TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja
Text Analysis and Knowledge Engineering Lab
Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS no. 1351

Application of Compositional Distributional Semantics for Semantic Text Similarity

Luka Skukan

Zagreb, February 2017

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING
MASTER THESIS COMMITTEE

Zagreb, 13 October 2016

MASTER THESIS ASSIGNMENT No. 1351

Student: **Luka Skukan (0036465873)**
Study: Computing
Profile: Computer Science

Title: **Application of Compositional Distributional Semantics for Semantic Text Similarity**

Description:

Compositional distributional semantics models the meaning of multi-word units and sentences by combining the distributional representations of the constituting words. The use of these representations as features for machine learning algorithms has proven to be useful in a number of downstream natural language tasks (NLP), such as question answering, semantic relatedness, paraphrase detection, and sentiment analysis.

The topic of this thesis is the application of compositional distributional semantic models, with a special focus on semantic text similarity (STS) for Croatian. Compile a sufficiently large dataset for Croatian STS and carry out a detailed analysis. Implement and compare various STS models, including TakeLab's STS system (Šarić et al., 2012), skip-gram (Mikolov et al., 2013), and skip-thought vector models (Kiros et al., 2015). Provide a software implementation and a web-application accompanying it. Perform experimental evaluation using the implemented models for supervised and unsupervised semantic text similarity on texts in Croatian, as well as a detailed error analysis. All references must be cited, and all source code, documentation, executables, and datasets must be provided with the thesis.

Issue date: 14 October 2016
Submission date: 3 February 2017

Mentor:

Associate Professor Jan Šnajder, PhD

Committee Chair:

Full Professor Siniša Srbljić, PhD

Committee Secretary:

Assistant Professor Tomislav Hrkać, PhD

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 13. listopada 2016.

Predmet: **Analiza i pretraživanje teksta**

DIPLOMSKI ZADATAK br. 1351

Pristupnik: **Luka Skukan (0036465873)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Primjena kompozicijske distribucijske semantike u zadatku semantičke sličnosti teksta**

Opis zadatka:

Kompozicijska distribucijska semantika modelira značenje skupina riječi i rečenica kombiniranjem distribucijskih reprezentacija značenja njihovih sastavnih riječi. Upotreba takvih reprezentacija kao ulaznih značajki modela strojnog učenja pokazala se korisnom u mnogim zadacima obrade prirodnoga jezika, poput odgovaranja na pitanja, zadataka semantičke povezanosti, otkrivanja parafraza te analize sentimenta.

Tema ovog rada jest primjena kompozicijskih distribucijskih semantičkih modela, s posebnim naglaskom na zadatak semantičke sličnosti teksta (SST) u hrvatskome jeziku. Izgraditi skup podataka prikladne veličine za zadatak SST za hrvatski jezik te ga detaljno analizirati. Implementirati i usporediti razne modele za SST, uključujući TakeLabov sustav za SST (Šarić i dr., 2012) te vektorske modele skip-gram (Mikolov i dr., 2013) i skip-thought (Kiros i dr., 2015). Razviti programsku implementaciju i pripadajuću web-aplikaciju. Provesti eksperimentalno vrednovanje sustava za nadziranu i nenadziranu semantičku sličnost hrvatskih tekstova korištenjem implementiranih modela, kao i detaljnu analizu pogrešaka. Radu priložiti izvorni i izvršni kod razvijenog sustava, skupove podataka i programsku dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 14. listopada 2016.
Rok za predaju rada: 3. veljače 2017.

Mentor:

Izv. prof. dr. sc. Jan Šnajder

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za diplomski rad profila:

Prof. dr. sc. Siniša Srbljić

I would like to thank, first and foremost, my supervisor, Jan Šnajder, PhD. He has been extremely helpful, meticulous and patient. Without him this thesis would not be possible. Secondly, I thank my colleagues at Infinum, especially my fellow members of the JavaScript team, for support, understanding, and lots of manual annotation.

CONTENTS

1. Introduction

We live in an age of nearly infinite data, with corpora spanning nearly the entire public content of the Internet in a given language, like ukWaC (Ferraresi et al., 2008), frWaC (Ferraresi et al., 2010), and others, allowing researchers to work with over 2 trillion tokens and over 300 billion n-grams (Brants et al., 2007). This has driven the field of Natural Language Processing (NLP) to new heights, allowing researchers to use data-hungry models such as deep neural networks. Some of the widely studied approaches that have benefited greatly from this are distributional models of semantics. With the advent of social Internet content such as social networks to provide even more content, we've seen use of vector space models to model and reason about language learning rates (Landauer and Dumais, 1997), to automatically extract thesauri (Curran, 2004), polysemous word sense ranking (Padó and Lapata, 2007), and many other tasks in NLP and information retrieval. Likewise, we've seen new methods arise in the field of distributional semantics, mostly based on neural networks. Some of the most notable of those are the continuous Bag-of-Words and skip-gram models (Mikolov et al., 2013a).

However, due to the natural language's infinite capacity to produce new bodies of text from this finite amount of words, the distributional semantic approach is limited to small units of text, such as words and n-grams. Compositional distributional semantics is the field that tackles this issue, expanding the approach to larger units of text, like sentences, paragraphs, or even entire documents, and it has seen similarly significant improvements. Like distributional semantics, it has benefited from the increased ability to use deep learning models, resulting in an increased usage of neural networks for composition of distributional models for certain tasks (Socher et al., 2010, 2011, 2012), as well as creation of successful generalized compositional models, such as the paragraph vector (Le and Mikolov, 2014), and the especially notable skip-thought vector (Kiros et al., 2015).

While these advances have been made for the English language and on documents written in the English language, these models have been successfully applied to other languages as well, such as Italian (Berardi et al., 2015), French, German, Russian

(Servan et al., 2016), and even Croatian (Zuanović et al., 2014; Šnajder and Almić, 2015).

The goal of this thesis is to provide an overview of the fields of distributional semantics and compositional distributional semantics, to provide models for applying them to the Croatian language, and to extend the work done for distributional semantics in Croatian to compositional distributional semantics. These models are developed on a prepared dataset and evaluated on a variety of natural language processing tasks.

The rest of the thesis is organized as follows. An overview of the fields of distributional semantics and compositional distributional semantics if offered in Chapter 2. Chapter 3 describes how distributional and compositional distributional semantic models were trained to present features in the Croatian language. Chapter 4 contains a number of experiments in supervised and unsupervised natural language processing tasks, as well as a discussion of the results and an error analysis. Additionally, a simple website is described, showcasing one of the experiments performed in that chaptered. Finally, the conclusion is presented in Chapter 5.

2. Compositional Distributional Semantics

When dealing with units of text greater than words, it is often useful to have a representation of their semantics, in some manner of a semantic space. Compositional distributional semantic models are representations of such phrases, in such a way that they represent the semantics of larger units of texts, such as phrases, sentences, or paragraphs, in a semantic vector space. This approach is based on two assumptions: one is the *distributional hypothesis* (Harris, 1968), stating that words that appear in similar contexts have similar meanings; The other is the *principle of compositionality*, stating that the meanings of composite expressions can be determined from the meanings of its constituents, together with the rules used to combine them.

The idea is not new, and was most famously expressed by Frege (1884), who cautions that one must never ask for the meaning of a word in isolation, but only in the context of a statement. One of the linguistic bases for the principle of compositionality is the the *productivity argument* – the idea that humans only know the meaning of words and the rules of their composition, yet manage to understand and produce entirely novel sentences, having never heard them uttered before (Kartsaklis, 2014). However, no matter how instinctively obvious this idea might seem, it has proven to be far from trivial to represent computationally.

2.1. Distributional Semantic Models

The distributional hypothesis presents a framework in which to semantically represent words, encoding their relations to other words in form of a structure, encoding them in a semantic vector space.

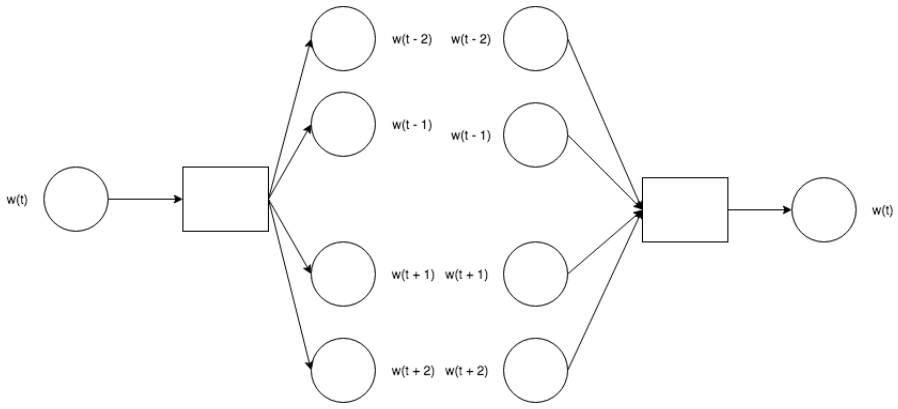
The following overview is based on (Kartsaklis, 2014), and presented with minor modifications referencing more recent work.

Traditional models of semantic representations fall under one of three categories:

semantic networks, feature-based models, and semantic spaces (and related topic models) (Markman, 2013). Semantic networks (Collins and Quillian, 1969) represent concepts as graph nodes, while semantic relationships between them are the edges. For example, such relationships might be `has(bear, fur)` or `is(bear, mammal)`. In such a framework, relationships can be denoted by the number, type, and length (i.e., number of edges) between two concepts (nodes). Semantically more related words will have a shorter path, or a larger number of short connections between them. The weakness of the original model is that the graphs need to be hand-modeled by human authors, who encode the relationships, which does not reflect the richness and changing nature of natural language. More recently, attempts were made (Steyvers and Tenenbaum, 2005) to create semantic networks from word association norms (Nelson et al., 2004), which is a relative encoding of free association between words, and from the WordNet (Miller, 1995). However, these models can only model a smaller vocabulary, much less than the vocabulary of an adult human speaker.

