Towards Qualitative Word Embeddings Evaluation: Measuring Neighbors Variation

Bénédicte Pierrejean and **Ludovic Tanguy**
CLLE: CNRS & Université de Toulouse
Toulouse, France
{benedicte.pierrejean,ludovic.tanguy}@univ-tlse2.fr

Abstract

We propose a method to study the variation lying between different word embeddings models trained with different parameters. We explore the variation between models trained with only one varying parameter by observing the distributional neighbors variation and show how changing only one parameter can have a massive impact on a given semantic space. We show that the variation is not affecting all words of the semantic space equally. Variation is influenced by parameters such as setting a parameter to its minimum or maximum value but it also depends on the corpus intrinsic features such as the frequency of a word. We identify semantic classes of words remaining stable across the models trained and specific words having high variation.

1 Introduction

Word embeddings are widely used nowadays in Distributional Semantics and for a variety of tasks in NLP. Embeddings can be evaluated using extrinsic evaluation methods, i.e. the trained embeddings are evaluated on a specific task such as part-of-speech tagging or named-entity recognition (Schnabel et al., 2015). Because this type of evaluation is expensive, time consuming and difficult to interpret, embeddings are often evaluated using intrinsic evaluation methods such as word similarity or analogy (Nayak et al., 2016). Such methods of evaluation are a good way to get a quick insight of the quality of a model. Many different techniques and parameters can be used to train embeddings and benchmarks are used to select and tune embeddings parameters.

Benchmarks used to evaluate embeddings only focus on a subset of the trained model by only evaluating selected pairs of words. Thus, they lack information about the overall structure of the semantic space and do not provide enough information to understand the impact of changing one parameter when training a model.

We want to know if some parameters have more influence than others on the global structure of embeddings models and get a better idea of what varies from one model to another. We specifically investigate the impact of the architecture, the corpus, the window size, the vectors dimensions and the context type when training embeddings. We analyze to what extent training models by changing only one of these parameters has an impact on the models created and if the different areas of the lexicon are impacted the same by this change.

To do so, we provide a qualitative methodology focusing on the global comparison of semantic spaces based on the overlap of the N nearest neighbors for a given word. The proposed method is not bound to the subjectivity of benchmarks and gives a global yet precise vision of the variation between different models by evaluating each word from the model. It provides a way to easily investigate selected areas, by observing the variation of a word or of selected subsets of words.

We compare 19 word embedding models to a default model. All models are trained using the well-known *word2vec*. Using the parameters of the default model, we train the other models by changing the value of only one parameter at a time. We first get some insights by performing a quantitative evaluation using benchmark test sets. We then proceed to a qualitative evaluation by observing the differences between the default model and every other model. This allows us to measure the impact of each parameter on the global variation as well as to detect phenomena that were not visible when evaluating only with benchmark test sets. We also identify some preliminary features for words remaining stable independently of the parameters used for training.

Proceedings of NAACL-HLT 2018: Student Research Workshop, pages 32–39
New Orleans, Louisiana, June 2 - 4, 2018. ©2017 Association for Computational Linguistics

2 Related Work

The problems raised for the evaluation of Distributional Semantic Models (henceforth DSMs) is not specific to word embeddings and have been given attention for a long time. Benchmarks only focus on a limited subset of the corpus. For example, WordSim-353 (Finkelstein et al., 2002) is testing the behaviour of only 353 pairs of words meaning we only get a partial representation of the model performance. To get a better idea of the semantic structure of DSMs and of the type of semantic relations they encode, some alternative datasets were designed specifically for the evaluation of DSMs (Baroni and Lenci, 2011; Santus et al., 2015). Although these datasets provide a deeper evaluation, they focus on specific aspects of the model and we still need a better way to understand the global impact of changing a parameter when training DSMs.

Some extensive studies have been made comparing a large number of configurations generated by systematic variation of several parameters. Lapesa and Evert (2014) evaluated 537600 models trained using combinations of different parameters. Other studies focused on a specific parameter when training embeddings such as the corpus size (Asr et al., 2016; Sahlgren and Lenci, 2016), the type of corpus used (Bernier-Colborne and Drouin, 2016; Chiu et al., 2016) or the type of contexts used (Levy and Goldberg, 2014; Melamud et al., 2016; Li et al., 2017). Results showed that choosing the right parameters when training DSMs improve the performance for both intrinsic and extrinsic evaluation tasks and can also influence the type of semantic information captured by the model. Levy et al. (2015) even found that tuning hyperparameters carefully could prove better in certain cases than adding more data when training a model.

