The Effect of Unobserved Word-Context Co-occurrences on a Vector-Mixture Approach for Compositional Distributional Semantics

Amir Bakarov
The National Research University Higher School of Economics,
Federal Research Center 'Computer Science and Control' of Russian Academy of Sciences,
Moscow, Russia
amirbakarov@gmail.com

Abstract

Swivel (Submatrix-WIse Vector Embedding Learner) is a distributional semantic model based on counting point-wise mutual information values, capable of capturing word-context co-occurrences in the PMI matrix that were not noted in the training corpus. This model outperforms mainstream word embedding training algorithms such as Continuous Bag-of-Words, GloVe and Skip-Gram in word similarity and word analogy tasks. But the properness of these intrinsic tasks could be questioned, and it is unclear if the ability to count unobservable word-context co-occurrences could also be helpful for downstream tasks. In this work we propose a comparison of Word2Vec and Swivel for two downstream tasks based on natural language sentence matching: the paraphrase detection task and the textual entailment task. As a result, we reveal that Swivel outperforms Word2Vec in both cases, but the difference is minuscule. We can conclude, that the ability to learn embeddings for rarely co-occurring words is not so crucial for downstream tasks.

1. Introduction

Distributional semantic models (DSMs) are instruments which can represent words through real-valued vectors of fixed dimensions. The word *distributional* here is a reference to a *distributional hypothesis* saying that word semantics is distributed together with its contexts (Harris, 1954). DSMs can capture various functional or topical relations (for example, *semantic similarity* also known as *synonymy*) between words through collocation contexts observed in a corpus, and the strength of such relation between two words can be computed as a distance between vectors corresponding to these words.

There are two taxonomic classes of DSMs. The first one is based on counting word co-occurrences in a corpus (for example, through constructing a TF-IDF matrix or a PMI matrix). Each word could be represented as a sparse vector, and its dimensionality can be lowered by applying dimensionality reduction techniques (like SVD) to the matrix – this is how some 'classic' distributional models like LSA or PPMI matrix work (Landauer et al., 1998; Turney and Pantel, 2010). They are called *count-based models*.

Another option of capturing word semantics is based on sampling the training corpus with a sliding window while each word is initialized with a vector, which values are optimized for the task of predicting a word using its context (or, *vice versa*, predicting context using the word) (Collobert et al., 2011). DSMs that work in such a way are called *predictive models*, and the dense vectors produced by such models are called *word embeddings*.

Predictive models are the most mainstream class of DSMs since they proved their effectiveness in most NLP tasks. One of the most popular DSM models is *Word2Vec* (Mikolov et al., 2013) which provides two training architectures: *Continuous Bag-of-Words* (CBOW) that predicts words given their contexts, and *Skip-Gram* (SG) that predicts the contexts from the words.

The effectiveness of predictive models was revealed in the classic paper *Don't Count, Predict!* (Baroni et al., 2014), which proved the benefits of predictive models against count-based models. However,

some researchers still try to propose novel approaches to count-based training that could pretend to outperform the predictive approach. One of the most recent interesting methods was introduced in a novel DSM called *Swivel* (which is an abbreviation for *Submatrix-WIse Vector Embedding Learner*) (Shazeer et al., 2016). This model is based on applying SVD for PMI matrix, and the main idea is to use a loss function which penalties depend on whether the word-context pair co-occurs in the corpus or not, so the algorithm could be trained to not to over-estimate PMI of common values whose co-occurrence is unobserved. Notably, Word2Vec with a negative sampling is also capable of taking unobserved co-occurrences into account, but it is done indirectly.

The central claim of the authors of Swivel is that none of the mainstream word embeddings provide any special treatment to unobserved word-context co-occurrences (Shazeer et al., 2016), so the ability to capture unobserved word-context co-occurrences helped to outperform other embedding training algorithms in word similarity and word analogy tasks. But the properness of the considered intrinsic evaluation tasks (word similarity and word analogy) could be questioned due to the recent negative critique of word similarity and word analogy as methods for evaluating DSMs (Batchkarov et al., 2016; Gladkova et al., 2016). Hence, it is unclear if the ability to count unobservable word-context co-occurrences could be also helpful in downstream tasks and allow the count-based models to outperform the predictive ones.

