Syntactic Interchangeability in Word Embedding Models

Daniel Hershcovich **Assaf Toledo** **Alon Halfon** **Noam Slonim**
IBM Research
daniel.hershcovich@gmail.com,
assaf.toledo@ibm.com,
{alonhal,noams}@il.ibm.com

Abstract

Nearest neighbors in word embedding models are commonly observed to be semantically similar, but the relations between them can vary greatly. We investigate the extent to which word embedding models preserve syntactic interchangeability, as reflected by distances between word vectors, and the effect of hyper-parameters—context window size in particular. We use part of speech (POS) as a proxy for syntactic interchangeability, as generally speaking, words with the same POS are syntactically valid in the same contexts. We also investigate the relationship between interchangeability and similarity as judged by commonly-used word similarity benchmarks, and correlate the result with the performance of word embedding models on these benchmarks. Our results will inform future research and applications in the selection of word embedding model, suggesting a principle for an appropriate selection of the context window size parameter depending on the use-case.

1 Introduction

Word embedding algorithms (Mikolov et al., 2013; Pennington et al., 2014; Levy et al., 2015) attempt to capture the semantic space of words in a metric space of real-valued vectors. While it is common knowledge that the hyper-parameters used to train these models affects the semantic properties of the distances arising from them (Bansal et al., 2014; Lin et al., 2015; Goldberg, 2016; Lison and Kutuzov, 2017), and indeed, it has been shown that they capture many different semantic relations (Yang and Powers, 2006; Agirre et al., 2009), little has been done to *quantify* the effect of model hyper-parameters on output tendencies. Here we begin to answer this question, evaluating fastText (Bojanowski et al., 2017) on benchmarks designed to measure how well a model captures the degree of similarity between words (§2).

In our experiments, we investigate how *syntactic interchangeability* of words, represented by their part of speech (§3), is expressed in word embedding models and evaluation benchmarks.

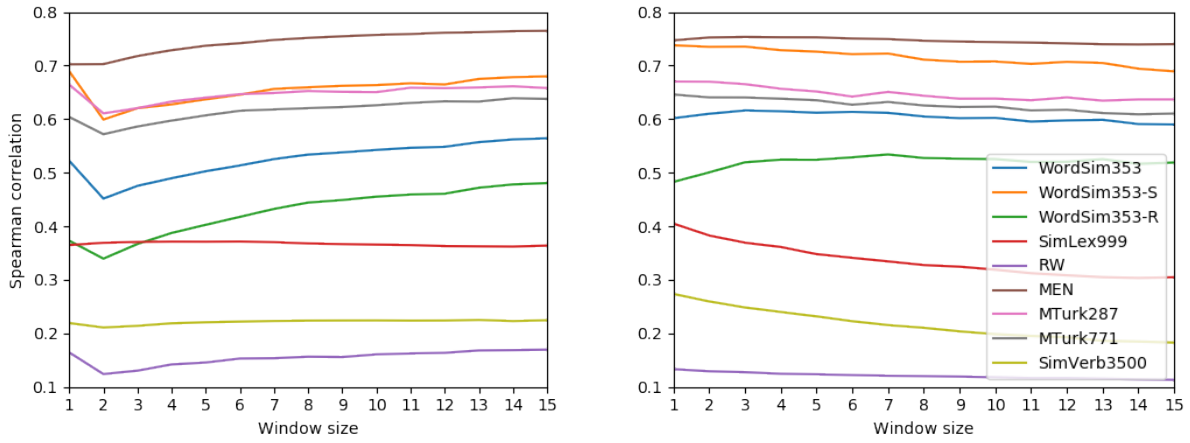
Based on the distributional hypothesis (Harris, 1954), word embeddings are learned from text by first extracting co-occurrences—finding, for each word token, all words within a context window around it, whose size (or maximal size) is a hyper-parameter of the training algorithm. Word vectors are then learned by predicting these co-occurrences or factorizing a co-occurrence matrix.