Chiu et al. (2016) showed that the performance of DSMs is influenced by different factors including corpora, preprocessing performed on the corpora, architecture chosen and the choice of several hyperparameters. However they also noticed that the effects of some parameters are mixed and counterintuitive.

Hamilton et al. (2016) measured the variation between models by observing semantic change using diachronic corpora. Hellrich and Hahn (2016) also used diachronic corpora to assess the reliability of word embeddings neighborhoods. Antoniak and Mimno (2018) showed how the corpus influences the word embeddings generated.

We relate to these studies but rather than finding the best combination of parameters or focusing on a single parameter, we assess the individual impact of selected parameters when training word embeddings. We intent to investigate those effects by getting a global vision of the change from one model to another. Unlike benchmarks test sets, we will not focus on evaluating only selected word pairs from the different models but we will evaluate the variation for each word from one model to the other.

3 Measuring neighbors variation

To evaluate the different models trained, we focus on the study of neighbors variation between two models. This type of approach was proposed by Sahlgren (2006) who globally compared syntagmatic and paradigmatic word space models by measuring their overlap. We go further by applying this method to a new type of models and by observing variation for words individually. We also identify zones with different degrees of variation.

The nearest neighbors of a given target word are words having the closest cosine similarity score with the target word. To compute the variation between models, we propose to compute the degree of nearest neighbors variation between two models. For two models M_1 and M_2, we first get the common vocabulary. We then compute the variation *var* by getting the common neighbors amongst the n nearest neighbors for each word in the two models such as:

$$var^n_{M_1,M_2}(w) = 1 - \frac{|neighb^n_{M_1}(w) \cap neighb^n_{M_2}(w)|}{n}$$

The value of n is important. To choose the most representative value, we selected a number of candidate values (1, 5, 10, 15, 25, 50 and 100). We found that for most pairs of models compared with this method, 25 was the value for which the variation scores had the highest correlation scores compared with other values of n across the entire vocabulary. In this work all comparisons use this value. We computed the variation for open-class parts of speech (henceforth POS) only i.e. nouns, verbs, adjectives and adverbs.

Parameters	Default	Values tested
Architecture	SG	CBOW
Corpus	BNC	ACL
Window size	5	1 to 10
Vectors dim.	100	50, 200, 300, 400, 500, 600
Context type	window	deps, deps+

Table 1. Parameters values used to train embeddings that are compared.

4 Experiments

4.1 Experiment setup

In this work, we use a DEFAULT model as a basis of comparison. Starting from this model, we trained new models by changing only one parameter at a time among the following parameters: architecture, corpus, window size, vectors dimensions, context type. We thus trained 19 models which will all be compared to the DEFAULT model. Although we compare less models and less parameters than other studies conducted on the evaluation of hyperparameters, we provide both a global and precise evaluation by computing the variation for each word of the model rather than evaluating selected pairs of words.

4.1.1 Default model

We trained our DEFAULT model using the widely used tool *word2vec* (Mikolov et al., 2013) with the default parameters values on the BNC corpus[1]. The parameters used were the following:

- architecture: Skip-gram,
- algorithm: negative sampling,
- corpus: BNC (written part only, made of about 90 million words),
- window size: 5,
- vector size: 100,
- negative sampling rate: 5,
- subsampling: 1e-3,
- iterations: 5.

The min-count parameter was set to a value of 100.

4.1.2 Variant models

5 different parameters are evaluated in this work: architecture, corpus, window size, vectors dimensions and context type. Using the default configuration, we then trained one model per possible parameter value stated in Table 1, e.g. we changed the value of the window size or the number of dimensions. We did not change more than one parameter when training the models since this work aims at evaluating the influence of a single parameter when training word embeddings. We chose the parameters to be investigated as well as their values based on selected studies that analyze the influence of parameters used when training DSMs (Baroni et al., 2014; Levy and Goldberg, 2014; Li et al., 2017; Melamud et al., 2016; Bernier-Colborne and Drouin, 2016; Sahlgren and Lenci, 2016; Chiu et al., 2016).