To this end, in this paper we want to consider a more proper evaluation method that introduces two downstream tasks based on *natural language sentence matching*. Our evaluation methods rely on testing the ability of the compared DSMs to build vectors of compositional linguistic units (like sentences and documents). Different techniques of vector composition allow to represent these units in a vector space and calculate a similarity value between vectors corresponding to them.

The scientific question that raised in this work is whether taking unobserved word-contexts co-occurrences into account could help count-based models to outperform the neural-based ones in a downstream task. *Our main contribution* is the comparison of performance of three different DSMs on three datasets in order to answer this question. Our work is the first towards an evaluation of Swivel and Word2Vec on extrinsic tasks.

We think that evaluation in a downstream task could also reveal differences between the methods that do not seem evident when only an intrinsic task is used. On the other hand, we see our next contribution in raising an issue of whether intrinsic evaluation is enough to make a conclusion about DSMs performance since its performance on an extrinsic task could differ. The results obtained in this study prove the existence of such issue.

The paper is organized as follows. Section 2 describes the background of tasks that we use for our downstream evaluation. Section 3 provides a brief introduction to the background of compositional distributional semantics and the approach for obtaining compositional unit representations which we consider in this paper. Section 4 is dedicated to the experimental setup, while section 5 covers the results of an evaluation and brings a discussion on them. Section 6 mentions recent studies that we find relevant to our topic. Section 7 concludes the article.

2. Downstream tasks

2.1. Paraphrase detection

Paraphrase is a restatement of a text giving the same semantic meaning in another textual form. *Paraphrase detection task* (which could be also mentioned as a *semantic similarity identification task, sentence similarity detection task* since the paraphrased sentences are, in substance, semantically similar expressions (Xu et al., 2015)) for a pair of natural language sentences T_1 and T_2, is a task of identification whether T_1 and T_2 have the same semantic meaning. In this case, the pair of sentences could be only related to one of two labels:

- **semantic similarity exists:** *If you help the needy, God will reward you & You will receive the reward of God for your charity*;
- **semantic similarity does not exist:** *If you help the needy, God will reward you & You will receive the reward of God for your charity*.

2.2. Textual entailment

Textual entailment (also called *natural language inference*) is a concept describing a directional relation between two text fragments which holds whenever the truth of one text fragment (called hypothesis) follows from another text (called text) (Dagan et al., 2006). Therefore, a pair of natural language sentences T and H contains textual entailment if a human reading text would infer that H directly follows T, and T does not directly follow H. So one should notice that the textual entailment connection is directional unlike the connection of other types of semantical relatedness like paraphrasing. The pair of sentences can be labeled with three types of available entailment relations:

- **entailment:** the hypothesis entails the text if the relations or events described by the hypothesis are likely to be true given the relations or events described by the text (e.g. text: *If you help the needy, God will reward you*, hypothesis: *Giving money to a poor man has good consequences*);
- **contradiction:** the hypothesis contradicts the text if the relations or events described by the hypothesis are highly unlikely to be true given the relations or events described by the text (e.g. text: *If you help the needy, God will reward you*, hypothesis: *Giving money to a poor man has no consequences*);
- **neutral relation:** the hypothesis and the text are in a neutral relation if the relations or events between them are not semantically connected to each other (e.g. text:*If you help the needy, God will reward you*, hypothesis: *Giving money to a poor man will make you a better person*).

3. Vector mixture

Most of DSMs (particularly, the ones which we consider in this work) rely only on lexical semantics and do not build representations for other levels of text such as sentences or documents. Semantic modeling for composed linguistics units (sentences or documents) based on a distributional approach is usually considered as a separate subfield called *compositional distributional semantics* (CDS) (Mitchell and Lapata, 2010). The purpose of a compositional DSM is to provide a function that could produce a vector representation of the meaning of composed linguistic units from the distributional vectors of the words contained therein. The motivation of such approach is that if a sentence is a function of meaning of all its words, then word embeddings could also be treated as the building blocks of compositional representation. While it has been shown that semantic relations can be mapped to translations in the learned vector space, the claim could be made for sentence representations of the embeddings.