The second approach are feature-based models, in which words are represented as feature lists (Smith and Medin, 1981). These features are sometimes manually modeled by the researchers (Hinton and Shallice, 1991), and in other cases obtained by asking native speakers to determine which features are most important in their language (Andrews et al., 2009). The words are then encoded as numerical representations inside this feature list. These models face several difficulties – most importantly, they require several annotators for encoding each word into its feature, and the quality of these features is determined heavily by the amount of time and manpower spent for encoding each word (Sloman and Rips, 1998). In practice, this limits the use of these models to only smaller lexicons.

Finally, there are the semantic space models, which are based on the distributional hypothesis, which are used most by far in the recent years. Since similar words (e.g. “dog” and “cat”) occur in similar contexts, these models measure similarity *quantitatively*, by representing them as vectors in high-dimensional space that represent co-occurrence of similar contextual elements. These elements are most commonly words, but can also be paragraphs or documents (Landauer and Dumais, 1997), n-grams (Jones and Mewhort, 2007), or other, more complex, representations. When using such a vector representation, one has the advantage of being able to simply compare representations of words using distance metrics, such as Euclidean distance or cosine similarity. There are a number of well-known semantic space representations, among them the *Hyperspace Analog to Language model*, *Latent Semantic Analysis*, and the *Continuous Bag-of-Words* and *Continuous Skip-gram* models, as well as related models such



(a) Skip-Gram **(b)** CBOW

Figure 2.1: Skip-gram (a) and Continuous Bag-of-Words (b) models (Mikolov et al., 2013a)

as *probabilistic topic models*.

The Hyperspace Analog to Language (HAL) model (Lund and Burgess, 1996) represents words via vectors corresponding to the words' co-occurrence in the corpus, using a fixed-size window to construct the co-occurrence record, taking the distance from the word into account by making the tie stronger if the word is closer. The process moves the window over all the words, creating a co-occurrence matrix, where the cells correspond to the sum of the co-occurrence counts for the word pair. The order of words in the word pair matters, and different cells are used for pairs (x, y) and (y, x).

The Latent Semantic Analysis (LSA) model (Landauer and Dumais, 1997) takes a similar approach, creating *tf-idf* co-occurrence matrix from a large corpus of documents. Matrix decomposition methods are then applied to this matrix to reduce its dimensionality and make it more informative.

Mikolov et al. (2013a) present a novel connectionist approach through the deep-learning-based Continuous bag-of-words (CBOW) and skip-gram models. These approaches model the similarity between words to either infer the most likely neighboring words from a given word (skip-gram) or a most likely word given its neighbors (CBOW). Since their introduction, these models have become the de-facto go-to solution for distributed semantic representations, being reasonably quick to train and achieving state-of-the-art results.

Probabilistic topic models (Blei et al., 2003; Griffiths et al., 2007) offer an alternative to the semantic space models. Similar to the LSA model, they assume a latent relationship between words in a corpora and the topic they appear in, and likewise compute a reduced-dimensionality description of the words and documents that they are linked to. Instead of as vectors, the words are represented as probability distribu-

tions over topics, while the topics are represented as distributions over the words. As such, the topics' contents are represented by the words given high probability in this distribution. It is also important to note that this model is *generative* – the distribution can be used to generate representative texts for a selected topic.

2.2. Composition of Distributional Semantic Models

To structure the semantic of larger structures or bodies of text from the semantic representations of their components, it is necessary to compose them in some manner. When dealing with composing the aforementioned distributional models, this approach is called *compositional distributional semantics*, and the resulting models are *compositional distributional semantic models*.

In its most naïve format, we can describe the meaning of the whole as a combination, or a function, of the meanings of its parts (Partee, 1995). For representations of two constituents, $\mathbf{x}$ and $\mathbf{y}$, a representation of their combination can be written as:

$$f(\mathbf{x}, \mathbf{y}) \tag{2.1}$$

This principle can then be extended to as many constituents as desired.

However, it is obvious even to the casual observer that this method of composition is not sufficient. The same constituent units can be used to represent non-synonymous structures. Syntactic structure is likewise significant when representing the semantic value of the whole. As an example, the sentences *"He helped the man get rich"* and *"The rich man helped get him."* are made out of the same basic lexemes, but have significantly different meanings.

Partee (1995) therefore proposes that the value is a combination of the representations of the constituents with the semantic relationship taken into consideration as R_s, represented as:

$$f(\mathbf{x}, \mathbf{y}, R_s) \tag{2.2}$$

Even this formulation might not suffice to represent the composition of meanings, as Lakoff (1977) suggests, stating that the meaning of the whole is greater than the meaning of its parts. He implies that the knowledge of the language and the knowledge of the world around us, provide additional context and information. As such, the knowledge would be represented as K in:

$$f(\mathbf{x}, \mathbf{y}, R_s, K) \tag{2.3}$$

In practical terms, this knowledge can be represented by various resources, like WordNet (Miller, 1995), but is often ignored as it is most difficult to model of the parameters.

2.2.1. Simple vector operations

As distributional semantic models are today most often vector space models, the function f is typically a vector composition operation.

Addition (equation 2.4) and *averaging* (equation 2.5) are very common operations over vectors, and most prevalent in literature (Landauer and Dumais, 1997; Kintsch, 2001). They are easy to implement, and reasonably quick to execute in simulations and experiments. Furthermore, they have been shown to work reasonably well for combinations of few word representations, such as word pairs (Mitchell and Lapata, 2010). However, it is important to note that, as vector addition is a commutative operation, these operations produce bag-of-words representations. All syntactic relationships between words are lost, and these operations effectively correspond to equation 2.1. These operations effectively represent the meaning of the whole as a blending of the meanings of its constituent parts.

Another important thing to note about addition and averaging is that the two are equivalent under the cosine similarity measure, since these two operations result in vectors differing only by a constant factor.

$$\vec{r_i} = \vec{x}_1 + \vec{x}_2 + \ldots + \vec{x}_n \tag{2.4}$$

$$\vec{r_i} = \frac{\vec{x}_1 + \vec{x_2} + \cdots + \vec{x}_n}{n} \tag{2.5}$$

$$r_i = x_{1i} \odot x_{2i} \odot \ldots \odot x_{ni}, \forall i \in \{1, \ldots, m\} \tag{2.6}$$

Elementwise vector multiplication, as described by equation 2.6 (where n is the number of vectors and m the length of each vector, while $\odot$ is the symbol for elementwise multiplication), is a popular alternative. It is likewise computationally simple, and maintains the resultant vector length. Like addition and averaging, it is commutative, and the resultant representation is still a bag-of-words model, and a mixture of the meanings of the constituents. As such, these compositional representations are sometimes referred to as *vector mixture models* (Kartsaklis, 2014). However, unlike addition

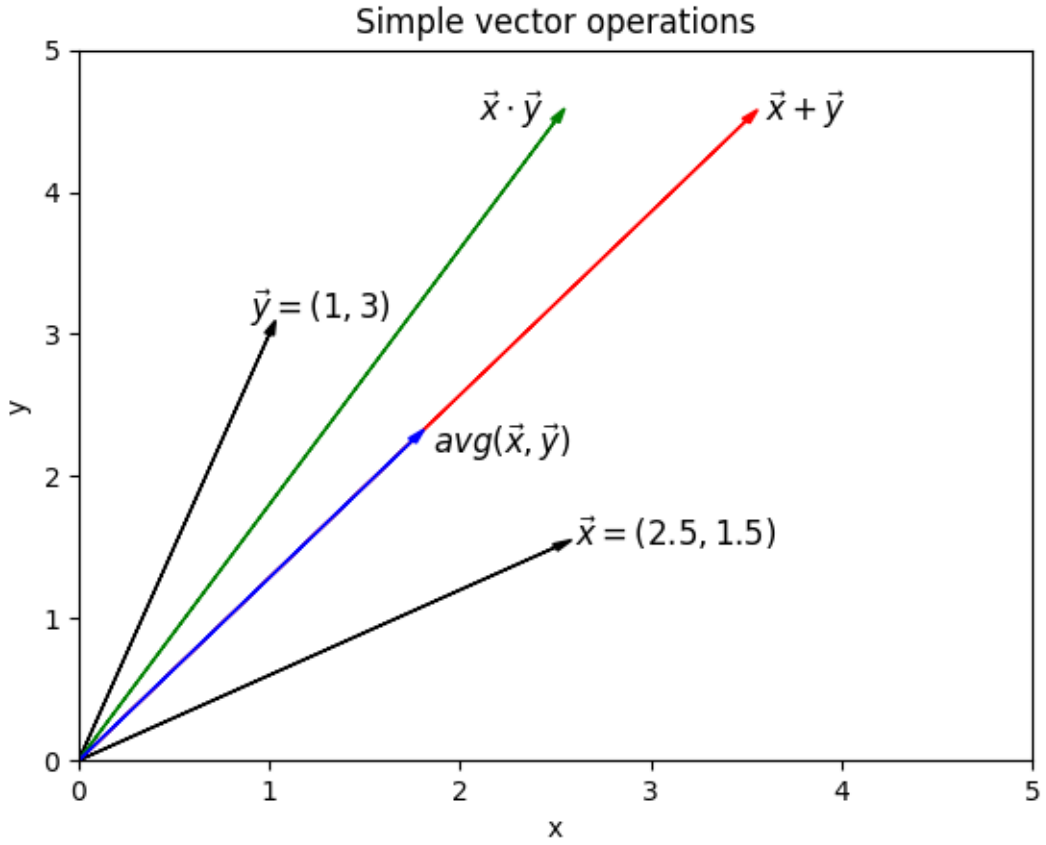


Figure 2.2: Addition, averaging, and multiplication of vectors

and averaging, multiplication eliminates a vector component entirely if any of the vectors has the component set to 0, and can be seen more as an intersection between the values.

