7932
IDF weighted word n-grams	0.5979	0.6432
IDF weighted character n-grams	0.6885	0.7869
POS n-grams	0.5331	0.5618
TF-IDF	0.5785	0.5892
CBOW composition	0.6774	0.6355
SkipGram composition	0.6299	0.6785
Char-SkipGram composition	0.5966	0.6396
Char-CBOW composition	0.4958	0.5102

Table 5.9: LR test of individual features, word base is lemma.

5.3.5 Discussion

Interesting results can be seen in Table 5.9 for standalone vector composition. The standard Skipgram model seems to be more suitable to carry the meaning of a sentence as a simple linear combination of word vectors, despite the fact that it has lower score on similarity measurements of individual words (see Table 5.6).

Czech is a language with rich morphology, as it can be seen from Table 5.9, so string features plays important role, especially *Greedy String Tailing*. The more matches found in words endings, the higher success of reasoning about two sentences. Results of testing lemma versus stemming techniques give similar score. Of course without preprocessing we get slightly lower score, this can be seen on n-gram features, where stemming is performing the best (see Table 5.10). When the model is covered by syntactic features, the situation for lemma and stemming techniques is nearly equal.

Together with presented Czech corpora we have original matching sentences in English, so our corpora can be used for new STS cross-lingual task without manually translating the sentences to English and can be evaluated directly with bilingual word embeddings methods [Vulić and Moens, 2015, Gouws and Søgaard, 2015] in future. These methods are getting popular in recent years and take a key part in the current SemEval competitions.

[Brychcín and Svoboda, 2016] showed that use of syntactic parse tree and training with tree-based LSTM [Tai et al., 2015b] does not provide benefit on English where classic bag-of-words semantic approaches does better job, however this situation might change on highly inflected languages as Czech and might be worth testing.

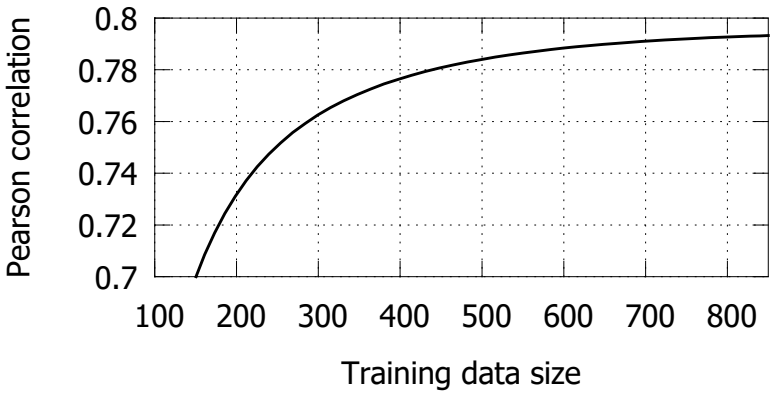


Figure 5.1: Pearson correlation achieved by linear regression with different training data size (ranging between 50 and 850 pairs).

Model features \Corpora	**Correlation**
ngram features (word)	0.6140
ngram features (lemma)	0.6959
ngram features (stem)	0.7319
ngram + string features (word)	0.7732
ngram + string features (lemma)	0.7897
ngram + string features (stem)	0.7829
all previous + syntactic (word)	0.7704
all previous + syntactic (lemma)	0.7860
all previous + syntactic (stem)	0.7865
all + CBOW composition (word)	0.7796
all + SkipGram composition (word)	0.7814
all + Char-SkipGram composition (word)	0.7774
all + CBOW composition (lemma)	0.7917
all + SkipGram composition (lemma)	0.7924
all + CBOW composition (stem)	0.7910
all + SkipGram composition (stem)	**0.7939**

Table 5.10: Pearson correlations on Czech evaluation data and comparison with the second best system from SemEval 2016 on English data. Test made with linear regression.

5.4 Conclusion

In this chapter we described our UWB system participating in SemEval 2016 competition in the task of Semantic Textual Similarity. We participated on both monolingual and cross-lingual parts of competition.

We introduced a new dataset for semantic textual similarity of Czech sentences. We created strong baseline based on state-of-the-art approaches. Our baseline on Czech achieved mean Pearson correlation of 80% (compared with 88% achieved on English data).

The Czech STS dataset with its original matching sentences in English is available for free at following link: `<https://github.com/Svobikl/sts-czech.git>`.

6 Aspect-Based Sentiment Analysis

In [Hercig et al., 2016b] we build a system for ABSA using distibutional semantic models on English, further in [Hercig et al., 2016a] we examine the effectiveness of several unsupervised methods for latent semantics discovery as features for ABSA on Czech language. We use the shared task definition from SemEval 2014.

In our experiments we use labeled and unlabeled corpora within the restaurants domain for two languages: Czech and English. We show that our models improve the ABSA performance and prove that our approach is worth exploring. Moreover, we achieve new state-of-the-art results for Czech.

Another important contribution of our work is that 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.

The structure of this Chapter is as follows: Section 6.1 puts our work into the context of the state of the art and introduces the ABSA task. Czech ABSA corpora are defined in Section 6.1.2, our model features in Sections 6.2. The experimental results are presented and discussed in Section 6.3. We conclude in Section 6.4.1.

6.1 Introduction

The majority of recent approaches to sentiment analysis try to detect the overall polarity of a sentence (or a document) regardless of the target entities (e.g. restaurants, laptops) and their aspects (e.g. food, price, battery, screen). In contrast, the current approach, ABSA identifies the aspects of a given target entity and estimates the sentiment polarity for each mentioned aspect.

In the context of the ABSA task, the bottleneck is the size of the annotated data, which should be considerably larger in order to simulate real world applications. Web content such as blogs, forums, reviews etc. present a large amount of easily accessible domain-relevant unlabeled data which we could use to add specific domain knowledge essential for improving the state of the art of sentiment analysis. Thus we try to demonstrate the usefulness of these data.

There have been several attempts in Czech as well [Veselovská et al., 2012, Brychcín and Habernal, 2013, Habernal et al., 2013], but all were focused on the global (sentence or document level) sentiment.

The first attempt at aspect-based sentiment analysis in Czech was presented in [Brychcín et al., 2014]. This work provides an annotated corpus of 1244 sentences from the restaurant reviews domain and a baseline model achieving 68.65% F-measure in aspect term extraction, 74.02% F-measure on aspect category extraction, 66.27% accuracy in aspect term polarity classification, and 66.61% accuracy in aspect category polarity classification. The work in [Tamchyna et al., 2015] creates a dataset in the domain of IT product reviews. This dataset contains 200 annotated sentences and 2000 short segments, both annotated with sentiment and marked aspect terms (targets) without any categorization and sentiment toward the marked targets.

The current state of the art of aspect-based sentiment analysis methods for English was presented at the latest SemEval ABSA tasks namely the SemEval 2015 – 2016 [Pontiki et al., 2015, 2016]. The detailed description of each system is beyond the scope of this thesis.

Our main goal is twofold: to show how unsupervised methods can improve an ABSA system in different languages; and the creation of sufficiently large corpora for the ABSA task in Czech.

6.1.1 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.

Subtask 1: Aspect term extraction

The aspect term extraction is based on experiences in NER [Konkol and Konopík, 2013, Konkol et al., 2015b]. 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 Conditional Random Fields (CRF) [Lafferty et al., 2002], 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. We omit the description of these experiments, as they are outside the scope of this thesis.

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 very 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*.

6.1.2 ABSA Corpora

The methods described in Section 6.3.1 require large unlabeled data in order to be trained. In [Hercig et al., 2016a] we used two types of corpora, labeled and unlabeled for both Czech and English. The properties of these corpora are shown in Table 6.1.

Labeled corpora for both languages are required to train the classifiers (see Section 6.2). 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 6.1).

