There's no 'Count or Predict' but task-based selection for distributional models

Martin Riedl and Chris Biemann
Universität Hamburg, Germany
{riedl,biemann}@informatik.uni-hamburg.de

Abstract

In this paper, we investigate the differences between prediction-based (word2vec), dense count-based (GloVe) and sparse count-based (JoBimText) semantic models. We evaluate the models, which were selected because they can all be computed efficiently on large data, based on word similarity tasks and a semantic ranking task both for verbs and nouns. We demonstrate that prediction-based models yield higher scores than the other two models at determining a similarity score between two words. To the contrary, sparse count-based methods perform best in the ranking task. Further, sparse count-based methods benefit more from linguistically informed contexts, such as dependency relations. In summary, we highlight differences of popular distributional semantic representations and derive recommendations for their usage.

1 Introduction

With the steady growth of textual data, NLP methods are required that are able to process the data efficiently. In this paper, we focus on efficient methods that are targeted to compute distributional models that are based on the distributional hypothesis of Harris (1951). This hypothesis claims that words occurring in similar contexts tend to have similar meanings. In order to implement this hypothesis, early approaches (Hindle, 1990; Grefenstette, 1994; Lin, 1997) represented words using count-based vectors of the context. However, such representations are very sparse, require a lot of memory and are not very efficient. In the last decades, methods have been developed that transform such sparse representations into dense representations mainly using matrix factorization. With word2vec (Mikolov et al., 2013), an efficient prediction-based method was introduced, which also represents words with a dense vector. However, also sparse and count-based methods have been proposed that allow an efficient computation, e.g. (Kilgarriff et al., 2004; Biemann and Riedl, 2013). A more detailed overview of semantic representations can be found in (Lund and Burgess, 1996; Turney and Pantel, 2010; Ferrone and Zanzotto, 2017).

In this work, we explore different aspects between three different methods for computing similarities: similarity computations that use sparse symbolic vectors for similarity computations, dense vector based methods that are based on co-occurrences and prediction-based methods. For this, we aim to focus on efficiently computable methods and selected SKIP and CBOW from word2vec, GloVe and JoBimText. Based on these methods we want to explore different aspects: 1) which method performs the best global similarity scoring using word pair similarity datasets 2) which method performs the best local ranking of most similar terms for a query term 3) which context works best for the different methods and 4) are there differences in the performance when evaluating on verbs and nouns.

2 Related Work

One of the first comparisons between count-based and prediction-based distributional models was performed by Baroni et al. (2014). For this, they consider various tasks and show that prediction-based word

embeddings outperform sparse count-based methods and dense count-based methods used for computing distributional semantic models. The evaluation is performed on datasets for relatedness, analogy, concept categorization and selectional preferences. The majority of word pairs considered for the evaluation consists of noun pairs. However, Levy and Goldberg (2014b) showed that dense count-based methods, using PPMI weighted co-occurrences and SVD, approximates neural word embeddings. Levy et al. (2015) showed in an extensive study the impact of various parameters and show the best performing parameters for these methods. The study reports results for various datasets for word similarity and analogy. However, they do not evaluate the performance on local similarity ranking tasks and omit results for pure count-based semantic methods. Claveau and Kijak (2016) performed another comparison of various semantic representation using both intrinsic and extrinsic evaluations. They compare the performance of their count-based method to dense representations and prediction-based methods using a manually crafted lexicon, SimLex and an information retrieval task. They show that their method performs better on the manually crafted lexicon than using word2vec. For this task, they also show that a word2vec model computed on a larger dataset yields inferior results than models computed on a smaller corpus, which is contrary to previous findings, e.g. (Banko and Brill, 2001; Gorman and Curran, 2006; Riedl and Biemann, 2013). Based on the SimLex task and the extrinsic evaluation they show comparable performance to the word2vec model computed on a larger corpus.