We discover a clear relationship between the context window size hyper-parameter and the performance of a word embedding model in estimating the similarity between words. To try to explain this relationship, we quantify how syntactic interchangeability is reflected in each benchmark, and its relation to the context window size. Our experiments reveal that context window size is negatively correlated with the number of same-POS words among the nearest neighbors of words, but that this fact is not enough to explain the complex interaction between context window size and performance on word similarity benchmarks.[1]

2 Word Similarity and Relatedness

Many benchmarks have been proposed for the evaluation of unsupervised word representations. In general, they can be divided into intrinsic and extrinsic evaluation methods (Schnabel et al., 2015; Chiu et al., 2016; Jastrzebski et al., 2017; Alshargi et al., 2018; Bakarov, 2018). While most datasets report the semantic similarity between words, many datasets actually capture semantic relatedness (Hill et al., 2015; Avraham and Goldberg, 2016), or more complex relations such as analogy or the ability to

[1] Our code and data are available at https://github.com/danielhers/interchangeability.



(a) CBOW (b) SGNS

Figure 1: Performance of the CBOW (a) and SGNS (b) algorithms on each benchmark, for each window size, measured by Spearman correlation between the benchmark score and the word embedding cosine similarity.

categorize words based on the distributed representation encoded in word embeddings. We focus on similarity and relatedness, and evaluate word embedding models on several common benchmarks.

2.1 Data

We learn word embeddings from English Wikipedia, using a dump from May 1, 2017.[2] The data is preprocessed using a publicly available preprocessing script,[3] extracting text, removing nonalphanumeric characters, converting digits to text, and lowercasing the text.

Benchmarks. We use the following benchmarks: WordSim-353 (Finkelstein et al., 2001) and its partition into WordSim-353-Sim (Agirre et al., 2009) and WordSim-353-Rel (Zesch et al., 2008), SimLex999 (Hill et al., 2015), Rare Words (RW; Luong et al., 2013), MEN (Bruni et al., 2012), MTurk-287 (Radinsky et al., 2011), MTurk-771 (Halawi et al., 2012), and SimVerb-3500 (Gerz et al., 2016). See Table 1 for the size of each benchmark.

2.2 Hyper-parameters

We use fastText (Bojanowski et al., 2017) to learn 300-dimensional word embedding models, using both the CBOW (continuous bag-of-words) and SGNS (skip-gram with negative sampling) algorithms (Mikolov et al., 2013). The context window size varies from 1 up to 15. We include only all words occurring 500 times or more (including function words), to avoid very rare words or uncommon spelling errors from skewing the results. All other hyper-parameters are set to their default values.

2.3 Evaluation on Benchmarks

To investigate the effect of window size on a model's performance on the benchmarks, we evaluate each model on each benchmark, using cosine similarity as the model's prediction for each pair. The performance is measured by Spearman correlation between the benchmark score and the word embedding cosine similarity (Levy et al., 2015).

Results. Figure 1 displays the performance of the CBOW and SGNS algorithms on each benchmark, with window sizes 1 to 15. Apart from a small dip between windows 1 and 2 for CBOW, the performance is either nearly constant, or changes nearly monotonically with window size in each setting.

The relative improvement (or deterioration), in percents, with the increase of window size from 2 to 15, are shown in Table 1 ($\Delta\texttt{win} = 2 \rightarrow 15(\%)$). Interestingly, CBOW exhibits a positive correlation of window size with model's performance for all benchmarks but SimLex999, while performance for SGNS barely changes with window size, except for SimLex999 and SimVerb3500, where we see a strong *negative* correlation.

Discussion. In SimLex999 and in SimVerb3500, the words in each pair have the same part of speech by design (in particular, SimVerb3500 only contains verbs). Hypothesizing that the effect of window size is related to the model's implicitly learned

[2] https://dumps.wikimedia.org/enwiki

[3] http://mattmahoney.net/dc/textdata.html

Benchmark	Size	Δwin $= 2 \rightarrow 15(\%)$ CBOW	Δwin $= 2 \rightarrow 15(\%)$ SGNS	# Related All	# Related Same-POS	# Unrelated All	# Unrelated Same-POS	p-value
WordSim353	353	24	-3	122	107	53	40	0.038
WordSim353-S	203	13	-6	60	53	53	40	0.061
WordSim353-R	252	42	4	104	89	39	31	0.26
SimLex999	999	-1	-20	234	199	334	295	0.897
RW	2034	37	-12	944	555	262	144	0.149
MEN	3000	9	-2	791	564	781	439	$3 \cdot 10^{-10}$
MTurk287	287	8	-5	49	39	119	68	0.004
MTurk771	771	12	-5	204	153	200	146	0.365
SimVerb3500	3500	6	-30	633	265	1217	566	0.974