Mainly, approaches for constructing compositional distributional representations are divided into *vector-mixture based* (they rely on element-wise arithmetic operations on vectors), *tensor-based* (they additionally represent neighbor words as matrices in order to build a sentence representation) and *network-based* (they consider construction of weights for the sentence vector as a neural network's objective) (Sadrzadeh and Kartsaklis, 2016).

Since the downstream tasks which we consider rely on natural language sentence matching, they are actually linked with the task of construction of sentence vectors. Hence, the performance of a DSM hypothetically depends on the chosen vector composition algorithm. We consider *vector mixture* the most robust and interpretable technique of CDS modeling (Mitchell and Lapata, 2008), because while neural networks and additional matrices (from tensor-based and network-based approaches) could introduce bias in the obtained results due to their stochastic nature and abundance of new parameters in the algorithm (it is impossible to say how much of the conclusions would due to the neural network architecture itself, and how much to the optimization function, and how much to the first initialization of the weights), the beautiful simplicity of element-wise operations on word vectors guarantees than only the ability of the word-level distributional model to construct adequate representations is taken into account. Of course, vector mixture approach also has certain limitations (for example, it does not consider word order in the sentence, treating sentence like an unordered bag-of-vectors), and its linguistic justification could be possibly questioned, but the idea of relevance of operations on vectors for obtaining meaning transformation was justified by other tasks using vector operations like analogical reasoning (Le and

Mikolov, 2014). The possible justification of mixture models could be explained by the fact that if the vector of a word shows the extent to which this word is related to other words in the corpus, so will the compositional vectors show the extent to which things related to a certain vector can also be related to other vectors (Zhang et al., 2018).

Two main options of vector mixture include vector composition and vector multiplication. Feature-wise vector addition can be seen as feature union, and vector multiplication as feature intersection. In this work we will obtain compositional distributional representations through vector mixture approach based on element-wise vector addition. Treating a sentence as a bag of words, we will obtain its vector as an average of all vectors corresponding to all words it contains.

Our idea proposes that it is possible to use components V as features for learning the automatic classifier. So the list $V_1, ..., V_t$ of t sentence pairs could be used as a feature matrix for learning the classifier, and a vector $i_1, ..., i_t$ reporting the type of the relation between the sentences could be used as a target vector.

More formally, given two sentences, X with n words and Y with k words, one could obtain their vector representations, $S(X)$ and $S(Y)$, by averaging the vector of words that constitute the sentences:

$$S(X) = \frac{s(x_1)+s(x_2)+...+s(x_n)}{n}$$
$$S(Y) = \frac{s(y_1)+s(y_2)+...+s(y_k)}{k}$$

And then obtain vector representation of the two sentences:

$$V = \frac{S(X)+S(Y)}{2}$$

Therefore, such embedding of a sentence pair could be used to train the classifier to distinguish the existence of paraphrasing or the existence of entailment.

4. Experimental setup

As we mentioned before, our experimental setting includes the comparison of DSMs with different architectures trained on the same corpus with the same hyperparameters. As the training algorithms we use implementations of SG, CBOW and Swivel from the official repository of Tensorflow (Abadi et al., 2016), the most popular framework for deep learning. Our choice for the training data was the Gutenberg Project corpus (Lahiri, 2014) of 520 000 tokens containing English fiction. We used a filtered and lemmatized version of the corpus (lemmatization was done with UDPipe (Straka et al., 2016)). The main concern in the use of project Gutenberg corpus is that it contains a lot of words that are not used in our datasets, so we suppose that the use of fiction corpus may answer the question whether the ability of Swivel to use unobserved word-context co-occurrences could have an effect on the results since we consider that in all tasks related to distributional semantics the choice of the training corpus is highly decisive.

