IMPROVING WORD MEANING REPRESENTATIONS USING WIKIPEDIA CATEGORIES

L. Svoboda[*], *T. Brychcín*[†]

Abstract: In this paper we extend *Skip-Gram* and *Continuous Bag-of-Words* Distributional word representations models via global context information. We use a corpus extracted from Wikipedia, corpus where articles are organized in a hierarchy of categories. These categories provide useful topical information about each article. We present the four new approaches, how to enrich word meaning representation with such information. We experiment with the English Wikipedia and evaluate our models on standard word similarity and word analogy datasets. Proposed models significantly outperform other word representation methods when similar size training data of similar size is used and provide similar performance compared with methods trained on much larger datasets. Our new approach shows, that increasing the amount of unlabelled data does not necessarily increase the performance of word embeddings as much as introducing the global or sub-word information, especially when training time is taken into the consideration.

Key words: *Word2Vec, word embeddings, global information*

Received: November 23, 2018 **DOI:** 10.14311/NNW.2018.28.029
Revised and accepted: December 31, 2018

1. Introduction

Distributional word representation methods are based on the word-context co-occurence and comes from the principle known as *Distributional hypothesis*: "linguistic items with similar distributions have similar meanings" [9]. The idea that "a word is characterized by the company it keeps" was popularized by Firth [6]. There are studies that demonstrate theoretical roots in the psychological reality, linguistics, or lexicography [4]. The implication of Distributional hypothesis is that two words are expected to be semantically similar if they occur in similar context (they are similarly distributed across the text). This research area is often referred to as *distributional semantics*. With the rise of a massive and easily-accessible digital corpora, and the power of computers, it has become very popular in the last

[*]Lukáš Svoboda – Corresponding author; Faculty of Applied Sciences, University of West Bohemia, Department of Computer Science and Engineering, Czech Republic, E-mail: svobikl@kiv.zcu.cz

[†]Tomáš Brychcín; NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Department of Computer Science and Engineering, Czech Republic, E-mail: brychcin@kiv.zcu.cz

two decades. It suggests an efficient and practical method to induce the meaning. Models based on this assumption are denoted as *distributional semantic models* (DSMs).

DSMs, also known as "word space" or "distributional similarity" models learn contextual patterns from a huge amount of textual data. They typically represent the meaning as a vector which reflects the contextual (distributional) information across the texts [33]. The words are associated with a vector of real numbers. Represented geometrically, the meaning is a point in a k-dimensional space. The words that are closely related in a meaning tend to be closer in the space. This architecture is sometimes referred to as a *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).

Word-based semantic spaces provide impressive performance in a variety of NLP tasks, such as language modeling [2], named entity recognition [14], sentiment analysis [11], and many others.

Different types of context induce different kinds of semantic spaces. [28] and [21] distinguish *context-word* and *context-region* approaches to the meaning extraction. In this paper 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 the 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 does not incorporate any metadata which Wikipedia offers.

We combine both, the local and the global context to improve the word meaning representation. We use local-context *DSMs – Skip-Gram* (SG) and *Continuous Bag-of-Words* (CBOW) models [22], often denoted as a tool called *Word2Vec* and incorporate Wikipedia categories as a global context. We present several approaches to enrich word meaning representation with such kind of information via a joint training objective.

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

The paper is organized as follows. Section 2 puts our work into the context of the state of the art. In Section 3 we review Word2Vec models on which our work is based. We define our model in Section 4 and 5. The experimental results on English corpora are presented in Sections 6.1, and 7. We conclude in Section 8 and offer some directions for future work.

2. Related work

During the past decades, simple frequency-based methods for deriving word meaning from a raw text were popular, e.g. Hyperspace Analogue to Language [18] or Correlated Occurrence Analogue to Lexical Semantics [29] as a representatives of local-context DSMs and Latent Semantic Analysis [15] or Explicit Semantic Analysis [7] as a representatives of global-context DSMs. All these methods record word/context co-occurrence statistics into one large matrix defining the semantic space.