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 6.1: 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 82.91% (aspect term extraction), 88.02% (aspect category extraction), 85.71% (aspect term polarity) and 88.44% (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 is taken from `<http://opentable.com>`.

6.2 ABSA System Description

We use and extend the systems created by Brychcín et al. [2014]. We implemented four separate systems – one for each subtask of ABSA. 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., 2016b].

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 6.3.1. 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 6.3.1. 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).

6.3 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 *+*. E.g. *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.

6.3.1 Unsupervised Model Settings

All unsupervised models were trained on the unlabeled corpora described in Section 6.1.2.

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 [Kachites McCallum, 2002] software package.

The detailed settings of all these methods are shown in Table 6.2.

	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 6.2: Model settings

CLUTO software package [Karypis, 2002] is used for words clustering with the k-means algorithm and cosine similarity metric. All vector space models in this chapter cluster the word vectors into four different numbers of clusters: 100, 500, 1000, and 5000. For stemming, we use the implementation of HPS [Brychcín and Konopík, 2015][4] that is the state-of-the-art unsupervised stemmer.

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

6.4 Results

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 6.3: Aspect term, category extraction (TE, CE) and and polarity (TP, CP) of 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 6.4: Aspect term, category extraction (TE, CE) and and polarity (TP, CP) of models combinations on Czech dataset

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 detailed results of all models tested separately are in [Hercig et al., 2016a].

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 Czech (see Table 6.4).

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 6.3 for English and Table 6.4 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 6.3 and 6.4 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.

6.4.1 Conclusion

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. To the best of our knowledge, these results are now the state-of-the-art 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 our model 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.

7 Word Embeddings and Global Information

In this chapter we evaluate our new approach based on the *Continuous Bag-of-Words* and *Skip-gram* models enriched with global context information on highly inflected Czech language and compare it with English results. As a source of information we use Wikipedia, where articles are organized in a hierarchy of categories. These categories provide useful topical information about each article.

Both models are evaluated on standard word similarity and word analogy datasets. Proposed models outperform other word representation methods when similar size of training data is used. The models provide similar performance to methods trained on much larger datasets.

The structure of this chapter is following. Section 7.2 puts our work into the context of the state of the art. In Section 7.3 we review Word2Vec models on which our work is based. We define our model in Section 7.5 and 7.4. The experimental results presented in Section 7.7. We conclude in Section 7.9 and offer some directions for future work.

7.1 Introduction

The principle known as the *Distributional Hypothesis* has been presented in Chapter 2; the research presented in this Chapter directly refers to it.

7.1.1 Local Versus Global Context

Different types of context induce different kinds of semantic spaces. [Riordan and Jones, 2011] and [McNamara, 2011] distinguish *context-word* and *context-region* pproaches to the meaning extraction. In this chapter we use the

notation *local context* and *global context*, respectively. Global-context *DSMs* are usually based on the *bag-of-words hypothesis*, assuming that the words are semantically similar if they occur in similar articles and the order in which they occur in articles has no meaning. These models are able to register long-range dependencies among words and are more topically oriented. In contrast, local-context *DSMs* collect short contexts around the word using moving window to induce the meaning. Resulting word representations are usually less topical and exhibit more functional similarity (they are often more syntactically oriented).

To create a proper *DSM* a large textual corpus is usually required. Very often Wikipedia is used for training, because it is currently the largest knowledge repository on the Web and is available in dozens of languages. Most current *DSMs* learn the meaning representation merely from the word distributions and do not incorporate any of the metadata which Wikipedia contains.

7.1.2 Our Model Using Global Information

In this work we combine both the local and the global context to improve the word meaning representation. We use local-context *DSMs Continuous Bag-of-Words* (CBOW) and Skip-Gram models [Mikolov et al., 2013a], the original tool is often denoted as *Word2Vec*. We incorporate Wikipedia categories as a global context.

We train our models on English and Czech Wikipedia. We evaluate it on standard word similarity and word analogy datasets. Proposed models significantly outperform other word representation methods when similar training data size is used and provide similar performance compared with methods trained on much larger datasets.

7.2 Related Work

In the past decades, simple frequency-based methods for deriving word meaning from raw text were popular, e.g. Hyperspace Analogue to Language [Lund and Burgess, 1996] or paper Correlated Occurrence Analogue to Lexical Semantics [Rohde et al., 2004] as a representatives of local-context DSMs and

Latent Semantic Analysis [Landauer et al., 1998]. Or Explicit Semantic Analysis [Gabrilovich and Markovitch, 2009] as a representatives of global-context DSMs. All these methods record word/context co-occurrence statistics into the one large matrix defining the semantic space.

Later on, these approaches have evolved in more sophisticated models. [Mikolov et al., 2013a] revealed neural network based models *CBOW* and *Skip-gram* that we are going to use as our baseline to incorporate Global context. His simple single-layer architecture is based on the inner product between two word vectors (detailed description is in Section 7.3). [Pennington et al., 2014] introduced Global Vectors, the log-bilinear model that uses weighted least squares regression for estimating word vectors. The main concept of this model is the observation that global ratios of word/word co-occurrence probabilities have the potential for encoding meaning of words.

7.2.1 Local Context with Subword Information

Above mentioned models currently serve as a basis for many researches. [Bojanowski et al., 2017] improved Skip-Gram model by incorporating subword information. Similarly, a recent study [Salle and Villavicencio, 2018] incorporated sub-word information into LexVec [Salle et al., 2016] vectors. Improvement is especially evident for languages with rich morphology. [Levy and Goldberg, 2014] used syntactic contexts automatically produced by dependency parse-trees to derive the word meaning. Their word representations are less topical and exhibit more functional similarity (they are more syntactically oriented).

[Huang et al., 2012] presented a new neural network architecture which learns word embeddings that capture the semantics of words by incorporating both local and global document context. It accounts for homonymy and polysemy by learning multiple embeddings per word. Authors introduce a new dataset with human judgments on pairs of words in sentential context, and evaluate their model on it. Their approach is focusing on polysemous words and generally does not perform as well as Skip-Gram or CBOW.

7.3 Word2Vec

This section describes the Word2Vec package which includes two neural network model architectures (CBOW and Skip-Gram) that produce distributional representations of words [Mikolov et al., 2013a]. Given the training corpus represented as a set of documents $\boldsymbol{D}$, each document (resp. article) $\boldsymbol{a}_j \in \boldsymbol{D}$ is a sequence of words $\boldsymbol{a}_j = \{w_{j,i}\}_{i=1}^{L_j}$, where L_j denote the length of the article $\boldsymbol{a}_j$. Each word w in the vocabulary $\boldsymbol{W}$ is represented by two different vectors $\mathbf{v}$ and $\mathbf{u}$ depending whether it is used as a context word $\mathbf{v}_w \in \mathbb{R}^d$ or a target word $\mathbf{u}_w \in \mathbb{R}^d$. The task is to estimate these vector representations in a way that optimize the objective functions described below.

We use a training procedure introduced in [Mikolov et al., 2013c] called *negative sampling*. For the word at position i in the article $\boldsymbol{a}_j$ we define the negative log-likelihood

$$E(w, \mathbf{h}) = -\log \sigma(\mathbf{u}_{w_o}^\top \mathbf{h}) - \sum_{w_n \in \boldsymbol{N}} \log \sigma(\mathbf{u}_{w_n}^\top \mathbf{h}), \tag{7.1}$$

where $\boldsymbol{N} = (w_n \sim P(\boldsymbol{W}) | n = 1, ..., K)$ is a set of negative samples (randomly selected words from a noise distribution $P(\boldsymbol{W})$, w_o is the output word, and u_{w_o} is its output vector; $\boldsymbol{h}$ is the output value of the hidden layer:
$h = \frac{1}{C} \sum_{C=1..N} \mathbf{v}_{w_c}$ for CBOW and $h = \mathbf{v}_{w_I}$ in the Skip-gram model;
$\sigma(x) = 1/(1 + exp(-x))$.