Each of the considered models was trained with context window of 10 and sub-sampling of 1e-3 (because these hyper parameters worked better among others that were compared); if it was possible, we turned on the negative sampling regime. For each of the training algorithms we trained a model with varying vector dimensions of 100, 150, 200, 250, 300, 350, 400, 450 and 500 (the highest boundary for vector size is explained by the fact that it is the threshold for which in our experiments most of the models stopped to increase performance significantly). We decided to check different vector dimensions since we suppose that in element-wise vector mixture the size of the vector could be crucial if we want to take into account all the data contained in vector components. All in all, we had 27 models: 3 different training algorithms and 9 vector sizes for each.

The datasets on which the models were evaluated are:

- **Sentences Involving Compositional Knowledge (SICK)**, 9 840 pairs of sentences assessed by textual entailment (Bentivogli et al., 2016);
- **Stanford Natural Language Inference (SNLI)**, 640 000 pairs of sentences assessed by textual entailment (Bowman et al., 2015);

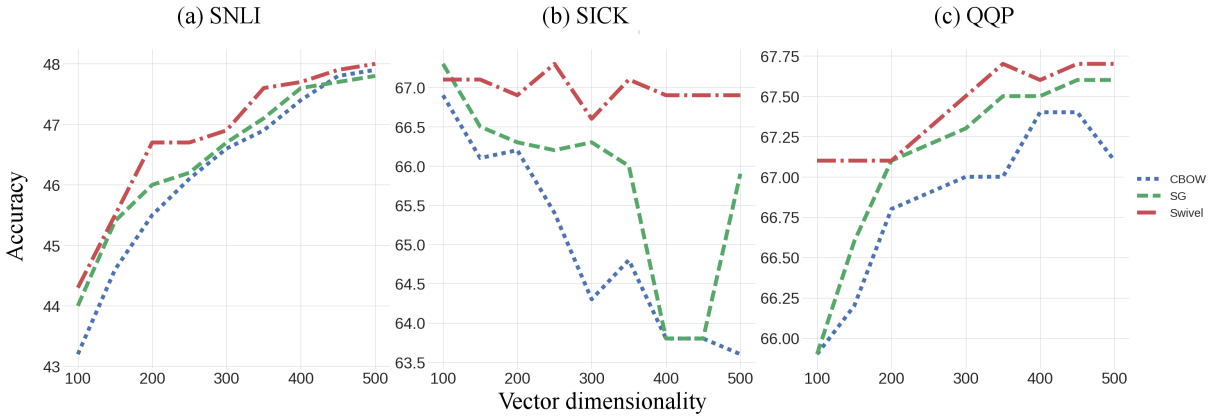

Figure 1: Accuracy of Swivel, CBOW and SG (in percents) on the considered datasets varying the dimensionality of word vectors.

- **Quora Question Pairs (QQP)**[1], 400 000 pairs of sentences assessed by semantic similarity.

The datasets were also lemmatized using the aforementioned UDPipe.

The method of capturing compositional distributional representation of the sentences included averaging all word embeddings of the sentence, as was mentioned in the previous section (out-of-vocabulary words were not taken into account). Then each dataset was represented as a labeled pair of vectors (binary labels in the case of the paraphrase detection task or multi-class nominal labels in the case of the textual entailment task).

These labels were used as target values, and each dataset was divided into two parts (in proportion $\frac{3}{1}$): one for training the classifier, and one for obtaining the final result of performance of given word embedding models by the classifier's prediction. For classification we used the *Logistic Regression* model implemented in a *Scikit-learn* module (Pedregosa et al., 2011). We also tested other popular classification algorithms (*Naive Bayes*, *Random Forest*, *K-Nearest Neighbors*, *Support Vector Machine*, and so on), but they were not able to outperform Logistic Regression (we do not mention the obtained results on these algorithms here since we find that they could blur the focus of our paper).

5. Results and discussion

	SNLI	SICK	QQP
1	**Swivel**	**Swivel**	**Swivel**
2	SG	SG	SG
3	CBOW	CBOW	CBOW

Table 1: Ranking of the compared models across the considered datasets by the best result shown on any vector dimensionality.