Models were trained on the BNC except for the ACL model which was trained on the ACL Anthology Reference corpus (Bird et al., 2008), a corpus made of about 100 million words. Both corpora were parsed using Talismane, a dependency parser developed by Urieli (2013). We trained models with dimensions ranging from 50 to 600 (DIM50 to DIM600 models). We used two different types of contexts: window-based and dependency-based contexts. For window-based models we used a window size from 1 to 10 (WIN1 to WIN10 models). For dependency-based models (DEPS and DEPS+) we used *word2vecf*, a tool developed by Levy and Goldberg (2014). This tool is an extension of *word2vec*'s Skip-gram. *Word2vecf* uses syntactic triples as contexts. We extracted the triples from the corpus using the scripts provided by Levy and Goldberg (2014)[2] and obtained triples such as *head modifier#reltype*. Prepositions were "collapsed" as described in Levy and Goldberg (2014). To investigate the influence of the triples used for training on the embeddings generated we decided to train with selected syntactic relations triples (DEPS+ model). We based our selection on Padó and Lapata (2007) work and chose to keep the following dependency relations: subject, noun modifier, object, adjectival modifier, coordination, apposition, prepositional modifier, predicate, verb complement. Prepositions were still collapsed *à la* Levy and Goldberg (2014) and the same was done for conjuctions.

4.2 Quantitative evaluation

To get an overview of the different models performance we first ran a partial quantitative evalu-

[1] http://www.natcorp.ox.ac.uk/

[2] https://bitbucket.org/yoavgo/word2vecf

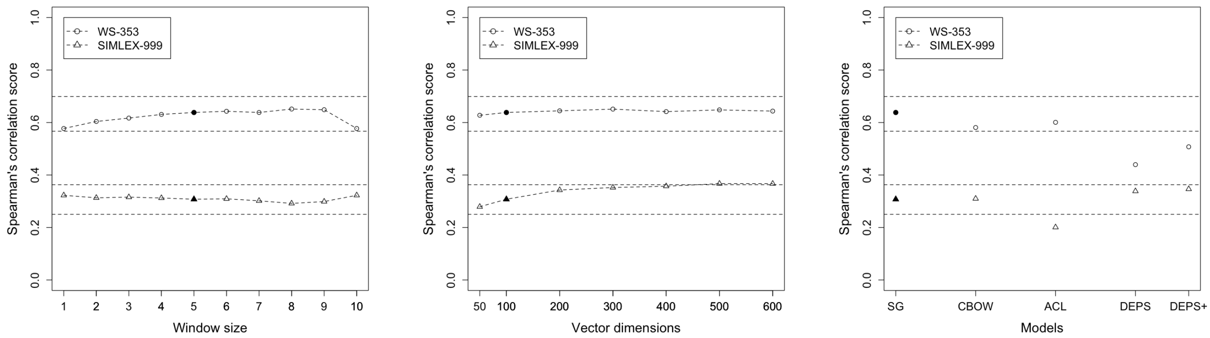

Figure 1. Evaluation results for all models on WordSim-353 and SimLex-999 with 95% confidence interval span computed from DEFAULT model (DEFAULT model is shown in bold).

ation. We used the toolkit[3] provided by Faruqui and Dyer (2014). This toolkit provides several benchmarks to test against the trained vectors. The evaluation is computed by ranking the different cosine scores obtained for each pair of the chosen dataset. The evaluation was run on WordSim-353 and Simlex-999 (Hill et al., 2015), two benchmarks commonly used for DSMs evaluation (e.g. see Levy and Goldberg (2014); Melamud et al. (2016)).

Figure 1 shows the performance of the different models on both test sets as well as the confidence interval for the DEFAULT model. We see that changing parameters creates differences in models performance and that this difference is generally not significant. Changing the architecture from Skip-gram to CBOW yields worse results for WordSim-353 than changing the corpus used for training. However, when testing on SimLex-999, performance is similar for the DEFAULT and CBOW models, while the ACL model performed worse. In a similar way, changing the training corpus gives better result on WordSim-353 than using a different type of contexts, as shown per the results of DEPS and DEPS+.