Considering articles a_j, in the *CBOW* architecture, the model predicts the current word $w_{j,i}$ from a window of surrounding context words $w_c \in \boldsymbol{C}_{j,i}$. The context is based on bag-of-words hypothesis, so that the order of the words does not influence the prediction. The CBOW model optimizes following objective function:

$$\sum_{\boldsymbol{a}_j \in \boldsymbol{D}} \sum_{i=1}^{L_j} E(w_{j,i}, \frac{1}{|\boldsymbol{C}_{j,i}|} \sum_{w_c \in \boldsymbol{C}_{j,i}} \mathbf{v}_{w_c}). \tag{7.2}$$

According to [Mikolov et al., 2013a], *CBOW* is faster than *Skip-Gram*, but *Skip-Gram* usually performs better for infrequent words.

7.4 Wikipedia Category Structure

Wikipedia is a good source of global information. Overall, Wikipedia comprises more than 40 million articles in 301 different languages. Each article references others that describe particular information in more detail. Wikipedia gives more information about an article that we might not see at the first glance, such as the mentioned links to other articles, or at the end of the article there is a section that describes all categories where current article belongs. The category system of Wikipedia (see fig. 7.1) is organized as an overlapping tree [Shuai et al., 2014] of categories[1] with one main category and a lot of subcategories. Every article contains several categories to which it belongs. Categories are intended to group together pages on similar subjects. Any category may branch into subcategories, and it is possible for a category to be a subcategory of more than one 'parent' category (A is said to be a parent category of B when B is a subcategory of A) [Shuai et al., 2014]. The page editor uses either existing categories, or creates one. Generally the user-defined categories are too vague or may not be otherwise suitable to use in our model as a source of global information. Fortunately, Wikipedia provides 25 main topic classification categories for all Wikipedia pages.

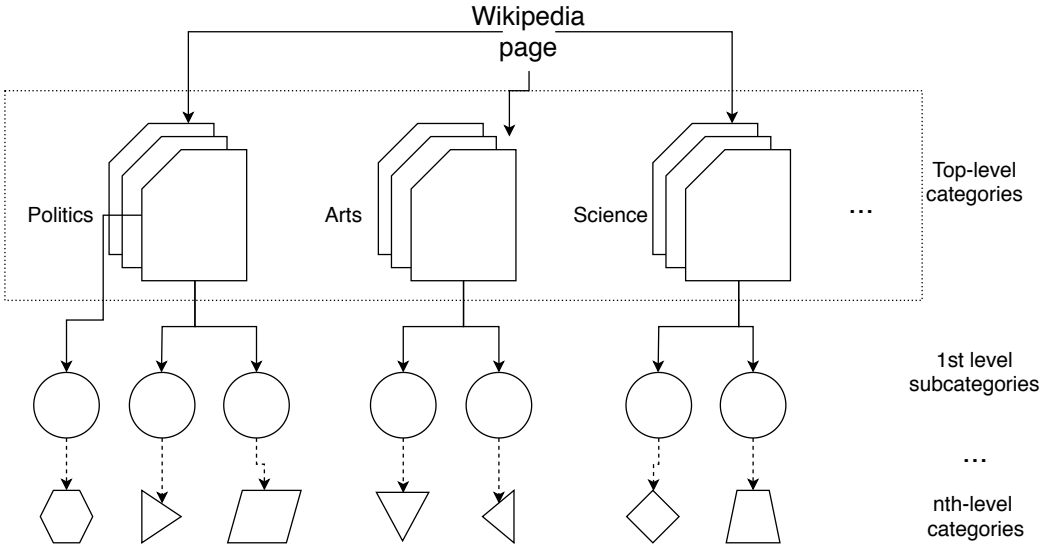


Figure 7.1: Wikipedia category system.

[1]https://en.wikipedia.org/wiki/Portal:Contents/categories

For example the article entitled *Czech Republic* has categories *Central Europe*, *Central European countries*, *Eastern European countries*, *Member states of NATO*, *Member states of EU*, *Slavic countries and territories* and others.
Wikipedia categories provide very useful topical information about each article. In our work we use extracted categories to improve the performance of word embeddings. We denote articles as a_j and categories as x_k.

7.5 Proposed Model

Some authors tried to extract a more concrete meaning using *Frege's principle of compositionality* [Pelletier, 1994], which states that the meaning of a sentence is determined as a composition of words. [Zanzotto et al., 2010] introduced several techniques to combine word vectors into the final vector for a sentence. In [Brychcín and Svoboda, 2016] we experimented with Semantic Textual Similarity, from the tests with words vector composition based on *CBOW* architecture, we can see that this method is a powerful way to carry the meaning of a sentence.

Our model is shown in Figure 7.2. We build up the model based on our previous knowledge and beliefs that global information might improve the performance of word embeddings and further lead to improvements in many NLP subtasks.

Each article $\boldsymbol{a}_j$ in Wikipedia is associated with the set of categories $\boldsymbol{X}_j$. We represent the category $x \in \boldsymbol{X}_j$ as a real-valued vector $\mathbf{m}_x \in \mathbb{R}^d$.

For the *CBOW* model optimize following objective function:

$$\sum_{\boldsymbol{a}_j \in \boldsymbol{D}} \sum_{i=1}^{L_j} E(w_{j,i}, \frac{\sum_{w_c \in \boldsymbol{C}_{j,i}} \mathbf{v}_{w_c} + \sum_{x \in \boldsymbol{X}_j} \mathbf{m}_x}{|\boldsymbol{C}_{j,i}| + |\boldsymbol{X}_j|}) \tag{7.3}$$

For the *Skip-gram* model optimize following objective function:

$$\sum_{\boldsymbol{a}_j \in \boldsymbol{D}} \sum_{i=1}^{L_j} \sum_{w_c \in \boldsymbol{C}_{j,i}} E(w_{j,i}, \mathbf{v}_{w_c} + \sum_{x \in \boldsymbol{X}_j} \mathbf{m}_x) \tag{7.4}$$

Visualization of the modified CBOW architecture is shown in Figure 7.2

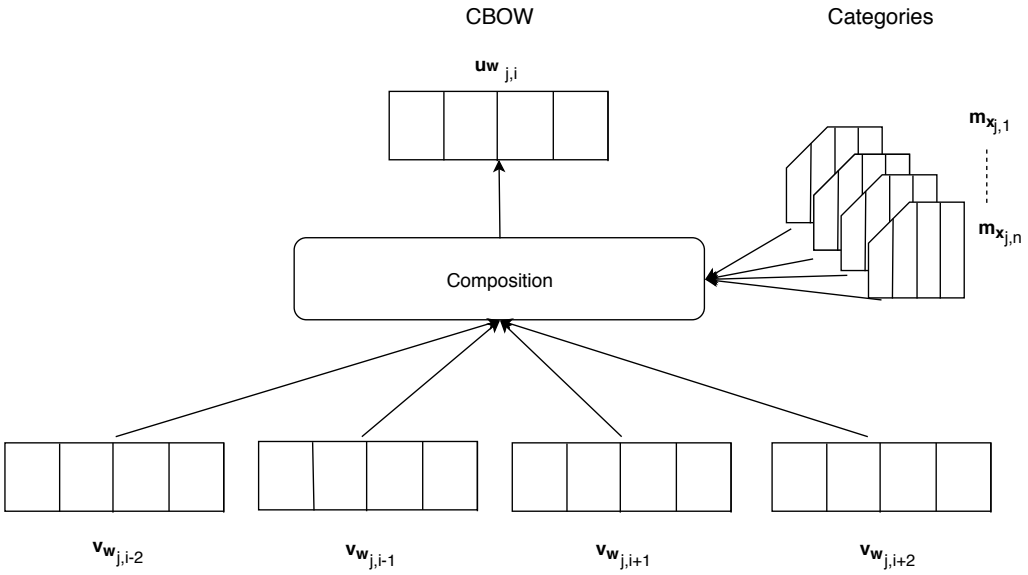


Figure 7.2: Architecture of enriched CBOW model with categories.