Vector mixture models are among the simplest, if not the simplest, ways of composition in distributional compositional semantics. Despite those facts, they have been widely used, and have often served as baselines for more complex models of composition (Kartsaklis, 2014). Perhaps surprisingly, Mitchell and Lapata (2010) show that vector addition and multiplication can sometimes be (almost) as effective as more complex state-of-the-art methods for certain use cases.

A simple alternative that partially preserves syntactic relationships is *concatenation*. Under this operation, vectors of constituent words or lexemes are simply concatenated together to form a vector of the larger language unit. However, the primary issue with this approach is that, unless the word count of the text groups is the same, the resulting vectors differ in length. Unless resolved by further processing, this poses a problem for many machine learning models, if the language constructs used are of

varying length. Still, concatenation has been shown to be useful in certain tasks of combining distributed vector representations of words (Garten et al., 2015).

Since these simple models have obvious downsides, attempts were made to improve them by making modifications to the base combinator function. One notable example is by Kintsch (2001), who attempts to model predicate-argument pairs by adding not only their representations, but the representations of the closest neighbours of the predicate (in semantic space), in an attempt to strengthen the features of the predicate and its bond with the argument. It was, however, shown to perform worse than the basic models outside of a set of hand-crafted examples (Mitchell and Lapata, 2010).

2.2.2. Tensor product models

As we have seen, the simplicity of the vector mixture based models comes at the price of loss of syntactic structure. While this might not prove to be too much of an issue in very small samples, such as word pairs, it can become insufficient when dealing with larger textual constructs. For example, in these representations, the sentences *"The man ate a shark"* and *"A shark ate the man"* are identical.

These issues motivated an investigation into non-commutative combinator operations, such as tensor products. (Smolensky, 1990) originally proposes the tensor product representation, described by equation 2.7, where c_i^v is the value of vector $\vec{v}$ in position i, while $\vec{n}_i$ is a one-hot vector with the value 1 in dimension i, with all other dimensions having a value of 0. Simply put, the tensor product increases the dimensionality of the constituents by creating a tensor of one order higher by multiplying all possible pairs of values from $\vec{x}$ and $\vec{y}$. For vectors $\vec{x}$ of rank n and $\vec{y}$ of rank m, this will result in a matrix of dimensions $n \times m$.

$$\vec{x} \otimes \vec{y} = \sum_{i,j} c_i^x c_j^y (\vec{n}_i \otimes \vec{n}_j) \tag{2.7}$$

A further refinement of this model was proposed by (Clark and Pulman, 2007), who represent words by combining their context vectors with vectors describing their grammatical role. For example, the sentence *"John drinks strong beer quickly"* is represented by equation 2.8, where constituent words stand in for their context vectors, while *subj*, *obj*, *adj* and *adv* stand in for the words' grammatical role type vectors. The ordering in the equation is a representation of a parse tree.

$$drinks \otimes subj \otimes John \otimes obj(beer \otimes adj \otimes strong) \otimes adv \otimes quickly \tag{2.8}$$

Albeit these models resolve the bag-of-words issue issue presented by the mixture models, they have one crucial flaw - that of space. Given a vector $\vec{n}$ of size $\|\vec{n}\|$, an m-word phrase has a size of $\|\vec{n}\|^m$ in the former model, and even more in the latter. Since the size of most distributional vector representations are well into the several hundreds, this approach is impractical for most real-world uses.

One of most popular solution for this issue is a mathematical function called *circular convolution*, defined by equation 2.9, introduced as an approach for solving this problem by (Plate, 1991). An example of this operation is given by equation 2.10. By using this technique, the combination of vectors is reduced to the size of the original vectors.

$$\vec{t} = \vec{x} \circledast \vec{y}, \quad t_i = \sum_{k=0}^{n-1} x_k y_{(j-k \pmod n)}, \quad \forall i \in \{0, \ldots, n-1\} \tag{2.9}$$

However, while solving the issue of space, circular convolution and other similar approaches are forms of noisy compression, and introduce noise into the data. Another issue with this approach is that circular convolution is also a commutative operation, reintroducing the bag of words problem.

$$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \tag{2.10}$$

$$\begin{aligned} \vec{t} &= \vec{x} \circledast \vec{y} \\ t_1 &= x_0 y_0 + x_1 y_2 + x_2 y_1 \\ t_2 &= x_0 y_1 + x_1 y_0 + x_2 y_2 \\ t_3 &= x_0 y_2 + x_1 y_1 + x_2 y_0 \end{aligned}$$

2.2.3. Tensor-based Models

All of the models outlined earlier, both those based on simple vector operations and those based on the tensor product, share a potentially undesirable trait – they treat all of the constituent words in the same way and represent them in the same space, ignoring

their roles in the sentence. In an adjective-noun or adverb-verb pair, both constituents reside in the same space and contribute equally to the resultant representation. However, relational words such as adjectives and adverbs are intuitively different – they *modify* the other words, acting upon functions upon them. Based on this intuition, an approach has been developed and formalized (Coecke et al., 2010) to represent such relational words as *higher-order* tensors (e.g., matrices or higher) acting as *linear maps* upon the arguments, which are *lower-order* tensors (e.g., vectors).

This approach is based on *tensor contraction*, a generalization of matrix multiplication, and is denoted with the $\times$ symbol, and relies on the *Choi-Jamiołkowsky isomorphishm*, expressed in generalized form for a multilinear map (a function with more than one argument) by equation 2.11. It states that every linear map from finite-dimensional Hilbert spaces $V_1, \ldots, V_k$ stands in a one-to-one correspondence with a tensor residing in the tensor product space $V_1 \otimes \cdots \otimes V_k$. The order of the tensor is, in general, equal to the number of arguments the modifier word would take, plus an additional one order for carrying the result. An adjective, as an unary function, is a tensor of order 2 (a matrix), while a transitive verb is a tensor of order 3. The result of combining two tensors of orders n and m through tensor contraction is a new tensor of order $n + m - 2$. Therefore, combining a noun (tensor of order one) and an adjective (tensor of order two), the resultant phrase is represented by a tensor of order one, a vector.

$$f : V_1 \to \cdots \to V_j \to V_k \cong V_1 \otimes \cdots \otimes V_j \otimes V_k \tag{2.11}$$

Tensor-based models provide a solution to the bag-of-words issues of mixture models and the circular convolution approach. Furthermore, they respect difference between word roles in a sentence, adhering to both intuition and formal semantics. Finally, unlike the general tensor product model, they do not suffer from the space complexity issues, since the vector contraction operations reduces the final result into a lower-order semantic space. However, it does have its issues – most notably it requires the mapping tensors to be constructed for all such relational (function) words. This process involves either a lot of manual labor, or an algorithm capable of constructing these mappings from another sufficiently detailed corpus or data about text. This problem is far from trivial, and presents one of the most important open issues concerning this approach. This issue restricts the application of this model to use on merely a finite number of well-defined sentence types (e.g., adjective-noun pairs) (Kartsaklis, 2014). Another issue is that they rely on parse tree structures. Therefore, they cannot be used for units

larger than sentences. This approach, and similar ones, have, however, been used with very encouraging results on examples where only such a limited set of function words and constructs needed to be encoded (Baroni and Zamparelli, 2010).

2.2.4. Deep Learning Models

With the recent rise of *deep learning* techniques in machine learning (LeCun et al., 2015), these methods have also been applied to the issue of compositionality of distributional models. These models are typically forms of neural networks that use multiple layers of representation to model the concepts, where shallower layers represent lower-level concepts, and higher-level concepts are derived by building them up from those representations, in the deeper levels of the network. These models are typically trained using the *backpropagation* algorithm to learn the weights of the neural network. This algorithm can be very time-consuming and cannot guarantee optimality. On the other hand, the deep neural networks benefit from their non-linearity and a larger number of layers, allowing them to learn a broader range of functional mappings than the simple vector combinations or tensor-based approaches.

Various types and architectures of neural networks have been used to model compositionality in a broad set of semantics-based tasks, with promising or even state-of-the-art results. Socher et al. (2010, 2011, 2012) use recursive neural networks to produce representations of variable-length sentences for a variety of tasks, while Kalchbrenner and Blunsom (2013a,b) use convolutional neural networks for discourse modeling for use in automatic translation and identifying dialogue segments in text, among others. While these models produce high-quality results, they are tuned specifically for the given task and do not perform as well in other tasks.

An alternative approach is offered by Le and Mikolov (2014), who introduce the *paragraph vector*, a method of unsupervised learning that likewise results in fixed-length vectors, but aims for generalized representations.[1] A noted weakness of this model is that it needs to perform an inference step at prediction time, employing gradient descent to compute the paragraph vector for the new input while using the learned parameters for the neural network.