Performance is not consistent between the two benchmarks. DEPS and DEPS+ both yields the worst performance on WordSim-353 but at the same time their performance on SimLex-999 is better than most other models. The same is true for the WIN1 and WIN10 models. Increasing the vector dimensions gets slightly better performance, independently of the benchmark used. Increasing the window size gives better performance results for WordSim-353 but worse for SimLex-999. Dependency-based models performs the worst on WordSim-353.

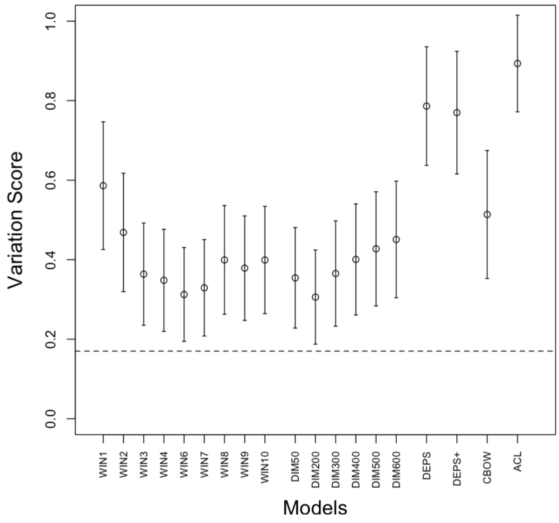

Figure 2. Mean variation value with standard deviation interval for all trained models compared to DEFAULT model. The dashed line corresponds to the mean variation value for the DEFAULT model trained 5 times with the exact same parameters.

This kind of evaluation is only performed on selected pairs of words and despite small differences in performance scores, larger differences may exist. In the next section we introduce a method that quantifies the variation between the different models trained by evaluating the distributional neighbors variation for every word in the corpus.

4.3 Qualitative evaluation

4.3.1 Exploring the variation

Figure 2 shows the mean variation score with the standard deviation span between the DEFAULT model and the 19 other models[4]. Since it is known

[3] https://github.com/mfaruqui/eval-word-vectors

[4] For models trained on the BNC, 27437 words were evaluated. When comparing DEFAULT to ACL the vocabulary size was smaller (10274) since the models were not trained on the same corpus.

there is inherent variation when training embeddings (Hellrich and Hahn, 2016), we measured the variation across 5 models using *word2vec* default settings. This variation is much lower than for the other models (0.17).

Training using the same parameters triggers variation with the DEFAULT model. Even for models varying the least, the variation is high with an average variation score of at least 0.3. This means that by changing only one parameter, among the 25 nearest neighbors of a given word about 1 neighbor out of 3 is different from one model to the other. Some variation scores are higher than 0.8 meaning that the two models compared are drastically different.

The ACL model is the one showing the highest variation. This is not surprising since it was trained on a specialized corpus. However, it is more surprising that DEPS and DEPS+ also display a very high variation. This could be explained by the fact that dependency-based and window-based models capture different type of semantic information (Levy and Goldberg, 2014).

Models showing the lowest variation are models with less drastic differences with the DEFAULT model, namely the vector size was changed from 100 to 200 or the window size from 5 to 6. A general tendency is that models trained with minimum and maximum values for a given parameter show more variation. Going back to the performance of the models (see Figure 1), we also notice that models having a performance score close to the DEFAULT model can still display a high variation. This is the case of the DIM600 model which had a performance score very close to the DEFAULT model on both benchmarks but still displays a variation higher than 0.4.

We observed that the variations between models do not follow the differences in performance on test sets shown in Figure 1. We measured the absence of correlation between the variation score and the performance scores on WordSim-353 ($\rho = -0.08$, $p = 0.78$) and Simlex-999 ($\rho = 0.25, p = 0.36$).

For every comparison shown in Figure 2, we can see a high standard deviation. This means that there are different behaviors across the lexicon and some words vary more than others. In the next section, we show how the frequency affects the variation across the different models.