Later on, these approaches have evolved into more sophisticated models. [22] came with two neural network based models CBOW and *Skip-Gram* on which this work is based. This single-layer architecture is based on the inner product between the two word vectors (detailed description is in Section 3). [27] 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 the meaning of words.

Both above mentioned models currently serve as a basis for the work of many researchers. [1] improved the Skip-Gram model by incorporating a sub-word information. Similarly, in the most recent study [32] incorporated a sub-word information into the LexVec [31] vectors. This improvement is especially evident for languages with rich morphology. [17] 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).

During the most recent years the Deep Learning[16] methods based on CNN or LSTM architectures give the best score on various NLP tasks. Those approaches can directly extract word embeddings in the first layers and use them in deeper layers for decision making. However, we believe that extracted word embeddings are not in the quality to be used for wide range of NLP tasks and are suited to particular task on which the network is being trained. Deep Learning approaches need a lot of data of particular NLP task that are not always available. We also believe that highly tuned word embeddings trained separately that are further used together with Deep Learning architectures often lead to even better accuracy and generalization.

[13] 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 a 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.

Our approach is focusing on the most widely used Skip-Gram/CBOW methods and as a source of a global document (respective article) context uses a Wikipedia, which is currently available for 301 languages. Therefore, our approach can be adopted to any other language without the necessity of manual data annotation.

3. Word2Vec

This section describes the Word2Vec package which utilizes two neural network model architectures (CBOW and Skip-Gram) to produce a distributional representation of words [22]. Given the training corpus represented as a set of documents $\mathbf{D}$. Each document (resp. article) $\mathbf{a_j} \in \mathbf{D}$ is a sequence of words $\mathbf{a_j} = \{w_{j,i}\}_{i=1}^{L_j}$, where L_j denote the length of the article $\mathbf{a_j}$. Each word w in the vocabulary $\mathbf{W}$ is represented by the two different vectors v and u – depending on whether it is used as a context word $\mathrm{v}_w \in \mathbb{R}^d$ or a target word $\mathrm{u}_w \in \mathbb{R}^d$. The task is to estimate these vector representations in a way that optimizes bellow described objective functions.

We use training procedure introduced in [23] called *negative sampling*, we define the negative log-likelihood:

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

where $\mathbf{N} = (w_n \subset P(\mathbf{W}) | n = 1, \ldots, K)$ is a set of negative samples (randomly selected words from a noise distribution $P(\mathbf{W})$, 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 $w_c \in \mathbf{C}_{j,i}$. The context is based on the bag-of-words hypothesis, so that the order of the words does not influence the prediction. CBOW optimizes the following objective function:

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

The position of a vector for each word is optimized in addition to the computed error.

In the Skip-Gram architecture, the model uses the current word $w_{j,i}$ to predict the surrounding context words $w_c \in \mathbf{C}_{j,i}$. Skip-Gram select the context window size randomly from uniform distribution $[1, S]$. Thus, nearby words have higher chance to be selected as a context words. This is based on intuition that nearby words have higher impact on semantics of center word $w_{j,i}$. Skip-Gram model optimizes the following objective function:

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

According to [22], CBOW is faster than Skip-Gram, but Skip-Gram usually perform better for infrequent words.

4. Wikipedia category representation

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 describes particular information in more detail. Wikipedia gives in

general much more information about an article, such as mentioned links to other articles, or at the end of the article there is a section that describes all categories where the actual article is belonging. Wikipedia has a big tree structure[1] of categories with one main category and a lot of subcategories. Every article contains several categories to which it belongs. Categories are intended to group together with pages on similar subjects.

For example the article entitled *United States* has categories *Countries in North America*, *English-speaking countries and territories*, *Federal constitutional republics*, *G7 nations*, 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.