Kiros et al. (2015) present the *skipthought* model, which attempts to distance itself from the composition function, and instead attempt to model a loss function as to allow a sentence to be predicted from the context of its surrounding sentences, abstracting

[1] Despite the name, the method works for various chunk sizes of text, from phrases, through sentences, to entire documents.

the skip-gram model to the sentence level. They use corpora of contiguous texts to model sentences as triples (s_{i-1}, s_i, s_{i+1}), where the elements are the preceding, current, and next sentence, respectively. Then then use an encoder-decoder model, with the encoder mapping words to sentence vectors, and the decoder being used to generate surrounding sentences. The decoder and encoder are both *recurrent neural networks* (RNNs), with the encoder using a *gated recurrent unit* (GRU) (Chung et al., 2014) activation function for the encoder, and a conditional GRU for the decoder. Additionally, they learn a mapping from the word embedding vector space to the vector space of the sentence vectors, allowing them to process even previously unseen words. They use this model on a wide variety of tasks, using only linear classifiers and no additional fine-tuning, and achieve results comparable to state-of-the-art methods on all tasks, showing the model to be robust. The one notable issue with the model is that it is quite slow to evaluate, and especially train, partly due to its use of the backpropagation algorithm as a means of training. The behaviour of the encoder is described by the following equations.

$$\vec{r}_t = \sigma(\vec{x}_t \mathbf{W}_r + \mathbf{U}_z \vec{h}_{t-1})$$
$$\vec{z}_t = \sigma(\vec{x}_t \mathbf{W}_z + \mathbf{U}_z \vec{h}_{t-1})$$
$$\hat{h}_t = \tanh(\mathbf{W}\vec{x}_t + \mathbf{U}(\vec{r}_t \odot \vec{h}_{t-1})$$
$$\vec{h}_t = (1 - \vec{z}_t) \odot \vec{h}_{t-1} + \vec{z}_t \odot \hat{h}_t$$

Values are given with index t, which indicates a discrete time unit, updated every time the next word of the sentence is introduced to the encoder. r_t is the value of the reset gate at time t, z_t the value of the update gate, while $\hat{h}_t$ is the proposed state update at that time, with h_t being the final value. Matrices $\mathbf{W}$, $\mathbf{W}_r$ and $\mathbf{W}_z$ represents the weights connecting the hidden, reset, and update gates to the input layer, while matrices $\mathbf{U}$, $\mathbf{U}_r$, and $\mathbf{U}_z$ represent their connections to the hidden layer of the architecture. Finally, the function σ is the logistic sigmoid, normalizing the values to the range $(0, 1)$.

The decoder represents the currently encoded steps in the hidden state h_t. That means that for a set of words $w_1, w_2, \ldots, w_N$, it is the final computed state, $h_{t(N)}$, that represents the entire sentence.

The decoder follows a similar set of equations, with the addition of bias values for equations for the reset, update, and proposed state gates. Unlike the encoder, of which there is only one, two decoders are present in the architecture – one for the previous sentence s_{i-1}, and another for the next sentence s_{i+1}. The decoders do not share any

of the parameters in the equation, except for a vocabulary matrix $\mathbf{V}$, which represents a distribution of words, and is connected to the hidden state of the encoder-decoder architecture as a weight matrix.

3. Modeling the Croatian Language

To employ distributional and compositional distributional semantic models for a language, in this case the Croatian language, it is necessary to select and train the distributional models, as well as choose a method or methods of composition. For the purposes of this thesis, the distributional model used was the continuous bag-of-words model and the `word2vec` implementation,[1] which was already shown to achieve state-of-the-art results for Croatian (Zuanović et al., 2014). Additionally, one of the selected models for computing compositional distributional semantic was the Skip-thought model (Kiros et al., 2015), which requires additional training on a corpus after a CBOW or skip-gram model has been trained.

3.1. Corpus Preparation

To train such deep learning models, a sufficiently large and varied corpus was required. Several such corpora exist for Croatian, such as the *hrWaC* (Ljubešić and Erjavec, 2011), or the filtered *fhrWaC* (**?**). However, unlike the word2vec library, the skip-thought implementation is quite slow to train with corpora of this size. Instead of further filtering one of these corpora to reduce their size further, a different corpus was used – the contents of the Croatian Wikipedia. A dump of the entire Wikipedia contents was taken on the 1st of January 2017 and was used in its entirety.

The corpus was pre-processed before training the models. Firstly, all of the documents inside the corpus are represented in the Extensible Markup Language (XML), and contain not only the text of the articles themselves, but also various metadata such as links and tables of contents. This data was stripped, leaving only the raw tex, which was then transformed into lowercase form. Additionally, the Croatian language is much more morphologically complex than English, for which the aforementioned models have been developed. To avoid low appearance counts for individual lexemes, the words in the corpus were lemmatized using the CST lemmatizer (Jongejan and

[1] Available at `https://code.google.com/archive/p/word2vec`

Dalianis, 2009) and the existing rules for Croatian (Agic et al., 2013). The resulting corpus is stored as a raw textual document, where each article of the original Wikipedia dump is stored as a single row.

The resulting corpus consists of $113,219$ documents in the Croatian language of varied length, and altogether $3,9768,872$ word tokens, or $1,034,125$ unique tokens. The average number of tokens per document is 351.25, with the shortest document containing 50 tokens, and the longest containing as much as $32,845$ tokens.

It is worth noting that the Croatian Wikipedia is not vetted for grammatical correctness, and contains mistakes in spelling, punctuation, and capitalization. While this is unlikely to affect the semantic space produced by the CBOW model, a significant percentage of errors could affect the sentence modeling by the skip-thought model.

3.2. Model Training

The CBOW model was trained directly on the resulting Croatian Wikipedia corpus, using the `gensim` library[2] that provides a wrapper around the `word2vec` library for the `Python` programming language. The training was performed using the default settings of the library: a vector size of 100, a window size of 5, and a minimum word frequency of 5, with training lasting for 5 iterations.

Unlike the `word2vec` library, the skip-thought implementation[3] requires additional preprocessing. Namely, it requires that a pre-trained CBOW or skip-gram model be provided, as well as a dictionary of all words appearing in the text used to train the skip-thought model. The distributional model used was the CBOW model trained on the Croatian Wikipedia, which was also used to construct the dictionary. It should be noted that this did not have to be so – as mentioned before, the skip-thought model can map words from the word embedding semantic space to the skip-thought encoder semantic space, and may freely be trained on a different corpus. In this case, that feature of the model was not used, hence it would be unable to process previously unseen words (as the CBOW model is not capable of this behavior). Instead, unseen words would simply be skipped by the model resulting from the training process on this dataset.

The default settings were used for skip-thought training, which means that the vector size was 620, 2400 GRU units were used, and using a maximum word length of

[2]Available at `https://radimrehurek.com/gensim/`

[3]Available at `https://github.com/ryankiros/skip-thoughts`

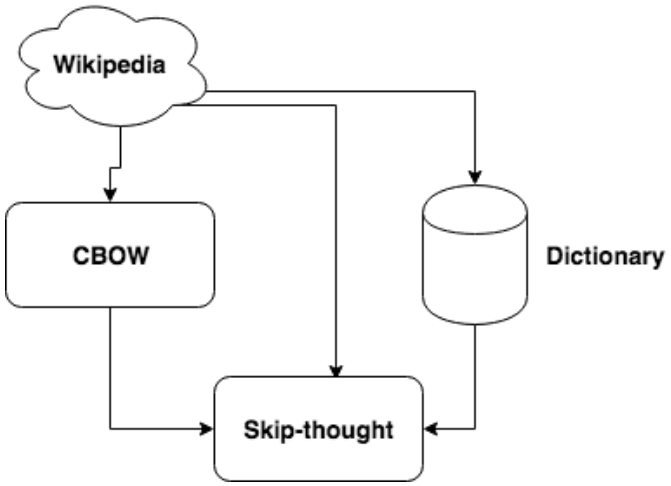


Figure 3.1: Graphical representation of model training dependencies

30 characters. Due to time constraints, the model was trained for only one epoch.[4] For comparison, the model for the English language was trained on three epochs of the training algorithm, on a dataset that was also considerably larger in both number of documents and average document length – for purposes of comparison, the English Wikipedia has $5,325,852$ articles, and over $2,900,000,000$ word tokens, as of the writing of this thesis. This *might* not be overly significant for the resultant model, as only the first half of the first epoch of training resulted in a noticeable consistent lowering of the cost of the training. However, it is impossible to know or prove this assumption without having let the training proceed further, as the process is stochastic. This means that the outcome was not certain, but also that the exact process likely cannot be replicated.

As was mentioned in subsection 3.1, the corpus on which was trained was lemmatized before the training process was run.[5] Therefore, the models do not reflect all of the morphological richness of the Croatian language, instead choosing to focus on the distribution of the words themselves. If one wishes to make use of the syntactic structure of the language, this data must be provided as features in addition to the vectors yielded by this model.

A simple graphical overview of the modeling process is given in Figure 3.1, with arrows showing which resource was used in building which other resource or model, and each resource or model annotated with its name or description.

[4] In actuality, more than one epoch, but only a single epoch was completed fully.

[5] The corpus is also made available in the pre-lemmatized form with the thesis.

4. Experimental Evaluation

To evaluate the trained models, a number of experiments are performed, for both supervised and unsupervised machine learning tasks. For all experiments, the base distributional model of semantics used is the CBOW model trained on the Croatian Wikipedia corpus. In all cases, the experiments use the following four compositional models:

1. Vector addition ($+$),
2. Elementwise vector multiplication ($\odot$),
3. Vector averaging (avg),
4. The skip-thought model.

In all cases except for the sentence similarity experiment in subsection 4.3, the result of the composition is the only feature used, allowing for an evaluation of the raw models.

4.1. Topic Classification and Clustering

In this experiment, a preliminary evaluation of the models is made on a relatively simple task – given three different topics taken from *Forum.hr*, a large Croatian public forum,[1] the goal is to match the posts from these topics to the topic they originate from. The experiment consists of two approaches: one uses supervised learning methods to match the posts to their respective topic, while the other uses unsupervised learning methods and evaluates the results. The results are then compared to simpler baselines, and a detailed error analysis is performed.