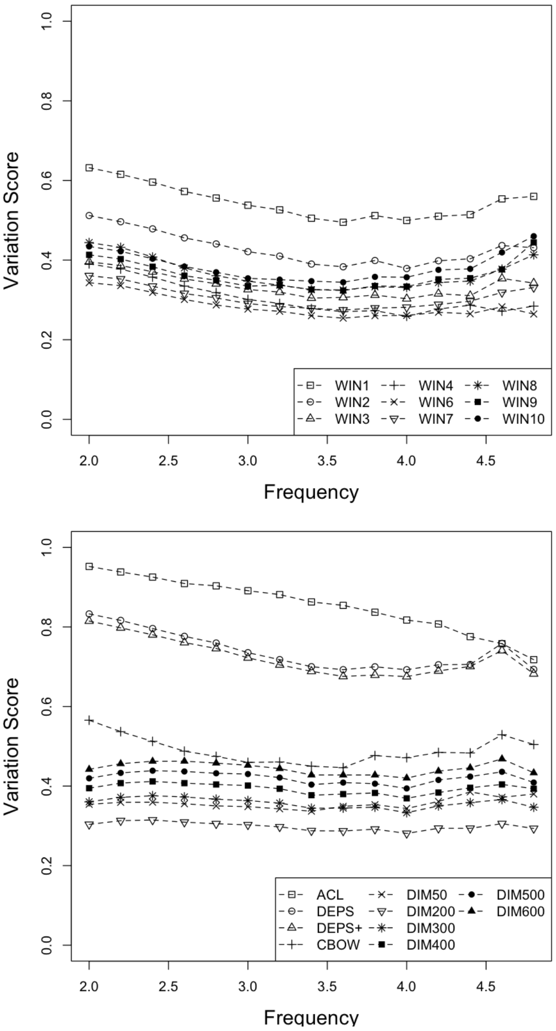

Figure 3. Effect of frequency on words variation.

4.3.2 Role of POS and frequency

To find some cues explaining the variation, we first investigated the interaction between the POS of a word and its variation score. However, we found for all models that the repartition of the variation was similar independently of the POS. This is surprising as we know embeddings perform differently across POS, especially when contexts vary.

We then investigated the role of the frequency in the variation. Figure 3 shows the average variation given the frequency of a word. For all window-based models, we observe a clear pattern: words in the mid-frequency range (1000 to 10000) display less variation than words in lower and higher frequency ranges. This is in line with Sahlgren and Lenci (2016) who showed that DSMs perform the best for medium to high-frequency ranges items. Models trained with different dimensions seem less affected by frequency. The variation is quite constant across all frequency ranges. CBOW, DEPS and DEPS+ follow the same pattern than

Model	Var.	Identified semantic classes
ACL	Low	numerals (*2nd*, *14th*, *10th*...) nationalities (*hungarian*, *french*, *danish*, *spanish*...) time nouns (*afternoon*, *week*, *evening*...)
	High	specialized lexicon (*embedded*, *differential*, *nominal*, *probabilistic*, *patch*, *spell*, *string*, *graph*...)
DIM200	Low	numerals (*40th*, *15th*...) nationalities (*hungarian*, *dutch*, *french*, *spanish*...) family nouns (*grandparent*, *sister*, *son*, *father*...)
	High	generic adjectives (*all*, *near*, *very*, *real*...) polysemic nouns (*field*, *marker*, *turn*, *position*...)

Table 2. Words showing lowest and highest variation for ACL and DIM200 compared to DEFAULT.

the window models, with a variation less high for medium frequency words. ACL[5] display a very high variation for low frequency words but the variation decreases with frequency.

4.3.3 Getting a preview of the variation

The variation measure can also be used to examine more local differences. For example, for given pairs of models we can easily identify which words show the most extreme variation values. We did this for two of our models: ACL which shows the highest variation and DIM200 which shows the lowest variation. Table 2 shows a few of the most stable and unstable words. It appears that different semantic classes emerge in each case. It seems that these classes correspond to dense clusters, each word having all others as close neighbor. Some of these clusters remain the same across the two pairs of models (e.g. nationality adjectives) while other clusters are different. In the ACL model, we find a cluster of time nouns while in the DIM200 model we find family nouns. We see that words varying the most for the specialized corpus are words carrying a specific meaning (e.g. *nominal*, *graph*). We also find that words with a high variation score are highly polysemic or generic in the DIM200 model (e.g. *field*, *marker*). In the future we want to analyze the impact of the degree of polysemy on the variation score along with other characteristics of words.