5. Proposed model

Some authors tried to extract a more concrete meaning using *Frege's principle of compositionality* [26], which states that the meaning of a sentence is determined as a composition of words. [34] introduced several techniques to combine a word vectors into the final vector describing the sentence. [3] experimented with Semantic Textual Similarity; from the tests with words vector composition based on CBOW architecture, we can see that this method is powerful to carry the meaning of a complete sentence.

Our new model is shown in Fig. 1. We build up the model based on our previous knowledge and belief that Global information might improve the performance of word embeddings and further lead to improvements in many NLP subtasks such as Semantic Textual Similarity. Each article $\mathbf{a}_j$ in Wikipedia is associated with the set of categories $\mathbf{X}_j$. We represent the category $x \in \mathbf{X}_j$ as a real-valued vector $\mathbf{m}_x \in \mathbb{R}^d$.

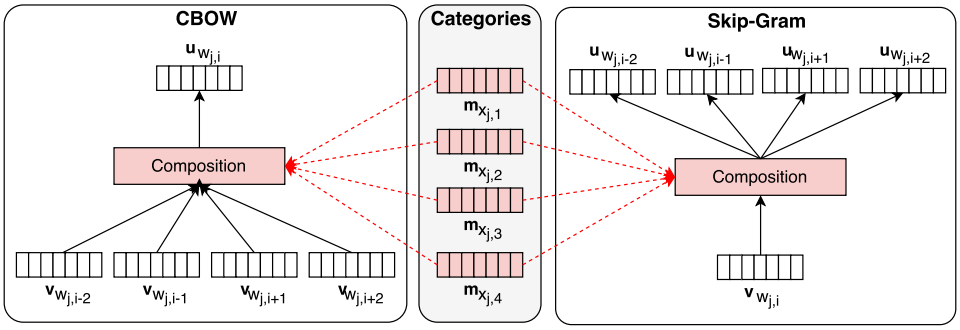

Fig. 1 *Architecture of proposed extension of CBOW and Skip-Gram models.*

The original CBOW model was adapted by incorporating categories and optimizes the following objective function:

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

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

Skip-Gram model optimizes the following objective function:

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

We tested with CBOW and Skip-Gram architectures. For CBOW architecture that is much faster and easier to train, we also experimented with further model setups:

5.1 Setup 1

Categories are initialized with uniform vector distribution and no training of categories is performed. Only a word embeddings are trained. Output of this setup is a model with trained word embeddings. The objective function 5 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 word vector representation, especially in describing the words with similar meaning and context.

5.2 Setup 2

Many models benefit from the weighing of words in a sentence using *Term Frequency – Inverse Document Frequency* (TF-IDF) [20]. Categories are initialized with a uniform vector distribution. Vectors representing categories were not trained, only weighted using *TF-IDF*. Punctuations, prepositions, conjunction and others have smaller impact on the overall meaning of the 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. The adapted objective function is as follows:

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

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

5.3 Setup 3

The Model is initialized with categories uniformly distributed, embeddings for categories are trained during training of word embeddings. Motivation of this setup

comes from Distributional hypothesis. If we train the categories, we assume they would behave similarly. For example, having article with categories 'vehicles' and 'transportation', these categories will likely have similar distribution of articles and they will slowly come closer to each other in vector space during the training. With uniformly distributed vectors representing such categories, we would not benefit from Distributional hypothesis. Outputs of the model are embeddings trained for both – categories and for words.

5.4 Setup 4

Firstly, the Model has trained the vectors representing categories (using *Setup 3*) and in second round we have used such pre-trained vectors and continue with *setup 1* – the use of the pre-trained embeddings for categories. The main motivation is to have categories organised in vector space according the meaning and help the word vectors from document to end up on vector positions that has better semantic and syntactic properties.