In this work, we do not focus on the best performing systems for each dataset, like e.g. retrofitting embeddings (Kiela et al., 2015; Rothe and Schütze, 2015), but want to carve out the difference of existing methods for computing distributional similarities.

3 Methods for Distributional Semantics

For the efficient and scalable similarity computation, we select SKIP and CBOW from word2vec as prediction-based, GloVe as dense count-based[1] and JoBimText as sparse count-based method.

Word2Vec

We use the SKIP-gram model, which predicts for a word the neighboring words within a symmetric window of w. Considering the CBOW model, a word is predicted by its neighboring words. For the computation, we use the implementation by Mikolov et al. (2013)[2]. In addition, we use the extension of word2vec, which was introduced by Levy and Goldberg (2014a)[3] and allows to use arbitrary contexts for computing dense vector representations for similarity computations.

Global Vectors (GloVe)

As dense count-based approach, we select GloVe (Pennington et al., 2014).[4] GloVe achieves its representation based on logarithmic co-occurrences between words and context. This representation is learned using matrix factorization methods.

JoBimText (JBT)

We consider JoBimText (Biemann and Riedl, 2013) as symbolic count-based method[5] that produces word similarities encoded in a distributional thesaurus (DT, cf. Lin, 1998). The method is based on a term-context representation and can handle arbitrary contexts. For an efficient computation it considers

[1] In this study, we consider GloVe as a dense count-based method. Although GloVe uses a classifier in order to optimize its cost function, it is based on co-occurrence statistics and does not predict contexts from words directly, as performed in word2vec.

[2] `https://code.google.com/archive/p/word2vec/`

[3] `https://bitbucket.org/yoavgo/word2vecf`

[4] `https://nlp.stanford.edu/projects/glove/`

[5] `http://sf.net/p/jobimtext/`

several pruning techniques and uses Lexicographer's Mutual Information (LMI) (Evert, 2005) to determine relevant contexts per word. In addition, we show results when using the frequency (freq) for ranking which turned out to perform well in Padró et al. (2014). For each term the 1000 contexts with the highest LMI score or frequency are kept. Additionally, contexts are removed that co-occur with more than 1000 terms. The similarity score is only computed between terms that share at least one context and is based on the logarithmic sum of the reciprocal value of the number of terms a context co-occurs (log). Furthermore, we computed similarity scores by using solely the number of contexts two terms share (one).

4 Experimental Setting

For performing the studies, we rely on two different evaluation methods. First, we show results based on datasets that contain averaged similarity scores for word pairs annotated by humans. We use SimLex-999 (Hill et al., 2015), which consists of 999 word pairs, formed by 666 noun, 222 verb and 111 adjective pairs and the SimVerb-3500 dataset (Gerz et al., 2016) which comprises of 3500 verb pairs. The evaluation scores are computed using the Spearman rank correlation coefficient between the gold standard scores and the similarity scores obtained with the semantic methods. These evaluations validate the ability of semantic methods to provide similarity scores that demonstrate the performance for a global ranking between word pairs scores. We name this task as a 'global ranking task' as the semantic models have to provide a score between two given word pairs and the evaluation score is computed by the correlation between similarity scores given by the model and averaged similarity scores given by humans.

In a so-called local ranking task, we will evaluate how well semantic models can retrieve the most similar words for a given term. For this, we sample 1000 low-, middle- and high frequent nouns and verbs. In order to compute the semantic similarities between the most similar terms, we use the WordNet Path measure (Pedersen et al., 2004) and perform an evaluation that is similar to the one used by Biemann and Riedl (2013). This Path measure is the shortest reciprocal distance + 1 between two words based on the IS-A path.

The computation of the various models is performed using a dump of English Wikipedia that comprises of 35 million sentences. The similarities are computed on raw tokenized text, then on lemmatized and POS-tagged tagged text and finally using dependency parses[6] as context representation, which has been shown to work well for computing similarities (Lin, 1997; Biemann and Riedl, 2013; Levy and Goldberg, 2014a). Whereas the tokens and lemmas can be processed with all methods, the dependency parses can only be used with a modification of word2vec (Levy and Goldberg, 2014a) and JBT.