5 Conclusion

This work compared 19 models trained using one different parameter to a default model. We measured the differences between these models with benchmark test sets and a methodology which does not depend on the subjectivity and limited scope of benchmark test sets. Test sets show marginal differences while neighbors variation revealed that at least one third of the nearest 25 neighbors of a word are different from one model to the other. In addition it appears that the parameters have a different impact depending on the way differences are measured.

We saw that the variation is not affecting all words of the semantic space equally and we found features which help identify some areas of (in)stability in the semantic space. Words having a low and high frequency range have a tendency to display more variation. Words in the medium frequency range show more stability. We also found word features that could play a role in the variation (polysemy, genericity, semantic clusters etc.). These features can help understanding what really changes when tuning the parameters of word embeddings and give us more control over those effects.

In further work, we want to extend our analysis to more parameters. We especially want to see if the observations made in this study apply to models trained with specialized corpora or corpora of different sizes. We also want to distinguish features that will help classify words displaying more or less variation and qualify the variations themselves.

Acknowledgments

We would like to thank the Student Research Workshop for setting up the mentoring program that we benefited from. We especially thank our mentor, Shashi Narayan, for his very helpful comments. We are also thankful to the anonymous reviewers for their suggestions and comments.

[5] The variation for ACL was measured on a smaller vocabulary set. The frequency used in Figure 3 is the one from the BNC.

Experiments presented in this paper were carried out using the OSIRIM platform that is administered by IRIT and supported by CNRS, the Région Midi-Pyrénées, the French Government, and ERDF (see http://osirim.irit.fr/site/en).

References

Maria Antoniak and David Mimno. 2018. Evaluating the Stability of Embedding-based Word Similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

Fatemeh Torabi Asr, Jon A. Willits, and Michael N. Jones. 2016. Comparing Predictive and Co-occurrence Based Models of Lexical Semantics Trained on Child-directed Speech. In *Proceedings of the 37th Meeting of the Cognitive Science Society*.

Marco Baroni, Georgiana Dinu, and German Kruszewski. 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*, pages 238–247, Baltimore, Maryland, USA.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, EMNLP 2011*, pages 1–10, Edinburgh, Scotland, UK.

Gabriel Bernier-Colborne and Patrick Drouin. 2016. Evaluation of distributional semantic models: a holistic approach. In *Proceedings of the 5th International Workshop on Computational Terminology*, pages 52–61, Osaka, Japan.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proceedings of Language Resources and Evaluation Conference (LREC 08)*, Marrakesh, Morrocco.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to Train Good Word Embeddings for Biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174, Berlin, Germany.

Manaal Faruqui and Chris Dyer. 2014. Community Evaluation and Exchange of Word Vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24, Baltimore, Maryland, USA.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing Search in Context: The Concept Revisited. In *ACM Transactions on Information Systems*, volume 20, page 116:131.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501, Berlin, Germany.

Johannes Hellrich and Udo Hahn. 2016. Bad Company - Neighborhoods in Neural Embedding Spaces Considered Harmful. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2785–2796, Osaka, Japan.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics*, 41:665–695.

Gabriella Lapesa and Stefan Evert. 2014. A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection. *Transactions of the Association for Computational Linguistics*, 2:531–545.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 302–308, Baltimore, Maryland, USA.

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.

Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017. Investigating Different Syntactic Context Types and Context Representations for Learning Word Embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2411–2421.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The Role of Context Types and Dimensionality in Learning Word Embeddings. In *Proceedings of NAACL-HLT 2016*, San Diego, California.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Neha Nayak, Gabor Angeli, and Christopher D. Manning. 2016. Evaluating Word Embeddings Using a Representative Suite of Practical Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 19–23, Berlin, Germany.

Sebastian Padó and Mirella Lapata. 2007. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University, Sweden.

Magnus Sahlgren and Alessandro Lenci. 2016. The Effects of Data Size and Frequency Range on Distributional Semantic Models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Austin, Texas.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics (LDL-2015)*, pages 64–69, Beijing, China.

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.

Assaf Urieli. 2013. *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. Ph.D. thesis, University of Toulouse, France.