[1] `http://www.forum.hr/`

4.1.1. Dataset

For this task, three topics were selected. All of the topics were from the political sub-forum, and are highly polarized debates written in the Croatian language, and are among the longest topics on the forum in number of posts. The three topics are:

1. Debate about the education reform,[2]
2. Debate about gay marriage in Croatia,[3]
3. Debate about Cyrillic alphabet signs in the city of Vukovar.[4]

The topics were all pre-processed in the same way. Not all of the steps might be relevant, since a detailed analysis of how these pre-processing steps influence the results of the experiment was not performed. No effort was made to remove personal names, usernames, or foreign language words from the corpus, as this would prove very time-consuming in this type of weakly structured discourse. The pre-processing steps were:

1. HTML content was flattened to a plain textual format,
2. Quotes of previous posts were removed,
3. All links were removed,
4. All digits were removed,
5. All words were transformed into lowercase,
6. Punctuation and related symbols were removed,[5]
7. All whitespace was flattened into single spaces,
8. The tokens were lemmatized,
9. Posts shorter than 25 characters were removed from the dataset.

[2] `http://www.forum.hr/showthread.php?t=920734`
[3] `http://www.forum.hr/showthread.php?t=777241`
[4] `http://www.forum.hr/showthread.php?t=756655`
[5] This step was skipped when feeding the data into the skip-thought model.

After processing the corpus in this way, the result contained $21,284$ posts, of which $3,911$ (18.38%) belong to the school reform topic, $10,356$ (48.66%) to the gay marriage topic, and the remaining $10,017$ (47.06%) to the Cyrillic alphabet signs category. The average document contains 69.0 word tokens, with the minimum being 3 and the maximum being $1,605$. The average word tokens per topic are 100.3 for the school reform topic, 64.9 for the gay marriage topic, and finally 59.8 for the Cyrillic alphabet signs topic. An attempt was not made at this point to filter out off-topic posts from the dataset, although many could be considered such.

4.1.2. Evaluation and Results

After processing the corpus in the described way, four features sets were computed from them for each document using the composition methods listed earlier ($+$, $\odot$, avg, and skip-thoughts). These features were used in isolation, without combining them with any additional information about the texts.

Given these features sets, two experiments were performed – one using supervised learning algorithms, and the other using unsupervised learning. In the first experiment, an attempt is made to classify the posts into a category corresponding to the topic they originated from; In the second, several clustering algorithms were used to observe natural clustering in the dataset and compare it with the true alignment between the posts and topics they belong to.

Post Classification

For the first task, three-way post classification, a SCIKIT-LEARN (Pedregosa et al., 2011) implementation of a support vector classifier was used. A grid search was used to select the kernel optimize the hyperparameters (C and γ, for the rbf model), through nested cross-validation on the training set. The entire dataset was split into a training set (75%) and a test set (25%).

To put the results of the evaluation into context, they were compared to two simpler baselines: a majority baseline (relatively low due to the balanced size of the two larger topics, see 4.1.1), and a naïve Bayes classifier, using a bag-of-words representation of the posts as a feature. The F1-score (micro) was used to compare the results of these methods and the baselines.

The results are given in Table 4.1 for evaluation on the training set, and in Table 4.2 for the test set. The additive model was the most successful model, with the averaging model equaling it in performance on the test set, and nearly equaling it on the training

Model (SVM kernel)	F1-Score
Majority Baseline	0.45
BOW Naïve Bayes	0.65
Vector Addition (rbf)	**0.83**
Element-wise Vector Multiplication (linear)	0.43
Vector Averaging (linear)	0.82
Skip-thoughts (linear)	0.64

Table 4.1: Post classification results – training set

Model (SVM kernel)	F1-Score
Majority Baseline	0.43
BOW Naïve Bayes	0.75
Vector Addition (rbf)	**0.80**
Element-wise Vector Multiplication (linear)	0.42
Vector Averaging (linear)	**0.80**
Skip-thoughts (linear)	0.59

Table 4.2: Post classification results – test set

set. The other two models, skip-thought and the multiplication model, underperform significantly, neither one managing to beat the naïve Bayes. Furthermore, the multiplication model does not even manage to perform better than the majority baseline.

It is also interesting to note that all of the models, except for the skip-thought model, perform nearly identically, in terms of F1-score, on the test set in comparison to the training set.

For each model, a random sample of the incorrect classifications was taken to study the type of document the errors occurred for. The errors for the best-performing models, the additive and averaging model, are nearly identical, judging from the taken sample, and can be roughly grouped into three categories:

1. Document for which it is impossible to determine the category for without additional context,

2. Documents written with incorrect Croatian grammar,

3. Very long documents.

The first category includes documents that typically have no content regarding the topic at hand, but are simply agreements or disagreements with some previous post, or general statements. For example, statements "I agree with everything you said!" or "You have no idea what you're talking about". The second includes posts that do not adhere to standards of literacy maintained on the training corpus, the Wikipedia dump, or contain many foreign words or abbreviations. It is probable that after eliminating these words, the remainder of the text was simply not modeled properly in the semantic space. Finally, very long documents also seem to be difficult to classify when using these combinator functions. This is likely due to the fact that these two functions model the texts as an averaged mixture of the word representations, and the semantic information was lost when dealing with such a large number of tokens.

The skip-thought model also seems to struggle with very long documents. Interestingly enough, it also seems to have difficulties classifying short posts, even when the target class would be obvious to a human annotator, and would be successfully annotated by the simpler additive or averaging models. Of the medium-length documents (ranging from 3-5 medium-sized sentences, judging from the sample taken), there was no noticable pattern in the incorrectly classified documents. This could be due to the different type of discourse when compared to the corpus it was trained on – as a complete compositional distributional semantic model, the skip-thought model would be more sensitive to this than the merely distributional semantic model presented by the

CBOW. An alternate plausible explanation could simply be that the model was not trained for long enough, as was noted earlier.

Finally, observing the sample, no regularity was observed for the multiplication model. It would appear that it is simply insufficient for modeling texts for this task.

Post Clustering

To investigate natural similarities between the computed features, and how they map to the categories, an experiment in unsupervised learning was performed as well. The posts were clustered using several clustering algorithms, and were compared to the true grouping of the posts using several metrics.

The clustering methods used were K-means clustering (MacQueen et al., 1967) and agglomerative hierarchical clustering. The number of clusters for K-means, as well as the cut-off point for the hierarchical clustering, was set to 3, matching the true number of topics. For the latter model, all combinations of three linkage functions (average, Ward, and complete) and two distance metrics (cosine and Euclidean) were tested. For each combination method, experiments with both distance metrics are reported, but only in combination with the most successful linkage function. Initial centroids for K-means were chosen randomly.

The success rate of the clustering experiment is expressed in three metrics for similarity of data clustering, for all of the cases: the adjusted Rand index (ARI) (Hubert and Arabie, 1985), a measure analogous to accuracy, but insensitive to the exact label, and adjusted for chance groupings of elements. The adjusted Rand index scales from -1 to 1, where 0 signifies random labeling, 1 is a complete match, while negative values are worse than random labeling; The completeness score (c) (Rosenberg and Hirschberg, 2007), which measures whether all data points of a given class were assigned to the same cluster; The homogeneity score (h) (Rosenberg and Hirschberg, 2007), which measures whether a cluster contains only data points from a single class.

The results are given in Table 4.3. It can be observed that all the results tend towards 0, signifying a random clustering in the case of the adjusted Rand index, and a mismatch between the true structure and the inferred structure in the other metrics. While some perform better than others in certain cases, none does so consistently. Furthermore, all of the results are barely, if at all, better than complete random clustering. From this, it can be inferred that the posts in this dataset do not have some internal structure in this semantic space that would significantly differentiate them from the posts belonging to other categories, or at least not a structure corresponding to that

Method	ARI	c	h
Vector addition			
K-Means	0.04	0.03	0.02
AHC (Euclidean, average)	0.02	0.05	0.01
AHC (Cosine, average)	0.04	0.03	0.02
Element-wise Vector Multiplication			
K-Means	0.00	0.08	0.00
AHC (Euclidean, complete)	0.00	**0.08**	0.00
AHC (Cosine, average)	0.00	**0.08**	0.00
Vector averaging			
K-Means	0.04	0.03	0.03
AHC (Euclidean, Ward)	**0.06**	0.07	**0.06**
AHC (Cosine, average)	0.05	0.03	0.05
Skip-thought			
K-Means	0.00	0.02	0.02
AHC (Euclidean, Ward)	0.00	0.04	0.02
AHC (Cosine, average)	0.00	0.04	0.03

Table 4.3: Evaluation of post topic clustering

expected by these clustering algorithms. It is interesting to note that this did not prevent the support vector machine based models from detecting a connection between the categories, even though most of them used non-linear kernels.

4.2. Stance Classification

Stance classification (Somasundaran and Wiebe, 2010) is one of the typical semantic analysis tasks. As such, it was selected for further evaluation of the trained models and the composition methods. The same combinator functions are used for two experiments with stance classification – in the first experiment, the posts are classified into one of three categories (positive, negative, and neutral); In the second experiment, the neutral labels are eliminated, and an attempt is made to differentiate between solely positively and negatively labeled posts.

4.2.1. Dataset

Of the three topics used as the dataset in the previous example (see Section 4.1), the debate concerning the school reform was selected, due to it being the shortest, for the reason of reducing the amount of manual annotation needed. Additionally, posts shorter than ten words were removed, after noticing that most of these posts are non-informative, and removing them would reduce the size of the dataset by 13.6%.