5 Word Similarity Evaluation

In this section, we show the Spearman correlations for the different models using SimLex and SimVerb[7]. First, we perform the computation of the models on raw text (see Table 1). Using various parameters for both word2vec models[8], we observe the best results for the SimLex dataset when computing both SKIP and CBOW with 500 dimensions, using random sampling ($s = 1E^{-5}$), 10 negative examples and a word window size of 1 (W1). This is in line with Melamud et al. (2016), who mostly obtain the highest scores for word similarity tasks when using a comparably high number of dimensions. For GloVe we obtain the best results with the same parameters as for word2vec: we use a window size of 1 and 500 dimensions.[9] The CBOW model performs best on the SimVerb dataset but does not yield the best scores for the verbs in SimLex. However, we could not detect much differences between the two sets, as we observe a correlation of 0.9177 for 90 verb pairs that are shared in both datasets. GloVe performs best on

[6] We use the Stanford dependency parser (de Marneffe et al., 2006)

[7] All word pairs not contained in the model are scored with zero.

[8] We tested different values for random sampling ($s = \{0, 1^{-5}\}$), dimension size ($d = \{100, 200, 500\}$), window size ($w = \{1, 5, 10, 15\}$) and negative examples ($n = \{0, 5, 10\}$

[9] We tested various window sizes ($w = \{1, 2, 5, 10, 15\}$) and various number of dimensions ($d = \{50, 100, 200, 500\}$).

	Method	SimLex all	SimLex NN	SimLex VB	SimLex JJ	SimVerb
raw text	SKIP W1 100	0.3105	0.3488	0.1630	0.4345	0.2113
	SKIP W1 500	**0.3908**	0.4223	0.2616	0.5324	0.2656
	SKIP W5 500	0.3364	0.3758	0.1741	0.4531	0.2335
	CBOW W1 100	0.3159	0.3529	0.1683	0.4575	0.2121
	CBOW W1 500	0.3901	0.4193	0.2638	0.5284	**0.2677**
	CBOW W5 500	0.3427	0.3821	0.1698	0.4798	0.2339
	GloVe W1 100	0.2359	0.2367	0.1633	0.3567	0.1565
	GloVe W1 500	0.3055	0.2832	**0.2679**	**0.5359**	0.1903
	JBT freq one	0.2940	0.3934	0.0576	0.3742	0.1469
	JBT freq log	0.3085	0.4032	0.0726	0.4071	0.1599
	JBT LMI one	0.3140	0.4113	0.0741	0.4144	0.1763
	JBT LMI log	0.3306	**0.4231**	0.0942	0.4328	0.1889
lemma	SKIP W1 500	**0.4024**	0.4465	0.2041	0.5347	0.3012
	CBOW W1 500	0.3997	0.4409	0.2037	0.5353	**0.3023**
	GloVe W1 500	0.3751	0.3786	**0.2411**	**0.5437**	0.3017
	JBT LMI log	0.3784	**0.4583**	0.2100	0.3906	0.2961
dep.	SKIP W1 500	0.3480	0.4089	**0.2678**	0.3220	0.2552
	JBT LMI log	**0.3869**	**0.4475**	0.2649	**0.3841**	**0.3276**

Table 1: Spearman correlation with SimLex and SimVerb for models computed on tokenized text.

adjectives and verbs for the SimLex dataset, but cannot reach the highest scores on the SimVerb dataset. Although, JBT is not optimized for global similarity scoring, as it does not compute normalized similarity scores between two terms, the correlation scores are are highest for the SimLex's nouns. In contrast to Padró et al. (2014), computing JBT by using the frequency (JBT freq log and JBT freq one) for ranking relevant contexts does not yield the best performance. Here the highest scores are achieved using LMI with a logarithmic scoring, which confirms the findings by Riedl (2016). As the selected parameters also performed best for the lemmatized and dependency-parsed data, we restrict the presentation of results to this setting in the remainder.