We evaluated 27 models on the test chunk on SNLI (a), SICK (b) and QQP (c), and calculated accuracy varying the vector dimensionality (we used accuracy since the datasets were balanced, and this measure was able to report actual quality of the classification). Figure 1 illustrates the results of the evaluation while vector dimensionality of the compared models varies. The overall relative rankings of compared models and their best results on each of datasets are presented at Table 1. Swivel showed the best performance on all tasks (48% of correct answers on SNLI, 66.9% on SICK and 67.2% on Quora).

Such result shows that Swivel actually works better than Word2Vec (although the difference in the results is small). This could be explained by the fact that the style used in project Gutenberg corpus

[1] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

differs from the one used in the proposed datasets. A claim could be made that this should just play to the strengths of Swivel, since it is supposedly strong for rarely co-occurring words. The results on all the datasets are far from state of the art (Conneau et al., 2017) (89.3% on SNLI (Tay et al., 2017), 80% on SICK (Bentivogli et al., 2016), 88.1% on QQP (Wang et al., 2017)). Anyway, the main aim of our work was not to propose the approach with the best performance, but to make a comparison of DSMs involving a robust and interpretable method, and we suppose that the performance gap is caused by the algorithm, not by the corpus – averaging word embeddings for obtaining sentence embeddings should work worse for long (more than approximately 10 words) sentences than sentence-level embeddings like *Skip-Thought* (Kiros et al., 2015) that were used in other papers. And this may be the cause why the SICK task demonstrates considerable changes in accuracy for different dimensions – sentences in SICK dataset are notably shorter than in two other datasets.

Another interesting result that we obtained is that models' performance in most cases increased with increasing vector dimensionality: for example, the dependence of accuracy of vector dimensionality for SNLI could be approximated with an ascending function. On the other hand, on SICK the performance of CBOW and SG sharply decreases on a threshold of 300 while the accuracy with Swivel did not decreased. The possible explanation is that SICK contains lexemes that rarely occurred in the training corpus, and the use of unobserved word-context co-occurrences helped Swivel to outperform other models.

The main limitation of Swivel that we see is that it requires notably more resources for training than Word2Vec. Training of Swivel on our small cluster equipped with ASUS GeForce GTX Titan X took up to 4 hours versus 10 minutes on CBOW and 30 minutes on SG. We can assume that in some tasks the difference in the results can be less notable feature than the time for training the model.

6. Related work

Exploratory research of DSMs was an object of big interest from the NLP community in recent years, and this interest is increasing with the popularity of word embeddings. Various researchers investigated different aspects of DSM, such as the underlying latent semantic structure (Senel et al., 2017), the effect of the chosen corpora (Kutuzov and Kunilovskaya, 2017) or the training algorithm (Bakarov and Gureenkova, 2017), the topology of gender (Bolukbasi et al., 2016), the nearest neighbors of frequent words (Radovanović et al., 2010), the types of semantic relations (Rei and Briscoe, 2014; Melamud et al., 2014) and so on. Particularly, in our work we actually put the effect of the training algorithm as a primary object of our research. To measure the performance we evaluated how well the considered model could work with two tasks based on natural language sentence matching: the paraphrase detection task and the textual entailment task.

The early approaches to a paraphrase detection task did not considered semantic structure at all, taking into account only word- and subword-levels (Dolan et al., 2004). Later, some researchers started to use manually constructed thesauri like WordNet (Lee and Cheah, 2016). Nowadays the state-of-the-art methods rely on bidirectional deep neural networks; for example, BiMPM (Wang et al., 2017). As the most comprehensive work on the task of paraphrase detection as well as textual entailment we consider (Androutsopoulos and Malakasiotis, 2010). Due to the fact that the task of textual entailment is also strongly linked with the task of matching natural language sentences, a lot of approaches for this task are highly similar to the approaches for paraphrase detection. All in all, these two tasks are highly popular in word embeddings community and are used as subtasks of more global tasks, like information retrieval, plagiarism detection, and more (Zubarev and Sochenkov, 2017).