6. Training

We trained 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 that were assigned to categories, further we removed articles with less than 100 words or less than 10 sentences. We removed categories that had less than 10 occurrences in all articles. The final corpus used for training consist of 1,554,079 articles. Detailed statistics on these corpora are shown in Tab. I. For evaluation, we experiment with word analogy and a variety of word similarity datasets.

- Word similarity: These datasets are conducted to measure the semantic similarity between pair of words. For English, these include WordSim-353 [5], RG-65 [30], RW [19], LexSim-999 [12], and MC-28 [24].
- Word analogies: Follow observation that the word representation can capture different aspects of meaning, [22] introduced evaluation scheme based on word analogies. 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: $\mathrm{vec}(king) - \mathrm{vec}(queen) = \mathrm{vec}(man) - \mathrm{vec}(woman)$. We evaluate on English word analogy datasets, proposed by [22]. The word-phrases were excluded from original datasets, resulting in 8869 semantic and 10,675 syntactic questions for English (19,544 in total), and 6018 semantic.

6.1 Training setup

We tokenize and lowercase the corpus data. We use simple tokenizer based on regular expressions. After the model is trained, we keep the most frequent words

[2] dumps.wikimedia.org

	English (dump statistics)
Articles	5,164,793
Words	1,759,101,849
Categories	4,908,011

	English (final clean statistics)
Articles	1,554,079
Avg. words per article	437
Avg. number of categories per article	5
Number of unique words	4,754,040
Categories	4,015,918

Tab. I *Training corpora statistics. English Wikipedia dump from June 2016.*

in a vocabulary ($|\mathbf{W}| = 300,000$). Vector dimension for all our models is set to $\mathbf{d} = 300$. 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. $|\mathbf{C}_{j,i}| = 20$. The set of negative samples $\mathbf{N}$ is always sampled from unigram word distribution raised to 0.75 that has been experimentally shown to work the best [23] and has fixed size $|\mathbf{N}| = 10$ words. We do not use the subsampling of frequent words. Process of parameter estimation process is described in [8]. We prefixed categories to be unique in training and not interferred with words during training phase.

Setup 3 specified in Section 5.3 is easy to train, does not take much longer than usual training using Word2Vec (depending on categories volume). Due to the potential convergence issues (infinity vectors, adaptation of SkipGram algorithm has to be made) and potential much longer training time of Setup 4, the experiments with Skipgram architecture are made with setup 3 only. This setup can be an easy extension of Word2Vec or fastText toolkit as is.

6.2 Other models setup

- *fastText* – trained on our Wikipedia dump 2016 (see Tab. I).
- Subword *LexVec* English Wikipedia 2015 + NewsCrawl[3], has 7 billion words, 368,999 of unique words and vectors of 300 dimensions. Both (fastText and LexVec) models use character n-grams of length 3-6 as subwords.
- For a comparison with much larger training data, we downloaded *GoogleNews 100B*[4] model that is trained using Skipgram architecture on 100 billion words corpus and negative sampling, vocabulary size is 3,000,000 words.
- *GloVe* [27] models has 6 billion words, 400,000 unique words, are uncased, 300d vectors. Second model has 42 billion words, 1.9M of unique words and 300d vectors.

[3] http://www.statmt.org/wmt14/translation-task.html
[4] https://developer.syn.co.in/tutorial/bot/oscova/pretrained-vectors.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
	GloVe 300d 6B	65.80	77.80	72.70	–	77.40	67.00	71.70
	GloVe 300d 42B	**75.90**	**82.90**	**83.60**	–	**81.90**	69.30	**75.00**
	ESA [10]	**74.80**	74.90	–	27.10	–	–	–
	Huang [13]	**71.30**	–	–	–	–	–	–
	Levy [17]	62.60	77.10	–	–	16.20	52.60	36.10
	LexVec 7B	59.53	74.64	74.08	40.22	**80.92**	66.11	72.83
With cat.	CBOW 300d wiki + setup 1	62.25	67.13	74.93	33.66	73.98	68.26	70.55
	CBOW 300d wiki + setup 2	67.09	68.95	73.26	35.69	72.11	70.79	71.32
	CBOW 300d wiki + setup 3	63.20	78.16	78.11	**40.32**	77.31	68.68	72.13
	CBOW 300d wiki + setup 4	64.42	**79.36**	79.55	**40.32**	79.04	**71.16**	74.31
	SG 300d wiki + setup 3	62.55	**80.25**	**86.07**	33.54	80.77	71.05	74.93