To obtain the gold standard, this topic was given to five annotators to manually annotate in one of three categories:

- *Positive* – The post supports this specific attempt to reform education,
- *Negative* – The post opposes this specific attempt to reform education,
- *Other* – The post is off-topic, neutral (e.g., simply informative), or the stance cannot be determined from its contents.

The corpus was split for annotation in this way: 200 posts were annotated by all of the annotators, while every annotator also got an even share of the remaining posts to annotate alone (up to a difference of one due to rounding). As a whole the annotators spent approximately 30 hours annotating. This way, each annotator annotated either 942 or 943 posts. For the first 200 posts, the gold standard annotation was determined by the majority of annotations. In cases where this was not possible (i.e., there was a tie), the annotators were asked to re-examine the posts and re-classify them; This was

Scoring method	IAA
κ	0.38
α	0.37
F1-score	0.80

Table 4.4: Inter-annotator agreement for stance classification dataset

Scoring method	IAA
κ	0.45
α	0.43
F1-score	0.88

Table 4.5: Inter-annotator agreement without Annotator 5

required for six posts. Finally, in one case, the tie was still not resolved, and the gold standard annotation was determined by the author.

Inter-annotator agreement (IAA), shown in Table 4.4, was computed on the 200 posts annotated by all annotators, using three metrics: Fleiss' kappa score (Fleiss, 1971), Krippendorff's alpha score (Hayes and Krippendorff, 2007), and an averaged F1-score computed by pretending that the first annotator was the gold standard and the other annotators' classifications were scored against it.

We can use these metrics to estimate the difficulty of the task and the quality of the annotations. There is no standard guideline for estimating the quality from the kappa score, but certain scales have been proposed (Landis and Koch, 1977). Where Krippendorff's alpha score is concerned, the minimum acceptable coefficient value varies, but has been suggested to limit at 0.667 (Krippendorff, 2004). When considering both metrics, the data can be considered unreliable, or the task very difficult even for the human annotator. Further investigation into the content and the differences suggests that it is a combination of both.

When studying pairwise comparisons, annotator 5 stands out as quite different to other annotators. An adjusted metric without taking this annotator's annotations into consideration is given in Table 4.5. However, even with the notable improvement, these results are still sub-par by most metrics.

The other difficulty is the observed objective difficulty of classifying some of these posts. Furthermore, large number of posts are either unclear without further context, or demand outside knowledge from the annotator (i.e., the name of the actors in the

Model (SVM kernel)	F1-Score
Majority baseline	**0.8**
Vector Addition (rbf)	**0.8**
Element-wise Vector Multiplication (linear)	**0.8**
Vector Averaging (linear)	**0.8**
Skip-thoughts (linear)	**0.8**

Table 4.6: Three-way stance classification results

events being discussed). This influences the ability of the annotator to make correct decisions. Finally, one final difficulty is related to the events of the topic at hand – during the progression of the debate, a change in the governing party in Croatia introduced radical changes to the reform process, changing the way the discussion was led. This might have led to many false positives and false negatives.

After the annotation process was concluded, the results were such: of the 3377 annotated documents, 2802 (83.0%) were annotated as neutral, 328 (9.7%) as positive, and 247 (7.3%) as negative. In the test set, 80% are neutral, 11.4% are positive, and 8.4% are negative.

4.2.2. Evaluation and Results

Same as in the previous experiment, the listed four combinator functions were used – $+$, $\odot$, avg, and skip-thoughts, without any additional features. Two related experiments were then performed on the prepared data. Firstly all of the dataset entries were used and three-way classification was performed, to match the annotations to the full scale of the annotations. Secondly, the neutral entries were filtered out of the dataset and two-way classification was performed between the negative and positive entries. In both cases, a grid search was used, with cross-validation on the training set, to select the best kernel and hyperparameters (C, γ) for the SVM.

The results of the first experiment are given in Table 4.6, with the micro F1-score reported. It can be noted that all of the results are the same and match the majority baseline. It is obvious that all of the models simply match every entry to the majority class, and do not manage to exceed its performance. As such, no further error analysis can be performed.

Motivated by the lack of information gained from the first experiment, the second experiment was performed, where no neutrally-labeled entries were used. Its results

Model (SVM kernel)	F1-Score
Majority baseline	57.0
Vector Addition (rbf)	57.0
Element-wise Vector Multiplication (linear)	59.4
Vector Averaging (linear)	59.3
Skip-thoughts (rbf)	**60.9**

Table 4.7: Two-way stance classification results

are given in Table 4.7. The additive model again performs in the same manner, matching every entry to the majority class (positive in this experiment). While the other models do outperform the baseline, the results are not much better. Given a sample taken of the errors for each model, the model based on element-wise vector multiplication appears to be balanced in its predictions, matching and mismatching entries from both categories. The averaging-based model evaluates similarly to the vector addition model, matching nearly all of the positive examples, but also managing to match some of the negative examples. Unlike in the other experiments, the skip-thought model is the best performer in this case, if not by much. Like the multiplicative model, it seems not to favor any of the two classes considerably, offering a balanced grouping of errors by class.

4.3. Sentence Semantic Similarity

The final experiment evaluates *semantic text similarity* (STS) (Corley and Mihalcea, 2005) between pairs of sentences. In this experiment, sentences are paired up and their semantic similarity is estimated on a scale as a decimal number from 1.0 (meaning *highly unrelated*) to 5.0 (meaning *highly related*).

The composition methods used in previous tasks are compared to the TakeLab STS system (Šarić et al., 2012), which scored in the top five systems in the SemEval-2012 Semantic Textual Similarity Task (Agirre et al., 2012). This system was adapted for use with Croatian language texts, as described in Section 4.3.2.

4.3.1. Dataset

The dataset used for this task is based on the *SMTNews* corpus used in the SemEval-2012 Semantic Textual Similarity Task (Agirre et al., 2012). Due to time constraints, only a small section of the task was taken for this experiment – the first 100 sentence pairs. The sentence pairs were translated into the Croatian language in this way:

- The left-hand-side sentence of the pair was translated manually, in such a way that if the same sentence appeared multiple times in the corpus, the translation was made sure to be consistent;
- The right-hand-side sentence was translated automatically using the Google Translate online translation service.[6]

The automatic translation of the right-hand-side was done partly to reflect the make-up of the original dataset, in which the right-hand-side sentences were the result of translating to another language, and then back to English. The other reason was to reduce the effort of translation.

The annotations were kept unchanged from the English version of the dataset. As such, an error was likely introduced, as the translated pairs would almost certainly be annotated different by a human annotator. As such, the annotations can be considered a *silver standard* at best.

4.3.2. Preparing the TakeLab STS model

The TakeLab STS model was crafted specifically for the sentence similarity estimation task. Its *simple model* is made freely available online.[7] However, it was trained using English language corpora, as well as using the English WordNet for computing its knowledge-based word similarity features. As such, it needed to be adjusted to use Croatian language resources to be used for this task.

Originally, the dictionary and the word vectors were obtained from the English language Wikipedia and the New York Times Annotated Corpus (NYT) (Sandhaus, 2008), using latent semantic analysis (LSA) to obtain the vectors themselves. These corpora were replaced with the Croatian language Wikipedia outlined in Chapter 3, using GENSIM's implementation of LSA.[8] The code was adjusted to work with only one source of dictionary words, instead of the previous two.

[6] `http://translate.google.com/`
[7] `http://takelab.fer.hr/sts/`
[8] `https://radimrehurek.com/gensim/models/lsimodel.html`

The English WordNet resource, used in the model as a knowledge source for word similarity, was instead replaced with the Croatian variant of the same resource, the Croatian WordNet (CroWN) (Raffaelli et al., 2008).

4.3.3. Evaluation and Results

For the training of all models, the dataset was split into a training set (75%) and a test set (25%). For all of the systems, a grid search was performed using cross-validation. For the TakeLab STS system, the grid search was performed using the software attached with the system. For the other systems, the SCIKIT-LEARN implementation was used.

When using the TakeLab STS system, its way of extracting features was not modified. The vector combination systems used three features derived from the compositional distributed representation of the sentences – their vector sum, difference, and cosine distance.

The results were evaluated using Pearson's correlation (r) and the mean square error (MSE) metrics. The results are shown in Table 4.8, with the retrained TakeLab STS system used as a baseline. The TakeLab STS does not perform as well as it did for texts in English, producing results notably lower than even on that text in English. This may be due to the fact that the training set is several times smaller than the corresponding training set given to it in English, or a deficiency in the sources of words and knowledge.

Studying the errors made by this model, it can be seen that it consistently ranks sentences very high in similarity, mostly ranging between 4.0 and 5.0. Observing the correct annotations, it can be seen that that is indeed the range in which a majority of them occur, with only very few being lower than 3.0. Given that, it is most likely that the size and make-up of the training set are mostly to blame for the performance of this model.

The vector addition model again performed the best out of the tested composition models. Its results can be considered quite good, seeing how the results for the SMTNews corpus in the English language had a Pearson score of 0.61 (Agirre et al., 2012). However, the results are not directly comparable due to both the translated nature of the resources, and the fact that the dataset used in this example is only a subset of the SMTNews dataset. This would suggest that the task may have been made easier. Observing the errors on the dataset, it can be seen that this model predicts both high and low scores, but is typically a distance away (both above and below, without apparent

Model (SVM kernel)	r	MSE
TakeLab STS	0.36	0.49
Vector Addition (rbf)	**0.73**	**0.23**
Element-wise Vector Multiplication (linear)	0.10	0.55
Vector Averaging (linear)	0.39	0.45
Skip-thoughts (linear)	0.54	0.32

Table 4.8: Sematic sentence similarity evaluation results

rhythm) from the correct annotation.