7. Conclusions

Our experiments demonstrate that Swivel outperforms Continuous Bag-of-Word and Skip-Gram in modeling compositional distributional semantics on the variety of tasks of natural language sentence matching (however, the difference in the performance in most considered cases is tenths of a percent). We conclude that taking into account unobserved word-context co-occurrences plays a certain role in downstream tasks like the ones considered in the presented study.

Since we have not concluded which method is basically better we think that it will be reasonable to

use some kind of retrofitting technique for refining the word vectors in future (Faruqui et al., 2015) to be able to prove our hypothesis that fiction corpus should provide better results, so one could unearth how much does the corpus size and choice affect the embeddings.

In the future work we also wish to reproduce the results on the non-English data: for instance, on *Bulgarian language*. It is interesting to see whether the ranking positions of the compared algorithms will be reproduced. However, now we are not able to make such comparison since we are not aware of any paraphrase detection (or textual entailment dataset) for Bulgarian (despite the task of textual entailment was considered at the *Computational Linguistics in Bulgaria* conference, but not from the perspective of Bulgarian (Dias and Moraliyski, 2014)). Creation of such benchmark for Bulgarian is also in our plans for future work on this topic. In contrast to English, Bulgarian is a highly fusional language, and we think that research of word embeddings evaluation on Bulgarian could give many interesting insights for researchers in the field of evaluation of word embeddings as well for the international NLP community.

Acknowledgements

We thank our colleague, Andrey Kutuzov, for productive discussion on this paper. We also want to thank three anonymous reviewers for attentive reviewing and helpful suggestions.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Bakarov, A. and Gureenkova, O. (2017). Automated detection of non-relevant posts on the russian imageboard "2ch": Importance of the choice of word representations. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 16–21. Springer.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247.

Batchkarov, M., Kober, T., Reffin, J., Weeds, J., and Weir, D. (2016). A critique of word similarity as a method for evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 7–12.

Bentivogli, L., Bernardi, R., Marelli, M., Menini, S., Baroni, M., and Zamparelli, R. (2016). Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50(1):95–124.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Dias, S. P. G. and Moraliyski, R. (2014). Unsupervised and language-independent method to recognize textual entailment by generality. In *First International Conference Computational Linguistics in Bulgaria (CLIB 2014)*.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

Gladkova, A., Drozd, A., and Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

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

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Kutuzov, A. and Kunilovskaya, M. (2017). Size vs. structure in training corpora for word embedding models: Araneum russicum maximum and russian national corpus. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 47–58. Springer.

Lahiri, S. (2014). Complexity of word collocation networks: A preliminary structural analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–105.

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

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lee, J. C. and Cheah, Y.-N. (2016). Paraphrase detection using semantic relatedness based on synset shortest path in wordnet. In *Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On*, pages 1–5. IEEE.

Melamud, O., Dagan, I., Goldberger, J., Szpektor, I., and Yuret, D. (2014). Probabilistic modeling of joint-context in distributional similarity. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 181–190.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Rei, M. and Briscoe, T. (2014). Looking for hyponyms in vector space. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 68–77.

Sadrzadeh, M. and Kartsaklis, D. (2016). Compositional distributional models of meaning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 1–4.

Senel, L. K., Utlu, I., Yucesoy, V., Koc, A., and Cukur, T. (2017). Semantic structure and interpretability of word embeddings. *arXiv preprint arXiv:1711.00331*.

Shazeer, N., Doherty, R., Evans, C., and Waterson, C. (2016). Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Straka, M., Hajic, J., and Straková, J. (2016). Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *LREC*.

Tay, Y., Tuan, L. A., and Hui, S. C. (2017). A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Wang, Z., Hamza, W., and Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Xu, W., Callison-Burch, C., and Dolan, B. (2015). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 1–11.

Zhang, R., Guo, J., Lan, Y., Xu, J., and Cheng, X. (2018). Aggregating neural word embeddings for document representation. In *European Conference on Information Retrieval*, pages 303–315. Springer.

Zubarev, D. and Sochenkov, I. (2017). Paraphrased plagiarism detection using sentence similarity. *Dialog*.