Table 1: Analysis of interchangeability (by same-POS) in word similarity and relatedness benchmarks. Δwin $= 2 \rightarrow 15(\%)$ is the relative change, in percents, of the model's performance (by Spearman correlation) when going from window size 2 to window size 15, for the CBOW and SGNS algorithms (§2.3). *Related* and *Unrelated* are the top and bottom 30% of the pairs, by benchmark score, respectively. *P-value* is calculated using the hypergeometric test, comparing the enrichment of interchangeable pairs within related pairs, with a background of all related and unrelated pairs (§3.1).

concept of part of speech, we investigate this idea in the next section.

3 Syntactic Interchangeability

A word's part of speech (also known as syntactic category) is determined by syntactic distribution, and conveys information about how a word functions in the sentence (Carnie, 2002). We can generally substitute each word in a sentence with various words that are of the same part of speech, but not words that are of different parts of speech. While the same syntactic function can sometimes be fulfilled by words of various parts of speech or possibly longer phrases (such as adverbs and prepositional phrases, or multi-word expressions), part of speech is nonetheless a very good proxy for syntactic distribution (Mohammad and Pedersen, 2004).

Related to our work, Vulić et al. (2017) introduced a framework for automatic selection of specific context configurations for word embedding models per part of speech, improving performance on the SimLex999 benchmark. We take a different approach, investigating existing word embedding models and the way in which part of speech is reflected in them.

We define two words to be (syntactically) *interchangeable* if they share the same part of speech. We quantify interchangeability as a property of a word embedding model, as the proportion of words with the same part of speech within the list of nearest neighbors (that is, the most similar words according to the model) for each word in a pre-determined vocabulary. The higher the interchangeability ratio is, the more importance we assume the model implicitly places on interchangeability for the calculation of word similarity.

3.1 Interchangeability Analysis in Word Similarity Benchmarks

While all benchmarks we experiment with assign a score along a scale to each pair (calculated from human scoring), for our experiment we would like to use a binary annotation of whether a pair is related or not. For this purpose, we divide the whole range of scores, for each benchmark, to three parts: the lowest 30% of the range between the lowest and highest scores is considered "unrelated", the top 30% as "related", and the middle 40% are ignored.

Interchangeability enrichment. Given the binary classification obtained from the human-annotated scores for each benchmark, we can find the enrichment of interchangeable pairs among

related pairs. We use spaCy 2.0.11[4] (with the `en_core_web_sm` model) to annotate the POS for each word in each benchmark pair (tagging them in isolation to select the most probable POS), and look at the set of same-POS pairs in the benchmark. For each of the benchmarks, we calculate a p-value using the hypergeometric test, comparing the enrichment of same-POS pairs within related pairs, with a background distribution of all related and unrelated pairs (ignoring ones in the middle 40% range of scores).

Results. Table 1 shows the enrichment of interchangeable pairs among related and unrelated pairs for each benchmark. For WordSim353, MEN and MTurk287, the set of related pairs contains significantly more interchangeable pairs than the background set ($p < 0.05$),[5] suggesting that these benchmarks are particularly sensitive to POS.

3.2 Nearest Neighbor Analysis

To try and relate the results from §2.3 and §3.1, we measure the relation between window size and interchangeability by analyzing nearest neighbors in word embedding models. In our experiment, the *nearest neighbors* of a word are the words with the highest cosine similarity between their vectors.

Collecting pivots. We create a word list for each of the three most common parts of speech: nouns, adjectives and verbs. For each POS, we list all lemmas of all synsets of that POS from WordNet (Miller, 1998). To "purify" the lists and avoid noise from homonyms, we remove from each list any lemma that also belongs to a synset from another POS. As a further cleaning step, we use spaCy to tag each word, and only keep words for which the spaCy POS agrees with the WordNet POS. Without context, spaCy will likely choose the most common POS based on its training corpus, which is different from WordNet, increasing the robustness.

This process results in 6407 *uniquely-noun*, 2784 *uniquely-adjective* and 1460 *uniquely-verb* words, which we refer to as our *pivot lists*.

Calculating nearest neighbor POS. We find the 100 nearest neighbors for each word in our pivot lists, according to each fastText model with windows 1 through 15. We filter these neighbors to keep only words in the spaCy vocabulary, and inspect the remaining top 10. Again using spaCy, we tag the POS of each neighbor in the result. We subsequently calculate a histogram, for each POS x, of its *neighbor-POS* y, that is, the POS assigned to the neighbors of words with POS x.