Visualization of the modified Skip-gram architecture is shown in Figure 7.3

We tested with *CBOW* and *Skip-gram* architectures enriched with categories that are shown at Figures 7.2 and 7.3. The *CBOW* architecture is generally much faster and easier to train and gives a good performance. The *Skip-gram* architecture takes ten times longer to train, and was unstable during our setup with categories.

7.5.1 Setup 1

Categories are initialized with uniform vector distribution and no training of categories is performed. Only word embeddings are trained. Output of this setup is a model with trained word embeddings. The objective function 7.3

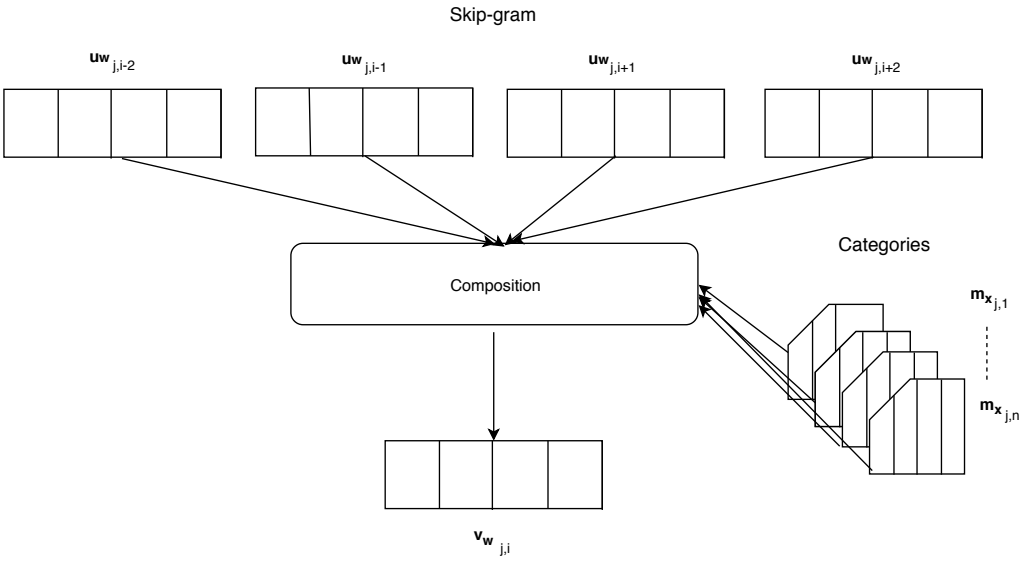


Figure 7.3: Architecture of enriched Skip-gram model with categories.

remains intact, the vectors m_x stays untouched during the complete training. The motivation behind this setup is, that some articles share similar categories. We expect, that if we sum vectors of similar categories and mix them with context word vectors, we end up closer each other in the n-dimensional vector space. We assume that improvement in training of individual words enriched with this information may lead to a better vector representation, especially in describing the words with similar meaning and context.

7.5.2 Setup 2

Many models benefit from the weighing of words in a sentence using *Term Frequency – Inverse Document Frequency* (TF-IDF) [Manning et al., 1999]. Categories are initialized with uniform vector distribution. Vectors representing categories were also not trained, only weighted using *TF-IDF*. In sentences the punctuation, prepositions, conjunctions and others have smaller impact on the overall meaning of sentence. The idea here is that not all categories have equal impact on description of the document. Output of this setup is a model with trained word embeddings.

Adapted objective function is as follows:

$$\sum_{\boldsymbol{a}_j \in \boldsymbol{D}} \sum_{i=1}^{L_j} E(w_{j,i}, \frac{\sum_{w_c \in \boldsymbol{C}_{j,i}} \mathbf{v}_{w_c} + \sum_{x \in \boldsymbol{X}_j} \text{tfidf}(x, d, D) \cdot m_x}{|\boldsymbol{C}_{j,i}| + |\boldsymbol{X}_j|}), \tag{7.5}$$

where $\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$, $f_d(t)$ is frequency of term t in document d. $\mathbf{D}$ is corpus of documents (resp. articles).

7.5.3 Setup 3

The model is initialized with categories uniformly distributed, embeddings for categories are trained during training word embeddings. Motivation of this setup comes from Distributional hypothesis [Harris, 1954] that says: "linguistic items with similar distributions have similar meanings". If we train the categories, we assume they would behave similarly. For example, having an article with categories 'vehicles' and 'transportation', those categories will likely have similar distribution of articles and they will slowly come closer to each other in vector space during training. With uniformly distributed vectors representing such categories, we would not benefit from Distributional hyphothesis adapted to categories. Outputs of the model are embeddings trained for both – categories and for words.

7.5.4 Setup 4

First, the model trains the vectors representing categories (using *Setup 3*) and in the second round we use those pre-trained category vectors and continue with *setup 1*, using the pre-trained embeddings for categories. The main motivation is to have categories organised in vector space according to the meaning and help the words from document to end up on vector positions that have better semantic and syntactic representations.

7.6 Training

We previously tested our models on the English Wikipedia dump from June 2016[2]. The XML dump consist of 5,164,793 articles and 1,759,101,849 words. We firstly removed XML tags and kept only articles marked with respective id, further we removed articles with less than 100 words or less than 10 sentences. We removed categories that has less than 10 occurrences in between all articles. We have removed the articles without categories. The final corpus used for training consist of 1,554,079 articles. The Czech Wikipedia dump comes from March 2017. Detailed statistics on these corpora are shown in Tables 7.1 and 7.2. For an evaluation, we experiment with word analogy and a variety of word similarity datasets.

	English (dump statistics)
Articles	5,164,793
Words	1,759,101,849
	English (final clean statistics)
Articles	1,554,079
Avg. words per article	437
Avg. number of categories per article	4.69
Category names vocabulary	4,015,918

Table 7.1: Training corpora statistics. English Wikipedia dump from June 2016.

Word similarity datasets are used to measure the semantic similarity between pair of words. For English, these include WordSim-353 [Finkelstein et al., 2002], RG-65 [Rubenstein and Goodenough, 1965], RW [Luong et al., 2013], LexSim-999 [Hill et al., 2015], and MC-28 [Miller and Charles, 1991]. For Czech, only two datasets are available and these include RG-65 [Krcmár et al., 2011] and WordSim-353 [Cinková, 2016]. Both datasets consists of translated word pairs from English, re-annotated by Czech native speakers.

Word analogies are following the observation that the word representation can capture different aspects of meaning, [Mikolov et al., 2013a] introduced an evaluation scheme based on word analogies. The scheme consists of questions, e.g. which word is related to *man* in the same sense as *queen* is related to *king*? The correct answer should be *woman*. Such a question can be answered with a simple equation: $\text{vec}(king) - \text{vec}(queen) = \text{vec}(man) - \text{vec}(woman)$.

[2] <dumps.wikimedia.org>

	Czech (dump statistics)
Articles	575,262
Words	88,745,854
	Czech (final clean statistics)
Articles	480,006
Avg. words per article	308
Avg. number of categories per article	4.19
Category names vocabulary	261,565

Table 7.2: Training corpora statistics. Czech Wikipedia dump from June 2016.

We evaluate on English and Czech word analogy datasets, proposed by [Mikolov et al., 2013a] and [Svoboda and Brychcín, 2016], respectively. Word-phrases were excluded from the original datasets, resulting in 8869 semantic and 10,675 syntactic questions for English (19,544 in total), and 6018 semantic and 14,820 syntactic questions for Czech (20,838 in total).