Tab. II *Word similarity correlations and word analogy results on English (semantic and syntactic analogies). Result is expressed in percentage.*

7. Experimental results and discussion

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

From Tab. II we can see that all our setups significantly outperform baseline models. Our models in some tests also outperforms fastText architecture [1] that is a recent improvement of Word2Vec with sub-word information. The Setup 3 gives the best balanced score in terms of the performance/training speed. Our proposed model based on Skipgram architecture outperform the much larger (50×) model trained by Google on RG-65 and MC-28 datasets. We also achieved a good performance with CBOW architecture which gives in general worse performance than SkipGram. With our adaptations the CBOW achitecture has even outperformed the Skipgram architecture trained on much larger data – see results on RG-65 and semantic oriented analogy questions in Tab. II. In general we can see, that our model is powerful in semantics. There is also significant performance gain on WS-353 similarity dataset between all our setups.

In each column of the Tab. II, we have marked the top three best score. In first column for WS-353 similarity tests, the best architectures are the ESA [10], Huang [13] and GloVe [27] model trained on larger corpora. Skipgram and CBOW generally perform poorer on similarity data sets. However, in RG-65 and MC-28 our models outperformed other architectures, even the one trained on much larger corpora. The best score on the Wikipedia and Word analogy tests gives the fastText incorporating the subword-information. If we consider only models trained on data from a Wikipedia, our approach is the winner on 4 from 7 tests.

8. Conclusion and future work

Proposed models significantly outperforms other word representation methods when the same size training data are used and provide the similar performance compared with methods trained on much larger datasets. We experimented with four model architectures and several settings. We believe that using our method can have even bigger impact on poorly resourced and highly inflected languages, such as Czech from Slavic family. As this research is focused on improving essential part of machine text understanding, it can lead to better performance gains in all NLP tasks where word embeddings is being used. Word2Vec is generally used among the other architectures because of its balanced training speed/performance [22]. Our approach with adapted Word2Vec models using setup 3 gives the best balanced performance, training speed is similar to the default CBOW/Skip-gram architecture and gives similar performance to model trained on 50 times larger data corpora. Setup 3 we take into considerations as our baseline for a future work.

Our future work can lead to test further languages and integrate our model to the latest architectures such as *fastText* or *LexVec* to get additional performance improvement from incorporating a sub-word information. We can also experiment with further model settings such as weighting categories using Point-wise Mutual Information (PMI) [25] instead of the TF-IDF.

We provide the extracted documents and the trained word vectors publicly for research purposes at https://github.com/Svobikl/global_context/.

Acknowledgement

This work has been partly supported from ERDF "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)" (No.: CZ.02.1.01/0.0/0.0/17_048/0007267) and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

References

[1] BOJANOWSKI P., GRAVE E., JOULIN A., MIKOLOV T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*. 2017, 5, pp. 135–146.

[2] BRYCHCÍN T., KONOPÍK M. Latent semantics in language models. *Computer Speech & Language*. 2015, 33(1), pp. 88–108.

[3] BRYCHCÍN T., SVOBODA L. 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)*, 2016, pp. 588–594,

[4] CHARLES W.G. Contextual correlates of meaning. *Applied Psycholinguistics*. 2000, 21(4), pp. 505–524.

[5] FINKELSTEIN L., GABRILOVICH E., MATIAS Y., RIVLIN E., SOLAN Z., WOLFMAN G., RUPPIN E. Placing search in context: The concept revisited. *ACM Transactions on information systems*. 2002, 20(1), pp. 116–131.