Algorithm	NOUN			ADJ			VERB		
	1	15	r	1	15	r	1	15	r
CBOW	79	70	-0.96	72	48	-0.93	55	41	-0.91
SGNS	78	66	-0.95	66	39	-0.94	51	41	-0.92

Table 2: Percentage of interchangeable neighbors per pivot POS for the smallest (1) and largest (15) windows in our experiment, for the CBOW and SGNS algorithms. The number of interchangeable neighbors has a strong negative Pearson correlation (r) with window size for windows 1 to 15 ($p < 0.01$, two-tailed t-test).

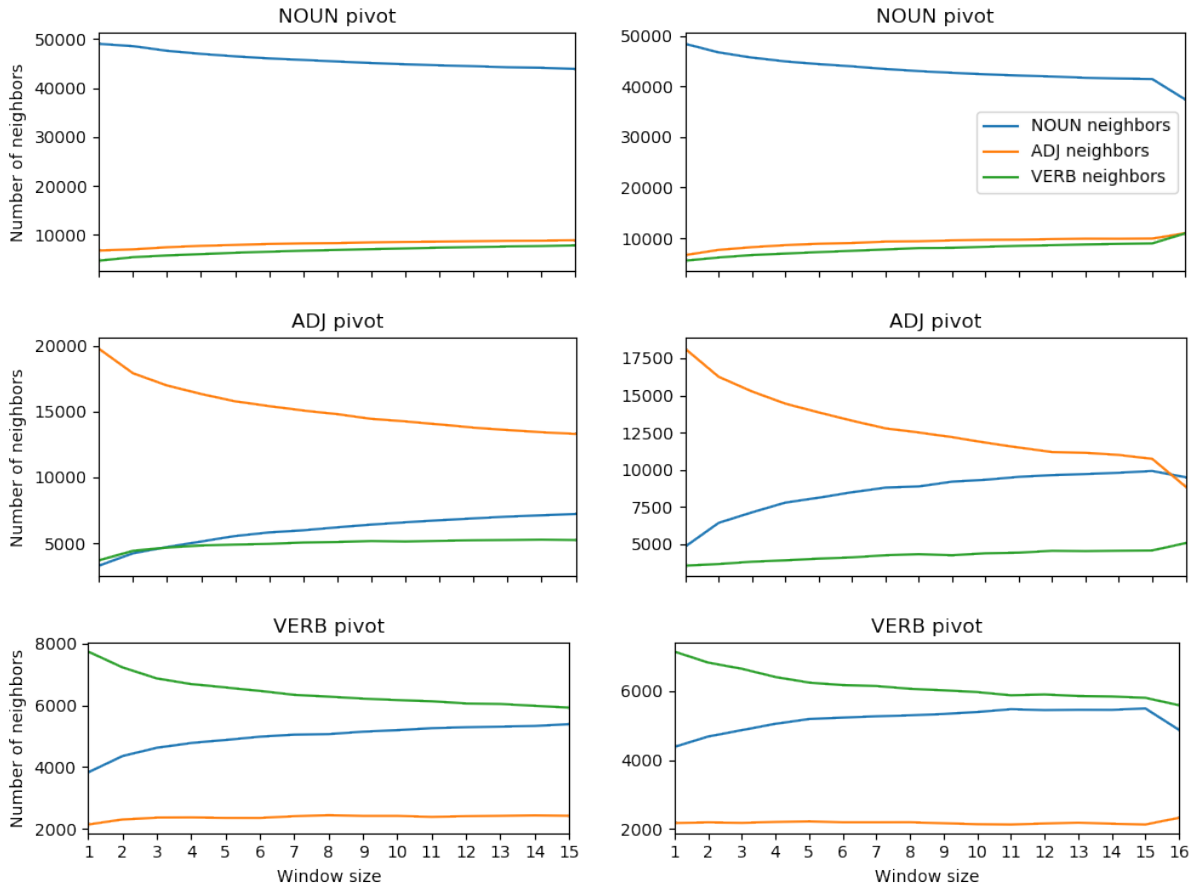
Results. Table 2 shows the results of this experiment. For nouns, adjectives and verbs, we consistently see a decrease in the number of same-POS neighbors when we increase the window size, relative to the total number of nearest neighbors.