7.6.1 Training Setup

We tokenize the corpus data. We use a simple tokeniser based on regular expressions. After the model is trained, we keep the most frequent words in the vocabulary ($|\boldsymbol{W}| = 300,000$). Vector dimension for all our models is set to $d = 300$. We always run 10 training iterations. The window size is set 10 to the left and 10 to the right from the center word $w_{j,i}$, i.e. $|\boldsymbol{C}_{j,i}| = 20$. The set of negative samples $\boldsymbol{N}$ is always sampled from unigram word distribution raised to 0.75 and has fixed size $|\boldsymbol{N}| = 10$. We do not use the sub-sampling of frequent words. The process of parameter estimation is described in detail in [Goldberg and Levy, 2014]. We prefixed categories to be unique in training and not interfere with words during the training phase.

fastText is trained on our Wiki. dumps (see results in Table 7.3 and 7.4). *LexVec* is tested only for English, trained on Wiki. 2015 and News-Crawl[3], has 7 billion tokens, vocabulary of 368,999 words and vectors of 300d. Both (fastText and LexVec) models use character n-grams of length 3-6 as subwords. For a comparison with much larger training data (only available

[3] <http://www.statmt.org/wmt14/translation-task.html>

		Word similarity				Word analogy		
	Model	**WS-353**	**RG-65**	**MC-28**	**Simlex-999**	**Sem.**	**Syn.**	**Total**
Baselines	fastText – SG 300d wiki	46.12	76.31	73.26	26.78	68.77	67.94	68.27
	fastText – cbow 300d wiki	44.64	73.64	69.67	38.77	69.32	81.42	76.58
	SG GoogleNews 300d 100B	68.49	76.00	80.00	46.54	78.16	76.49	77.08
	CBOW 300d wiki	57.94	68.69	71.70	33.17	73.63	67.55	69.98
	SG 300d wiki	64.73	78.27	82.12	33.68	83.64	66.87	73.57
	LexVec 7B	59.53	74.64	74.08	40.22	80.92	66.11	72.83
	CBOW 300d + Cat	63.20	78.16	78.11	40.32	77.31	68.68	72.13
	SG 300d + Cat	62.55	80.25	86.07	33.54	80.77	71.05	74.93

Table 7.3: Word similarity and word analogy results on English.

for English), we downloaded *GoogleNews100B*[4] model that is trained using Skipgram architecture on 100 billion words corpus and negative sampling, vocabulary size is 3,000,000 words.

Prefered model architecture

Previously, we talked about four different types of model architectures and approaches, how to incorporate the categories for training the word embeddings (see Sections 7.5.1, 7.5.2, 7.5.3 and 7.5.4 for further information). The results of different architectures are mainly presented on English.

For the Czech language, we chosed the model with Setup #3 defined in Section 7.5.3. We choose this setup due to its simplicity, faster and more stable training.

7.7 Results

In this section we present results of our DSMs improved with global information for Czech language.

As an evaluation measure for word similarity tasks we use Spearman correlation between system output and human annotations. For word analogy task we evaluate by accuracy of correctly returned answers. Results for English Wikipedia are shown in Table 7.3 and for Czech in Table 7.4. These

[4] <https://developer.syn.co.in/tutorial/bot/oscova/pretrained-vectors.html>

	Model	Word similarity			Word analogy		
		WS-353	**RG-65**	**MC-28**	**Sem.**	**Syn.**	**Total**
Baselines	fasttext – SG 300d wiki	67.04	67.07	72.90	49.03	76.95	71.72
	fasttext – CBOW 300d wiki	40.46	58.35	57.17	21.17	85.24	73.23
	CBOW 300d wiki	55.9	41.14	49.73	22.05	52.56	44.33
	SG 300d wiki	65.93	68.09	71.03	48.62	54.92	53.74
	CBOW 300d + Cat	54.31	47.03	49.31	42.00	62.54	58.69
	SG 300d + Cat	62	57.55	64.64	47.03	54.07	52.75

Table 7.4: Word similarity and word analogy results on Czech.

detailed results allow for a precise evaluation and understanding of the behaviour of the method. First, it appears that, as we expected, it is more accurate to predict entities when categories are incorporated.

7.8 Discussion

Type	Baseline	Cat
Antonyms (nouns)	15.72	7.14
Antonyms (adj.)	19.84	46.20
Antonyms (verbs)	6.70	5.00
State-cities	35.80	50.57
Family-relations	31.82	50.64
Nouns-plural	69.44	75.93
Jobs	76.66	95.45
Verb-past	51.06	61.04
Pronouns	11.58	10.42
Antonyms-acjectives	71.43	81.82
Nationalities	20.40	21.31

Type	Baseline	Cat.
Capital-common-countries	84.98	88.34
Capital-world	81.78	87.69
Currency	5.56	5.56
City-in-state	62.55	65.22
Family-relations	92.11	90.94
Adjective-to-adverb	25.38	29.38
Opposite	41.67	37.08
Comparative	79.14	78.82
Superlative	59.74	64.50
Present-participle	61.95	65.89
Nationality-adjective	91.39	98.69
Past-tense	63.66	66.59
Plural	74.19	71.67
Plural-verbs	62.33	46.33

Table 7.5: Detailed word analogy results comparison – left table shows Czech with CBOW and categories, right table shows English with CBOW and categories.

Distributional vector models capture some aspect of word co-occurrence statistics of the words in a language [Levy and Goldberg, 2014]. Therefore, if we allow that shared categories imply semantically similar textual data, these extended models produce semantically coherent representations, and we believe that the improvements presented in Tables 7.3 and 7.4 are the evidence for the existence of distributional hypothesis.

Our model on English also outperforms fastText architecture [Bojanowski et al., 2017], a recent improvement of Word2Vec with sub-word information. With our adaptation, the CBOW architecture gives similar performance to the Skipgram architecture trained on much larger data. On RG-65 word similarity test and semantic oriented analogy questions in Table 7.3 it gives better performance. We can see, that our model is powerful in semantics.

There is also significant performance gain on WS-353 similarity dataset and English language. Czech generally performs poorer, because there is less data for training and also because of the language properties. Czech has free word order and higher morphological complexity that influences the quality of resulting word embeddings. That is also the reason why the sub-word information tends to give much better results. However, our method shows significant improvement in semantics, where the performance with the Czech language has improved twofold (see Table 7.4).

The individual improvements of word analogy tests with CBOW architecture are available in Table 7.5. These detailed results allow for a precise evaluation and analyse the behaviour of our model.

In Czech, we see the biggest gain in understanding of the category "Jobs". This semantic category is specific to the Czech language as it distinguishes between feminine and masculine form of professions.

However, we do not see much difference in the section "Nationalities" that also relates countries and the masculine versus the feminine form of their citizens. We think this might be caused of lack data from Wikipedia. In Czech, we use mostly the masculine form in articles when talking about people from different countries. In a section "Pronouns" that deals with analogy questions such as: *"I,we"* versus *"you, they"*, we clearly cannot benefit from incorporating the categories. The biggest performance gain is as we expected in semantic oriented categories such as: *Antonyms, State-cities and Family-relations.*

English gives a slightly lower score in the *Family-relations* section of the analogy corpus. However, as English semantic analogy questions are already hitting correlations above 80% and especially for this section already more than 90%, we believe that we are already hitting the maximal capabilities of machine and humans agreement. This is the reason why we bring up the comparison with highly inflected language. In [Svoboda and Brychcín, 2016] and [Svoboda and Beliga, 2018] it has been shown that there is a room for the

performance improvement of current state-of-the-art word embedding models on languages from Slavic families – see in Chapter 4.