The skip-thought model scores significantly lower in both measured scores, but manages to outperform the other two composition models. Its rating tends more towards the average score as an attempt to minimize the error, similarly to the TakeLab STS model, but are still more correct that the scores given by the vector averaging model, which scores in that manner almost exclusively, with no estimation in the entire test set being outside the $(4.1, 5.0)$ window.

Finally, the vector multiplication model does not manage to correlate significantly with the data at all, assigning the data only one of two score: 3.9 or 4.65, with apparently very little correlation to whether the score should truly be low or high. While completely hypothetical, it might be assumed that the representations were skewed by combining first multiplication, then adding and subtracting vectors, while the other two mixture models, using similar methods in their core, were more robust when applied to this process.

4.3.4. Demo Website

To demonstrate the sentence similarity task, a demo website was created, featuring the vector-addition-based model, which was the best-performing model for this task. The website was written in the `Python` programming language (version 2.7), using the `flask` web framework.[9] It is a simple traditional web application, which are single-tiered, with all the page constructing, application and business logic happening in the back-end. This is unlike many other modern applications, where the front-end layer (e.g., a mobile, desktop or JavaScript web application) contains complex logic of its own, such as business logic or page routing logic). The application can be rendered in

[9] urlhttp://flask.pocoo.org/

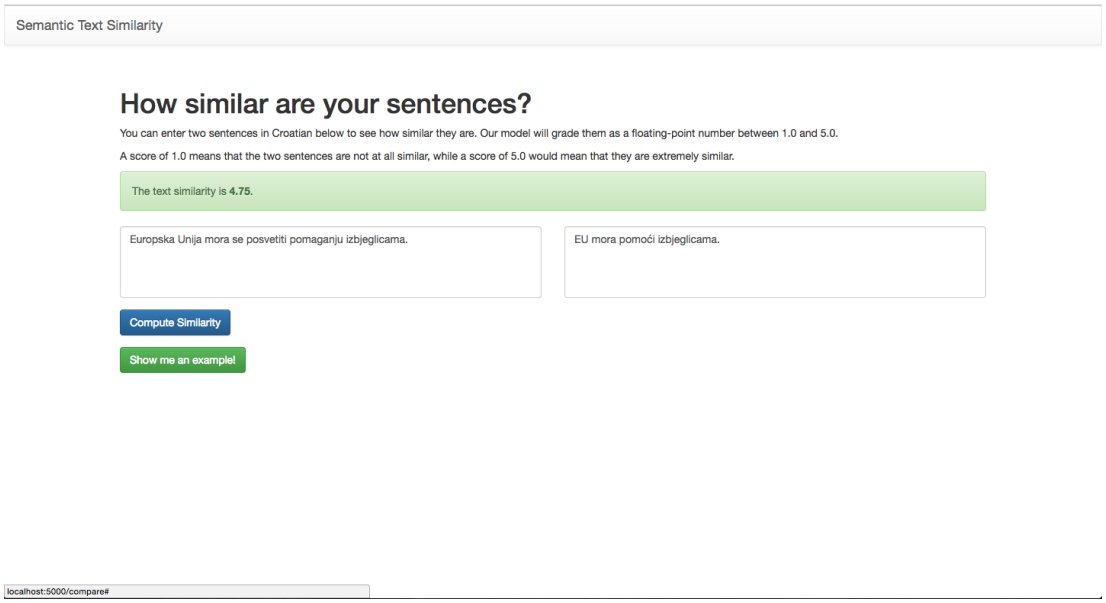


Figure 4.1: Desktop browser view of the application

all major modern browsers.

The application features a simple single-page web interface, whether the user is prompted to enter two sentences in the Croatian language and press a button, after which the similarity is computed and displayed. An additional button offers the functionality of entering and evaluating and example pairs of sentences from a predefined set. This example behavior mimics user action and makes it obvious how the system works.

The application is styled using the `Twitter Bootstrap`[10] Cascading StyleSheet (CSS) language library, and has been styled to be responsive, adjusting to the screen size of the viewing device for the best viewing experience. The desktop version can be seen in Figure 4.1, while the mobile view is preview in Figure 4.2.[11]

The application contains the sentence similarity evaluation model as a module within its own code. The module is loaded at start-time to prevent overly long response times, and evaluations are made separately for each request the browser makes to the server application.

[10] `http://getbootstrap.com/`

[11] Previews generated using Google Chrome device emulation view.

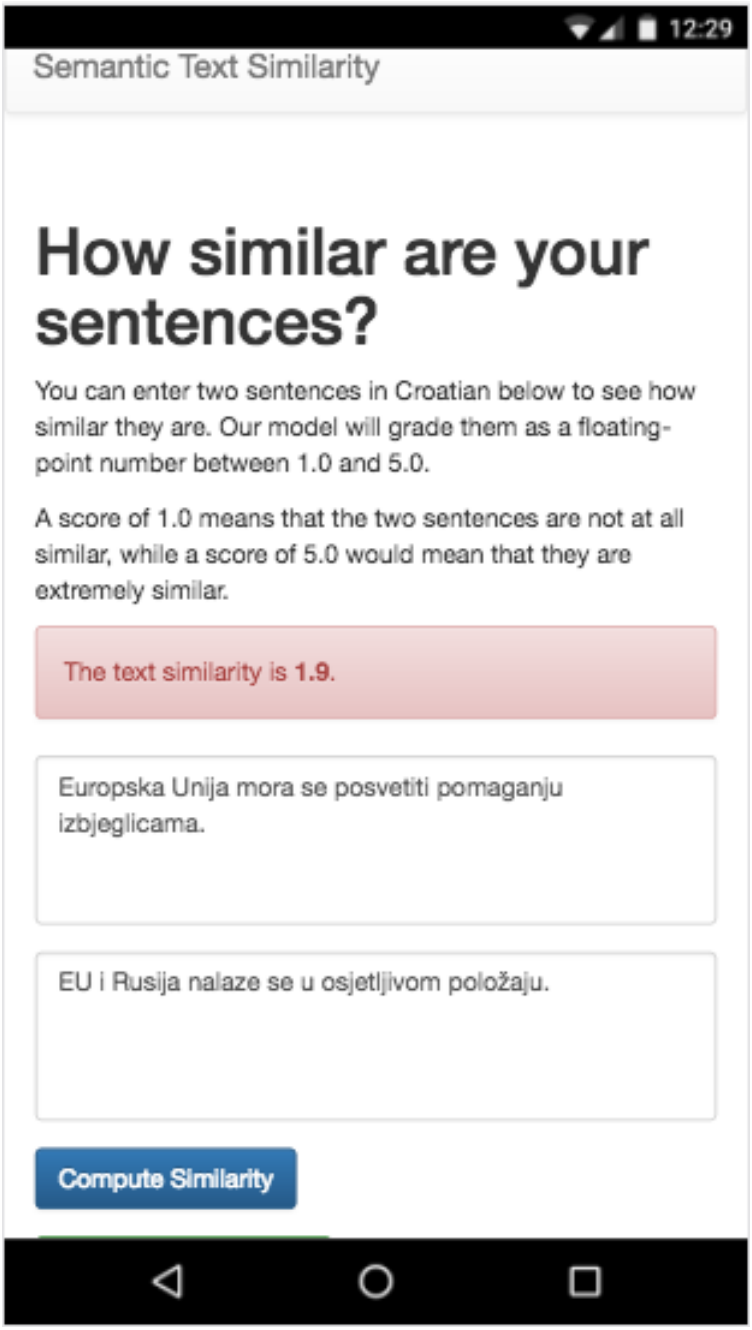


Figure 4.2: Mobile browser view of the application

4.4. Analysis

Four compositional distributional models were used, all based on the CBOW model as the basic source of distributional semantic word vectors. Three of these systems, the additive, multiplicative, and averaging model, were based on simple commutative vector operations, also called "mixture" combinators. The fourth was the skip-thought model, a state-of-the-art model of the English language, which has been retrained on a Croatian corpus.

Using these models, five experiments were performed, of them four in the domain of supervise learning, and one in the domain of unsupervised learning. These experiments are:

1. Post classification to corresponding topic of discourse;
2. Post clustering into topic clusters, comparison to natural topic groupings;
3. Three-way stance classification in online debate (*positive*, *neutral*, and *negative*);
4. Two-way stance classification in online debate (*positive* and *negative*);
5. Sentence similarity estimation.

Of these experiments, the clustering was by far the least successful – none of the models produced features that naturally clustered into anything alike the original structure. The experiment using the same dataset for classification was more successful, especially for the additive and averaging model. The other models did not show much success.

In the stance classification experiments, all of the models managed to match or outperform the baseline, but none by more than a few percent of the F1-score. When the neutral class was included, *all* of the models in fact matched the baseline, classifying every value into the majority class. This is the only experiment in which the skip-though model managed to outperform the other models, although not by much.

It is in the final experiment, sentence similarity, that some the models manage to achieve encouraging results.

In general, it can be observed that the skip-thought model has not achieved state-of-the-art results as it had in English. As mentioned in Chapter 3, one potential explanation for this is that the model was trained for a shorter period of time than the corresponding model of the English language. Another possible explanation is the difference in discourse type – while the model was trained on the Croatian Wikipedia,

the datasets on which tests were performed were quite different in type, namely online forum debates and sentences from news articles. Since this model learns sentences, it might not have adapted well to the Croatian language, it being relatively more complex in structure than English.

The other models show varied performance, with at least one model always being comparable to or better than the skip-thought model, even though they are mere bag-of-words representations, and were used as the sole features. This would seem to agree with the findings of Mitchell and Lapata (2010), who also find that simple mixture models perform reasonably well.