Figure 2 shows the the absolute number of neighbors per algorithm, pivot POS and neighbor POS, for all window sizes we experimented with. The number of nearest neighbors of the same POS is consistently decreasing with window size, while the number of nearest neighbors of other POS are increasing or unaffected.

Discussion. The results clearly suggest that for *both* CBOW and SGNS, models with a larger window size are less likely to consider words of the same POS as strongly related. That is, syntactic interchangeability is negatively correlated with window size. This is in sharp contrast to our results from §2.3, where performance for CBOW on almost all benchmarks (among them WordSim353, MEN and MTurk287, for which we showed that syntactic interchangeability plays a role) consistently *improved* with window size. We also find the conclusion to contradict the impression regarding SGNS, where SimLex999 and SimVerb3500 showed worse performance for larger windows: if POS should not play a role in these benchmarks, then models with a bias toward syntactic interchangeability (i.e., models with lower windows) should perform *worse* on these benchmarks.

[4] https://spacy.io

[5] The fact that not all pairs in SimLex999 and SimVerb3500 are judged as interchangeable in our experiment is due to ambiguity: for some words, spaCy selected a POS which is not the one intended when constructing the benchmark.



(a) CBOW

(b) SGNS

Figure 2: Number of neighbor per POS for each pivot POS and for each window size, for the CBOW (a) and SGNS (b) algorithms. The number of same-POS neighbors is consistently decreasing with window size.

4 Conclusion

We investigated the effect of the context window size hyper-parameter on the performance on word similarity benchmarks. We showed that (1) increasing the window size results in a lower probability of interchangeable nearest neighbors for both CBOW and SGNS algorithms; (2) in some widely used benchmarks, syntactic interchangeability increases the probability of similarity or relatedness; (3) increasing the window size typically improves performance in predicting similarity or relatedness for CBOW, but has little impact on SGNS.

SimLex999 and SimVerb3500 proved to be exceptions to both (2) and (3), since all pairs in them are interchangeable by construction, but on them, increasing the window size has no effect for CBOW and negative impact for SGNS.

This contradiction is presented as a challenge to the community, and could perhaps be explained by other factors affected by window size.

Our investigation focused on a specific relation between words, namely whether they share a part of speech. Many other relations are of interest to the NLP community, such as syntactic dependency relations, and semantic relations like hypernymy and synonymy. Furthermore, a similar analysis could be applied to other word embedding hyper-parameters, such as the vector dimension. While we used a constant vector dimension of 300 in our experiments, it is an open question whether models with different vector dimensions differ with respect to their tendency to capture different word relations. Future work will extend our analysis to other relations and hyper-parameters.

Acknowledgments

We thank the anonymous reviewers for their helpful comments.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Faisal Alshargi, Saeedeh Shekarpour, Tommaso Soru, Amit Sheth, and Uwe Quasthoff. 2018. Concept2vec: Metrics for evaluating quality of embeddings for ontological concepts. *arXiv preprint arXiv:1803.04488*.

Oded Avraham and Yoav Goldberg. 2016. Improving reliability of word similarity evaluation by redesigning annotation task and performance measure. *arXiv preprint arXiv:1611.03641*.

Amir Bakarov. 2018. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 809–815.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Andrew Carnie. 2002. *Syntax: A Generative Introduction (Introducing Linguistics)*. Blackwell Publishing.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *EMNLP*.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.

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

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

Stanisław Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. *arXiv preprint arXiv:1503.06760*.

Pierre Lison and Andrey Kutuzov. 2017. Redefining context windows for word embedding models: An experimental study. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 284–288. Association for Computational Linguistics.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

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

George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Saif Mohammad and Ted Pedersen. 2004. Combining lexical and syntactic features for supervised word sense disambiguation. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*.

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

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Ivan Vulić, Roy Schwartz, Ari Rappoport, Roi Reichart, and Anna Korhonen. 2017. Automatic selection of context configurations for improved class-specific word representations. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 112–122, Vancouver, Canada. Association for Computational Linguistics.

Dongqiang Yang and David Martin Powers. 2006. Verb similarity on the taxonomy of wordnet. In *Proceedings of GWC*.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using wiktionary for computing semantic relatedness. In *AAAI*, volume 8, pages 861–866.