For the Czech language, we saw a drop in performance of the Skip-gram model. This might be caused by insufficient data for the reverse logic of training the Skip-gram architecture.

7.9 Conclusion

7.9.1 Contributions

Our model with global information extracted from Wikipedia significantly outperforms the baseline CBOW model. It provides similar performance compared with methods trained on much larger datasets.

We focus on the currently widely used CBOW method and the Czech language. As a source of global document (in this case, article) context we used Wikipedia which is available in 301 languages. Therefore, our method can be adopted to any other language without necessity of manual data annotation. The model can help to create word embeddings that perform better with smaller corpora.

7.9.2 Future work

The future community work might lead to integrate our model to the latest architectures such as *fastText* or *LexVec* and improve the performance further by incorporating the sub-word information. This further improvement together with our method can have even bigger impact on poorly resourced and highly inflected languages, such as Czech. Also we suggest to take a look into the other possibilities, for extracting useful information from Wikipedia and ways to use it during training – such as references, notes, literature, external links, summary info and others.

The global information data and trained word vectors for research purposes are at `<https://github.com/Svobikl/global_context/>`.

8 Summary

This thesis presents an overview of the current state-of-the-art approaches for distributional semantics. The performance of modeling semantics representation has rapidly improved during recent years with use of neural networks and deep learning techniques.

We chose to aim at a hard target and tried to beat current state-of-the-art methods to extract word embeddings on highly inflected languages. We explore further use of machine learning techniques in solving NLP problems where the semantic knowledge is crucial to solve the actual problem. We achieved second (respective first) place among 113 submitted systems at the famous SemEval competition with our STS system. We have presented a new ideas for extracting word embeddings. In all our studies we focus on achieving best results and engineering novel features. Results and corpora from all our papers are publicly available[1].

8.1 Conclusions

We present our contribution to distributional semantics:

- We have done in-depth research on machine learning methods used in NLP tasks (see Chapter 3). Different classifiers, namely Naive Bayes, SVM (Support Vector Machines), Feed-forward Neural Network and LSTM Neural Network were used on large-scale labeled corpora (see Chapters 4, 5 and 7).
- We explored several pre-processing techniques and employed various features and classifiers in order to achieve artificial understanding of semantics and syntax of text (see Chapter 5 and 6).
- We propose a simple model that benefits from global information extracted from Wikipedia (see Chapter 7).

[1] <https://github.com/Svobikl/>

We aim to investigate the effectiveness of several models based on distributonal semantics to catch the meaning of textual data with focus on highly inflected languages. We believe that semantics contains hidden information that can improve various NLP tasks.

Czech/Croatian as a representative of inflective language is an ideal environment for the study of various aspects of text understanding for inflectional languages. It is challenging because of its very flexible word order and many different word forms.

8.2 Contributions

The contributions of the thesis are the following:

- We build a first Czech and Croatian word analogy corpora and various Croatian word similarity corpora, researchers are now able to tune performance of their extracted word embeddings on those languages and this can also have impact across different tasks in NLP area.

  We studied different languages from various language families and experimented with state-of-the-art methods for word embeddings, namely: CBOW, Skip-gram, GloVe, FastText. We have made a first evaluation of Croatian and Czech word embeddings. Focusing on inflectional languages, we proved their difficulty to model. These languages have not gained as much attention until now. We believe that the results of our studies will help the community to focus more on highly inflected languages (see Chapter 4). All corpora are available online[1] for research purposes.

- We introduced the first dataset for semantic textual similarity of Czech sentences and corpora for Aspect based sentiment analysis[1]. We created strong baselines based on state-of-the-art approach for both STS and ABSA task – see Chapter 5 and Chapter 6). Thanks to the presented datasets, the NLP community is able to do further research of aspect based sentiment analysis and semantic textual similarity tasks with Czech language.

- Our research presented in Chapter 7 focuses on modeling distributional semantics that is the backbone research area in NLP. We developed the

new approach to train word emebeddings using the global information extracted from Wikipedia. We need less data to train high quality word embeddings – our method gives similar performance compared to current state-of-the-art methods trained on much larger datasets (up to 100× larger). Such improved models can be used as basic features for sentiment analysis, machine translation, named entity recognition, semantic textual similarity and many other tasks across NLP area.

8.3 Fulfilment of the Thesis Goals

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

Study the influence of rich morphology on the quality of meaning representation. Most of the publications listed in Appendix A are directly or indirectly related to this point. In [Svoboda and Brychcín, 2016] we reported results of our initial experiments with word embeddings and the Czech language. We created the first Czech word analogy corpus to test the quality of word embeddings. Further, in [Svoboda and Beliga, 2018] we built the first word analogy and various word similarity corpora and tested word embedding properties on the Croatian language, another representative of the Slavic language family. We confirmed the lack of performence against English and the need of further research of current state-of-the-art method with focus on the Slavic language family.

In [Hercig et al., 2016a] we created two new Czech corpora within the restaurant domain for the ABSA task and achieved state-of-the-art results for the Czech language. We dealt with specific aspects of Czech using stemming techniques. The word clusters from semantic spaces (CBOW and GloVe) and the stems used as a separate features proved to be very useful combination to deal with ABSA task for Czech.

Our paper [Svoboda and Brychcín, 2018a] presents the first Czech corpora and state-of-the art system for semantic textual similarity. We showed importance of data preprocessing with lemmatization/stemming techniques on the Czech language and the robustness of our system using lexical, syntactic and semantic fetures.

Article [Brychcín et al., 2019] generalizes the word analogy task across languages, to provide a new intrinsic evaluation method for cross-lingual semantic spaces. We experiment with six languages within different language families, including English, German, Spanish, Italian, Czech, and Croatian. The rest of the publications are related indirectly.

All our experiments shows a need of further NLP research and community focus on highly inflected languages.

Propose of novel approaches based on neural networks for improving the meaning representation of inflectional languages. In [Svoboda and Brychcín, 2018b] we extend Skip-Gram and Continuous Bag-of-Words Distributional word representations NN-based models via global context information. We present four new approaches, to enrich word meaning representation with such information. Our model with global information extracted from Wikipedia significantly outperform the baseline CBOW and Skipgram models. We tested on various similarity corpora and standard word analogy corpus. Our method gives similar performance compared with standard methods trained on much larger (100x) datasets.

Later, in [Svoboda and Brychcín, 2019] we test the properties of our new model with highly inflected language. Our methods need much less data to provide a state-of-the-art performance. The lack of data is usually significant, especially in low-resource languages such as Czech.

In [Brychcín and Svoboda, 2016, Svoboda and Brychcín, 2018a] we explore semantic textual similarity using lexical, syntactic and semantic information on both Czech and English languages. We experiment with tree-structured Recurrent Neural Network with a complex computational unit and CBOW, SkipGram and GloVe models. We have also experimented with Paragraph2Vec NN-based model that includes not only the word vectors of each word in the context as CBOW/Skip-gram does, but also the paragraph vector during the training procedure. In the monolingual task, our system achieves mean Pearson correlation of 75.7% compared with human annotators. Our system was ranked second among 113 submitted systems. In the cross-lingual task, our system has correlation of 86.3% and is ranked first among 26 systems. To deal with Czech rich morphology, we use lemmatization and stemming techniques to preprocess the training data.

Use of distributional semantic models for improving NLP tasks. In our publications we specifically focus on achieving best results and engineering novel features. In [Brychcín and Svoboda, 2016, Svoboda and Brychcín, 2018a] we build novel models for Semanic Textual Similarity task, previewed in previous point.

In [Hercig et al., 2016b] we build system for ABSA using distibutional semantic models. As already discussed, in [Hercig et al., 2016a] we examine the effectiveness of several unsupervised methods for latent semantics discovery as features for aspect-based sentiment analysis (ABSA) on Czech language.