In all cases but potentially the last, none of the models have achieved what could be considered state-of-the-art results, even given the relative difficulty of some of the tasks. However, there has always been a model, as well as there always having been a simple *mixture model*, that was equal to or better than the more naïve baseline. Since the vector mixture models are simple and fast to prepare, it would not be unreasonable to use them in further experiments in combination with other features.

5. Conclusion

This thesis examined the field of compositional distributional semantics, and especially its application to semantic text similarity (STS) tasks. This area of research covers teks such as measuring text similarity (Corley and Mihalcea, 2005) and stance classification (Somasundaran and Wiebe, 2010). An overview was given of the field of compositional distributional semantics, and the corresponding compositional distributional semantic models, as well as the underlying field of distributional semantics.

The goal of this thesis was to investigate how modern distributional semantic models can be applied to semantic text similarity tasks on texts in the Croatian language. A corpus based on the contents of the Croatian Wikipedia was prepared, and used to train the continuous bag-of-words (CBOW) (Mikolov et al., 2013a) distributional model, and the compositional distributional model skip-thoughts (Kiros et al., 2015), both of which have achieved state-of-the-art results for such tasks in English.

Using these models, a series of experiments were performed for unsupervised and supervised learning in the field of natural language processing: post-topic classification, post-topic clustering, stance classification in online debates, and sentence similarity estimation. In the last of these, the TakeLab STS (Šarić et al., 2012) model, which has achieved encouraging results for this very task on English texts in the SemEval-2012 Task 6 (Agirre et al., 2012) was retrained for Croatian to be used as a baseline. Datasets were prepared for these experiments, and the results gathered, presented and analyzed.

The obtained results did not mirror the results achieved for English, in most cases not managing to significantly outperform the baseline. This is true even for the skip-thought model, which in most cases performed poorly even when compared to naïve composition models.

Although the models did not perform very well, certain obvious points of improvement present themselves. Firstly, the models were trained on the Croatian Wikipedia, which is an entirely different type of discourse than the one used in the experiment datasets. These results may be improved further by using a larger and more varied cor-

pus, such as the hrWaC corpus (Ljubešić and Erjavec, 2011). Secondly, it was noted that the training of the skip-thought model was abbreviated considerably when compared to the training for the English language. It is certainly possible that the results of the experiments would be much improved for this model if given further time to learn the sentence semantic space. Thirdly, this thesis does not perform experiments with many other mentioned composition models, such as circular convolution or paragraph vectors (Le and Mikolov, 2014). Investigation into application of these models could prove to be worthwhile. Lastly, this thesis used the CBOW model as the base distributional semantic model. It has been noted that the skip-thought model is slightly better at most tasks (Mikolov et al., 2013a). As such, small improvements might be obtained by swapping out these two models.

BIBLIOGRAPHY

Željko Agic, Nikola Ljubešić, and Danijela Merkler. Lemmatization and morphosyntactic tagging of Croatian and Serbian. U *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, stranice 48–57, 2013.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. U *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*, stranice 385–393. Association for Computational Linguistics, 2012.

Mark Andrews, Gabriella Vigliocco, and David Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological review*, 116(3): 463, 2009.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. U *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, stranice 1183–1193. Association for Computational Linguistics, 2010.

Giacomo Berardi, Andrea Esuli, and Diego Marcheggiani. Word embeddings go to italy: A comparison of models and training datasets. U *IIR*, 2015.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. U *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, stranice 546–556. Association for Computational Linguistics, 2012.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. Large language models in machine translation. U *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Citeseer, 2007.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Stephen Clark and Stephen Pulman. Combining symbolic and distributional models of meaning. U *AAAI Spring Symposium: Quantum Interaction*, stranice 52–55, 2007.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

Allan M Collins and M Ross Quillian. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2):240–247, 1969.

Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. U *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, stranice 13–18. Association for Computational Linguistics, 2005.

James Richard Curran. From distributional to semantic similarity. 2004.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukWaC, a very large web-derived corpus of English. U *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, stranice 47–54. sn, 2008.

Adriano Ferraresi, Silvia Bernardini, Giovanni Picci, and Marco Baroni. Web corpora for bilingual lexicography: a pilot study of English/French collocation extraction and translation. *Using Corpora in Contrastive and Translation Studies. Newcastle: Cambridge Scholars Publishing*, stranice 337–362, 2010.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

Gottlob Frege. Die Grundlagen der Arithmetik. *Eine logisch mathematische Untersuchung u'ber den Begrijfder Zahl. Breslau: Koebner*, 1884.

Justin Garten, Kenji Sagae, Volkan Ustun, and Morteza Dehghani. Combining distributed vector representations for words. U *Proceedings of NAACL-HLT*, stranice 95–101, 2015.

Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological review*, 114(2):211, 2007.

Zellig Harris. Mathematical structures of language. 1968.

Andrew F Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

Geoffrey E Hinton and Tim Shallice. Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological review*, 98(1):74, 1991.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Michael N Jones and Douglas JK Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, 114(1):1, 2007.

Bart Jongejan and Hercules Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. U *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, stranice 145–153. Association for Computational Linguistics, 2009.

Nal Kalchbrenner and Phil Blunsom. Recurrent convolutional neural networks for discourse compositionality. *arXiv preprint arXiv:1306.3584*, 2013a.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. U *EMNLP*, svezak 3, stranica 413, 2013b.

Dimitri Kartsaklis. Compositional operators in distributional semantics. *Springer Science Reviews*, 2(1-2):161–177, 2014.

Walter Kintsch. Predication. *Cognitive science*, 25(2):173–202, 2001.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. U *Advances in neural information processing systems*, stranice 3294–3302, 2015.

Klaus Krippendorff. Reliability in content analysis. *Human communication research*, 30(3):411–433, 2004.

George Lakoff. Linguistic gestalts. U *Papers from the... Regional Meeting. Chicago Ling. Soc. Chicago, Ill.*, svezak 13, stranice 236–287, 1977.

Thomas K Landauer and Susan 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.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, stranice 159–174, 1977.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. U *ICML*, svezak 14, stranice 1188–1196, 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Nikola Ljubešić and Tomaž Erjavec. hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. U *International Conference on Text, Speech and Dialogue*, stranice 395–402. Springer, 2011.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28 (2):203–208, 1996.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. U *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, svezak 1, stranice 281–297. Oakland, CA, USA., 1967.

Arthur B Markman. *Knowledge representation*. Psychology Press, 2013.

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

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

George A Miller. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429, 2010.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

Barbara Partee. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, 1995.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Tony Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. U *IJCAI*, stranice 30–35, 1991.

Ida Raffaelli, Marko Tadić, Božo Bekavac, and Željko Agić. Building Croatian wordnet. U *Fourth Global WordNet Conference (GWC 2008)*, 2008.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. U *EMNLP-CoNLL*, svezak 7, stranice 410–420, 2007.

Evan Sandhaus. The New York Times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

Christophe Servan, Alexandre Bérard, Zied Elloumi, Hervé Blanchon, and Laurent Besacier. Word2vec vs dbnary: Augmenting meteor using vector representations or lexical resources? *arXiv preprint arXiv:1610.01291*, 2016.

Steven A Sloman and Lance J Rips. Similarity as an explanatory construct. *Cognition*, 65(2):87–101, 1998.

Edward E Smith and Douglas L Medin. *Categories and concepts*. Harvard University Press Cambridge, MA, 1981.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216, 1990.

Jan Šnajder, Sebastian Padó, and Željko Agić. Building and evaluating a distributional memory for Croatian. U *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, svezak 2, stranice 784–789, 2013.

Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. U *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, stranice 1–9, 2010.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. U *NIPS*, svezak 24, stranice 801–809, 2011.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. U *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, stranice 1201–1211. Association for Computational Linguistics, 2012.

Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological online debates. U *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, stranice 116–124. Association for Computational Linguistics, 2010.

Mark Steyvers and Joshua B Tenenbaum. The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, 29 (1):41–78, 2005.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. TakeLab: Systems for measuring semantic text similarity. U *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, stranice 441–448, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S12-1060`.

Jan Šnajder and Petra Almić. Modeling semantic compositionality of Croatian multiword expressions. *Informatica*, 39(3):301, 2015.

Leo Zuanović, Mladen Karan, and Jan Šnajder. Experiments with neural word embeddings for Croatian. U *Proceedings of the 9th Language Technologies Conference*, stranice 69–72, 2014.

Application of Compositional Distributional Semantics for Semantic Text Similarity

Abstract

Modern approaches to compositional distributional semantics have revolutionized many areas based on semantic similarity in recent years. In this thesis, a survey of the field is given. Several of these modern models and approaches are selected and tried on a variety of supervised and unsupervised learning tasks in the field of semantics, on texts in the Croatian language, with attention given to semantic text similarity. A showcase web application is developed and presented.

Keywords: Compositional distributional semantics, distributional semantics, natural language processing, deep learning, vector mixture models, semantic similarity, Croatian language.

Primjena kompozicijske distribucijske semantike u zadatku semantičke sličnosti teksta

Sažetak

U zadnjih nekoliko godina, moderni pristupi kompozicijskoj distribucijskoj semantici donijelu su revoluciju u mnoga područja bazirana na semantičkoj sličnosti. U ovom je radu dan pregled tog područja. Izabrano je nekoliko modela i iskušani su na zadacima nadziranog i nenadziranog učenja u području semantike, na tekstovima na hrvatskome jeziku. Posebna pažnja dana je sematičkoj sličnosti teksta. Razvijena je i prikazana pokazna web aplikacija.

Ključne riječi: Kompozicijska distribucijska semantika, distribucijska semantika, obrada prirodnog jezika, duboko učenje, modeli mješanja vektora, sematička sličnost, hrvatski jezik.