[6] FIRTH J.R. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*. 1957.

[7] GABRILOVICH E., MARKOVITCH S. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*. 2009, 34, pp. 443–498.

[8] GOLDBERG Y., LEVY O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*. 2014.

[9] HARRIS Z.S. Distributional structure. *Word*. 1954, 10(2-3), pp. 146–162.

[10] HASSAN S., MIHALCEA R. *Semantic Relatedness Using Salient Semantic Analysis*. 2011. Available also from: https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3616.

[11] HERCIG T., BRYCHCÍN T., SVOBODA L., KONKOL M., STEINBERGER J. Unsupervised methods to improve aspect-based sentiment analysis in Czech. *Computación y Sistemas*. 2016, 20(3), pp. 365–375.

[12] HILL F., REICHART R., KORHONEN A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*. 2015, 41(4), pp. 665–695.

[13] HUANG E.H., SOCHER R., MANNING C.D., NG A.Y. 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*, 2012, pp. 873–882,

[14] KONKOL M., BRYCHCÍN T., KONOPÍK M. Latent semantics in named entity recognition. *Expert Systems with Applications*. 2015, 42(7), pp. 3470–3479.

[15] LANDAUER T.K., FOLTZ P.W., LAHAM D. An introduction to latent semantic analysis. *Discourse processes*. 1998, 25(2-3), pp. 259–284.

[16] LECUN Y., BENGIO Y., HINTON G. Deep learning. *nature*. 2015, 521(7553), pp. 436.

[17] LEVY O., GOLDBERG Y. Dependency-based word embeddings. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308,

[18] LUND K., BURGESS C. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*. 1996, 28(2), pp. 203–208.

[19] LUONG T., SOCHER R., MANNING C. Better word representations with recursive neural networks for morphology. In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 2013, pp. 104–113,

[20] MANNING C.D., MANNING C.D., SCHÜTZE H. *Foundations of statistical natural language processing*. 1999.

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

[22] MIKOLOV T., CHEN K., CORRADO G., DEAN J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013.

[23] MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G.S., DEAN J. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013, pp. 3111–3119,

[24] MILLER G.A., CHARLES W.G. Contextual correlates of semantic similarity. *Language and cognitive processes*. 1991, 6(1), pp. 1–28.

[25] PANTEL P., LIN D. Discovering Word Senses from Text. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada: ACM, 2002, pp. 613–619. KDD '02 series, doi: 10.1145/775047.775138. ISBN 1-58113-567-X.

[26] PELLETIER F.J. The principle of semantic compositionality. *Topoi*. 1994, 13(1), pp. 11–24.

[27] PENNINGTON J., SOCHER R., MANNING C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543,

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

[29] ROHDE D.L., GONNERMAN L.M., PLAUT D.C. An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Psychology*. 2004, 7, pp. 573–605.

[30] RUBENSTEIN H., GOODENOUGH J.B. Contextual correlates of synonymy. *Communications of the ACM*. 1965, 8(10), pp. 627–633.

[31] SALLE A., IDIART M., VILLAVICENCIO A. Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*. 2016.

[32] SALLE A., VILLAVICENCIO A. Incorporating Subword Information into Matrix Factorization Word Embeddings. *arXiv preprint arXiv:1805.03710*. 2018.

[33] TURNEY P.D., PANTEL P. From Frequency to Meaning: Vector Space Models of Semantics. *CoRR*. 2010, abs/1003.1141.

[34] ZANZOTTO F.M., KORKONTZELOS I., FALLUCCHI F., MANANDHAR S. Estimating Linear Models for Compositional Distributional Semantics. In: *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China: Association for Computational Linguistics, 2010, pp. 1263–1271. COLING '10 series. Available also from: http://dl.acm.org/citation.cfm?id=1873781.1873923,