In article [Brychcín et al., 2019] we created a unified semantic space for six languages, which produces very promising results on word analogy task between any pair of languages.

The rest of the publications are related to this point indirectly, our new approach [Svoboda and Brychcín, 2018b] for extracting high quality word embeddings will likely have an further impact on variety of NLP tasks.

8.4 Future Work

As an outcome from this thesis, we believe that using our method described in Chapter 7 together with a sub-word information can have even bigger impact on poorly resourced and highly inflected languages, such as Czech from the Slavic family. Therefore, the future community work might lead to integrate our model into the latest architectures such as *fastText* or *LexVec* and improve the performance further by incorporating sub-word information.

Use of external sources of information (such as part-of-speech tags, NER, or lemma/stemming and character n-grams) during training process of current state-of-the-art neural network based word embedding methods might lead to further performance gains. Also, we suggest to take a look into other possibilities, for extracting useful information from Wikipedia and ways to use it during training – such as references, notes, literature, external links, summary info (usually displayed on the right side of the screen) and others.

The NLP community can use either our word analogy and word similarity corpora to investigate performance bottlenecks of systems they applying to Czech or Croatian languages. We showed that corpora preprocessing which

simplifies morphological variations, such as stemming or lemmatization procedures, could also have an effect on quality of word embeddings and may be one of the future research directions on Czech and Croatian languages.

Researchers can also use our state-of-the-art models presented on STS and ABSA tasks including corpora for further research improvements on particular tasks.

A Author's publications

A.1 Conference Publications

[**c1**] L. Svoboda and T. Brychcín. New word analogy corpus for exploring embeddings of czech words. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 103–114. Springer, 2016

[**c2**] T. Brychcín and L. Svoboda. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. *In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June*, 16, 2016

[**c3**] T. Hercig, T. Brychcín, L. Svoboda, and M. Konkol. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June*, volume 16, 2016b

[**c4**] L. Svoboda and S. Beliga. Evaluation of croatian word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018

[**c5**] L. Svoboda and T. Brychcín. Czech dataset for semantic textual similarity. In *International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining*, pages 213–221. Springer, 2018a

A.2 Journal Publications

[**j1**] T. Hercig, T. Brychcín, L. Svoboda, M. Konkol, and J. Steinberger. Unsupervised methods to improve aspect-based sentiment analysis in czech. *Computación y Sistemas*, 20(3):365–375, 2016a

[**j2**] L. Svoboda and T. Brychcín. Improving word meaning representations using wikipedia categories. *Neural Network World*, 28(6):523–534, 2018b

[j3] L. Svoboda and Brychcín. Enriching word embeddings with global information and testing on highly inflected language. *Computación y Sistemas*, accepted, waiting for print, 2019

[j4] T. Brychcín, S. Taylor, and L. Svoboda. Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*, 135:287 – 295, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.06.021

Bibliography

E. Agirre, M. Diab, D. Cer, and A. Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 385–393, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=2387636.2387697>.

E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. *sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1004>.

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL <http://www.aclweb.org/anthology/S14-2010>.

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S15-2045>.

R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. *CoNLL-2013*, page 183, 2013.

J. Andreas and D. Klein. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology-new/P/P14/P14-2133.bib>.

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

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

D. Bär, C. Biemann, I. Gurevych, and T. Zesch. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation, held in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada, June 2012.

J. R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279 –1296, Aug. 2000. ISSN 0018-9219.

Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5 (2):157–166, 1994.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, Mar. 2003. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=944919.944966>.

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

Y. Bengio, Y. LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5), 2007.

G. Berardi, A. Esuli, and D. Marcheggiani. Word embeddings go to italy: A comparison of models and training datasets. In *IIR*, 2015.

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

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

A. Z. Broder. On the resemblance and containment of documents. In *SEQUENCES '97 Proceedings of the Compression and Complexity of Sequences*, pages 21–29, Jun 1997. doi: 10.1109/SEQUEN.1997.666900.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

T. Brychcín and I. Habernal. 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, September 2013. INCOMA Ltd. Shoumen, BULGARIA. URL <http://www.aclweb.org/anthology/R13-1016>.

T. Brychcín and M. Konopík. Morphological based language models for inflectional languages. In *Proceedings of IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, 2011.

T. Brychcín and M. Konopík. Latent semantics in language models. *Computer Speech & Language*, 33(1):88–108, 2015.

T. Brychcín and M. Konopík. Hps: High precision stemmer. *Information Processing & Management*, 51(1):68 – 91, 2015. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2014.08.006. URL <http://www.sciencedirect.com/science/article/pii/S0306457314000843>.

T. Brychcín and P. Král. Novel unsupervised features for czech multi-label document classification. In *Mexican International Conference on Artificial Intelligence*, pages 70–79. Springer, 2014.

T. Brychcín and L. Svoboda. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. *Proceedings of SemEval*, pages 588–594, 2016.

T. Brychcín and L. Svoboda. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. *In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June*, 16, 2016.

T. Brychcín, M. Konkol, and J. Steinberger. Uwb: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822, 2014.

T. Brychcín, S. Taylor, and L. Svoboda. Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*, 135:287 – 295, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.06.021.

W. G. Charles. Contextual correlates of meaning. *Applied Psycholinguistics*, 21(4):505–524, 2000.

C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, pages 2635–2639, Singapore, September 2014.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

F. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. In *Proceedings of the 2001 conference on empirical methods in natural language processing*, 2001.

S. Cinková. Wordsim353 for czech. In *International Conference on Text, Speech, and Dialogue*, pages 190–197. Springer, 2016.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398, 2011.

C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL <http://dx.doi.org/10.1023/A:1022627411411>.

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

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

H. Demir and A. Ozgur. Improving named entity recognition for morphologically rich languages using word embeddings. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 117–122. IEEE, 2014.

L. Dolamic and J. Savoy. Indexing and stemming approaches for the czech language. *Information Processing and Management*, 45:714–720, November 2009. ISSN 0306-4573.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 01 1948. doi: 10.1145/584091.584093.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

M. Elrazzaz, S. Elbassuoni, K. Shaban, and C. Helwe. Methodical evaluation of arabic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 454–458, 2017.

S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face detection using deep convolutional neural networks. *CoRR*, abs/1502.02766, 2015. URL <http://arxiv.org/abs/1502.02766>.

M. Faruqui and C. Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.

J. R. Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL <http://dl.acm.org/citation.cfm?id=1625275.1625535>.

E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009.

J. Gao, L. Deng, M. Gamon, X. He, and P. Pantel. Modeling interestingness with deep neural networks, Dec. 17 2015. US Patent 20,150,363,688.

D. Gildea and T. Hofmann. Topic-based language models using em. In *Proceedings of Eurospeech*, pages 2167–2170, 1999.

Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

S. Gouws and A. Søgaard. Simple task-specific bilingual word embeddings. In *HLT-NAACL*, pages 1386–1390, 2015.

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

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

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

M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6):989–993, 1994.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL <http://doi.acm.org/10.1145/1656274.1656278>.

L. Han, A. L. Kashyap, T. Finin, J. Mayfield, and J. Weese. Umbc_ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1005>.

Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.

R. Hecht-Nielsen. Neurocomputing / robert hecht-nielsen. *SERBIULA (sistema Librum 2.0)*, 359, 02 1990. doi: 10.1038/359463a0.

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

T. Hercig, T. Brychcín, L. Svoboda, and M. Konkol. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June*, volume 16, 2016b.

F. Hill, K. Cho, S. Jean, C. Devin, and Y. Bengio. Not all neural embeddings are born equal. *CoRR*, abs/1410.0718, 2014.

F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

S. Hochreiter and J. Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 873–882, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=2390524.2390645>.

R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. *CoRR*, abs/1412.1058, 2014. URL <http://arxiv.org/abs/1412.1058>.

R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350, 2015.

D. Jurgens and K. Stevens. The s-space package: an open source package for word space models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 30–35. Association for Computational Linguistics, 2010.

A. Kachites McCallum. Mallet: A machine learning for language toolkit. 01 2002.

G. Karypis. Cluto-a clustering toolkit. Technical report, MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE, 2002.

H. J. Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10): 947–954, 1960.

Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014. URL <http://arxiv.org/abs/1408.5882>.

M. Konkol. Brainy: A machine learning library. In *International Conference on Artificial Intelligence and Soft Computing*, pages 490–499. Springer, 2014.

M. Konkol and M. Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In *International Conference on Text, Speech and Dialogue*, pages 153–160. Springer, 2013.

M. Konkol, T. Brychcín, and M. Konopík. Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470–3479, 2015a.

M. Konkol, T. Brychcín, and M. Konopík. Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470–3479, 2015b.

M. Köper, C. Scheible, and S. S. im Walde. Multilingual reliability and" semantic" structure of continuous word spaces. In *IWCS*, pages 40–45, 2015.

L. Krcmár, M. Konopík, and K. Jezek. Exploration of semantic spaces obtained from czech corpora. In *DATESO*, pages 97–107, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

J. Lafferty, A. Mccallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc ICML*, 01 2002.

T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, 1997.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

O. Levy and Y. Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308, 2014.

O. Levy, A. Søgaard, and Y. Goldberg. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, 2017.

K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208, 1996.

M.-T. Luong, R. Socher, and C. D. Manning. Better word representations with recursive neural networks for morphology. In *CoNLL*, Sofia, Bulgaria, 2013.

G. Maltese, P. Bravetti, H. Crepy, B. J. Grainger, M. Herzog, and F. Palou. Combining word and class-based language models: a comparative study in several languages using automatic and manual wordclustering techniques. In *Proceedings of 7th European Conference on Speech Communication and Technology*, pages 21–24. Eurospeech, 2001.

C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN 0521865719. URL <http://nlp.stanford.edu/IR-book/>.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL <http://www.aclweb.org/anthology/P/P14/P14-5010>.

D. S. McNamara. Computational methods to extract meaning from text and advance theories of human cognition. *Topics in Cognitive Science*, 3(1): 3–17, 2011.

T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3, 2010.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013c.

G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

D. J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *IJCAI*, volume 89, pages 762–767, 1989.

F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005.

P. Pantel. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 125–132. Association for Computational Linguistics, 2005.

D. B. Paul and J. M. Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 357–362, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. ISBN 1-55860-272-0. doi: 10.3115/1075527.1075614. URL <https://doi.org/10.3115/1075527.1075614>.

F. J. Pelletier. The principle of semantic compositionality. *Topoi*, 13(1): 11–24, 1994.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 27–35, 01 2014. doi: 10.3115/v1/S14-2004.

M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos. 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, 2015.

M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30, 2016.

J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

B. Riordan and M. N. Jones. Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science*, 3(2):303–345, 2011.

D. L. Rohde, L. M. Gonnerman, and D. C. Plaut. An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Psychology*, 7:573–605, 2004.

H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

A. Salle and A. Villavicencio. Incorporating subword information into matrix factorization word embeddings. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 66–71, New Orleans, jun 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1209. URL <https://www.aclweb.org/anthology/W18-1209>.

A. Salle, M. Idiart, and A. Villavicencio. Matrix factorization using window sampling and negative sampling for improved word representations. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 419, 2016.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975. ISSN 0001-0782.

R. S. Scalero and N. Tepedelenlioglu. A fast new algorithm for training feedforward neural networks. *Signal Processing, IEEE Transactions on*, 40 (1):202–210, 1992.

H. Schutze and J. O. Pedersen. Information retrieval based on word senses. *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 08 1996.

Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014.

S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. Murthy. Improvements to the smo algorithm for svm regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, Sep 2000. ISSN 1045-9227. doi: 10.1109/72.870050.

X. Shuai, X. Liu, T. Xia, Y. Wu, and C. Guo. Comparing the pulses of categorical hot events in twitter and weibo. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 126–135. ACM, 2014.

S. K. Siencnik. Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 239–243, 2015.

J. Šnajder, S. Padó, and Ž. Agić. Building and evaluating a distributional memory for croatian. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 784–789, 2013.

J. Steinberger, T. Brychcín, and M. Konkol. Aspect-level sentiment analysis in czech. In *Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 24–30, 2014.

J. Straková, M. Straka, and J. Hajic. Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *ACL (System Demonstrations)*, pages 13–18, 2014.

M. Sultan, S. Bethard, and T. Sumner. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230, 2014a. ISSN 2307-387X. URL <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/292>.

M. A. Sultan, S. Bethard, and T. Sumner. Sentence similarity from word alignment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 241–246, Dublin, Ireland, August 2014b. Association for Computational Linguistics and Dublin City University. URL <http://www.aclweb.org/anthology/S14-2039>.

M. A. Sultan, S. Bethard, and T. Sumner. Dls@cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153, Denver, Colorado, June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S15-2027>.

L. Svoboda. Distributional semantics using neural networks: technical report no. dcse/tr-2016-04. 2016. URL <https://dspace5.zcu.cz/bitstream/11025/25377/1/Svoboda.pdf>.

L. Svoboda and S. Beliga. Evaluation of croatian word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

L. Svoboda and Brychcín. Enriching word embeddings with global information and testing on highly inflected language. *Computación y Sistemas*, accepted, waiting for print, 2019.

L. Svoboda and T. Brychcín. Czech dataset for semantic textual similarity. In *International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining*, pages 213–221. Springer, 2018a.

L. Svoboda and T. Brychcín. Improving word meaning representations using wikipedia categories. *Neural Network World*, 28(6):523–534, 2018b.

L. Svoboda and T. Brychcín. New word analogy corpus for exploring embeddings of czech words. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 103–114. Springer, 2016.

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1150>.

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015b.

Y. Tam and T. Schultz. Unsupervised language model adaptation using latent semantic marginals. In *Proceedings of Interspeech*, 2006.

Y.-C. Tam and T. Schultz. Dynamic language model adaptation using variational bayes inference. In *Ninth European Conference on Speech Communication and Technology*, 2005.

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

P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

S. Upadhyay, M. Faruqui, C. Dyer, and D. Roth. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1661–1670, 2016.

K. Veselovská, J. Hajič jr., and J. Šindlerová. Creating annotated resources for polarity classification in Czech. In J. Jancsary, editor, *Proceedings of KONVENS 2012*, pages 296–304. ÖGAI, September 2012. PATHOS 2012 workshop.

I. Vulić and M.-F. Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–372. ACM, 2015.

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

S. Wang, D. Schuurmans, F. Peng, and Y. Zhao. Semantic n-gram language modeling with the latent maximum entropy principle. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP03)*, 2003.

M. J. Wise. Yap3: Improved detection of similarities in computer program and other texts. In *Proceedings of the Twenty-seventh SIGCSE Technical*

Symposium on Computer Science Education, SIGCSE '96, pages 130–134, New York, NY, USA, 1996. ACM. ISBN 0-89791-757-X. doi: 10.1145/236452.236525. URL <http://doi.acm.org/10.1145/236452.236525>.

T. Yokoyama, T. Shinozaki, K. Iwano, and S. Furui. Unsupervised language model adaptation using word classes for spontaneous speech recognition. In *Proceedings of IEEE-ISCA Workshop on Spontaneous Speech Processing and Recognition*, pages 71–74, 2003.

F. M. Zanzotto, I. Korkontzelos, F. Fallucchi, and S. Manandhar. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1263–1271, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1873781.1873923>.

Y. Zhang and B. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

L. Zuanovic, M. Karan, and J. Šnajder. Experiments with neural word embeddings for croatian. In *Proceedings of the 9th Language Technologies Conference*, pages 69–72, 2014.