Inspecting the correlation scores on the lemmatization-based models equipped with POS-tags we observe a similar trend. In general, we examine higher scores than with raw text. For the entire SimLex and SimVerb dataset we again observe the best performance with the prediction-based models. In contrast to the previous evaluations, the scores from the JBT LMI log are closer to the highest correlation scores of CBOW. Again the best scores for verbs and adjectives are retrieved using GloVe.

Using dependency parses as context, we spot the best performance with JBT LMI log. For the SimVerb dataset, we get even higher results than using the best performing CBOW model using lemmas. Using the dependency-based SKIP model performs well for the SimLex verbs, but apart from that cannot even outperform the word2vec models computed on raw text.

6 Word Ranking Evaluation

In this section, we use the WordNet-based evaluation in order to show the performance of the methods based on a local similarity ranking. Here, we focus on the methods with its best performing parameters and show results for lemma and POS-based models and dependency-based models. Table 2 shows results for nouns and verbs for different frequent bands for the top $N = \{1, 5, 10, 50, 100\}$ highest ranked words. For low- and mid-frequent nouns the best scores up to the top 10 most similar nouns are achieved with the SKIP model. Beyond considering more than the 10 most similar terms the JBT model performs best. Whereas, up to the 50 most similar nouns, the performance of the different models is comparable, we observe performance drops for the top 100 ranked words for GloVe and SKIP in comparison to JBT.

	Method	freq	1	5	10	50	100
nouns	SKIP W1 500	high	0.3613	0.2759	0.2373	0.1326	0.0751
	GloVe W1 500	high	0.2612	0.2412	0.2266	0.1821	0.1439
	JBT LMI log	high	**0.3821**	**0.3007**	**0.2649**	**0.2013**	**0.1802**
	SKIP W1 500	mid	**0.2480**	**0.1887**	0.1649	0.1138	0.0736
	GloVe W1 500	mid	0.2270	0.1612	0.1429	0.1133	0.0980
	JBT LMI log	mid	0.2377	0.1828	**0.1660**	**0.1362**	**0.1249**
	SKIP W1 500	low	**0.1891**	**0.1461**	0.1320	0.0816	0.0477
	GloVe W1 500	low	0.1423	0.1174	0.1092	0.0864	0.0618
	JBT LMI log	low	0.1798	0.1508	**0.1392**	**0.1166**	**0.1062**
verbs	SKIP W1 500	high	0.4718	0.3384	0.2866	0.1574	0.0956
	GloVe W1 500	high	**0.4882**	**0.3683**	0.3217	0.2462	0.1864
	JBT LMI log	high	0.4611	0.3651	**0.3286**	**0.2686**	**0.2498**
	SKIP W1 500	mid	**0.3689**	0.2524	0.2139	0.1129	0.0676
	GloVe W1 500	mid	0.3286	0.2352	0.2111	0.1741	0.1481
	JBT LMI log	mid	0.3437	**0.2705**	**0.2520**	**0.2167**	**0.2052**
	SKIP W1 500	low	0.2481	0.1766	0.1469	0.0653	0.0354
	GloVe W1 500	low	0.1950	0.1768	0.1665	0.1276	0.0878
	JBT LMI log	low	**0.2544**	**0.2246**	**0.2140**	**0.1904**	**0.1773**

Table 2: Results of the lemma-based models for the WordNet-based evaluation showing results for the top N most similar words.

Considering the high frequent nouns the best performance is always obtained with the JBT model. For verbs Glove achieves the highest scores for when using the top 1 to 5 most similar terms for high frequent verbs. However, similar to the results based on nouns the best performance for the 10, 50 and 100 most similar terms ist gained using the JBT model.

Using dependency parses as context, we obtain the overall highest scores using JBT (see Table 3). Again, the modified SKIP model cannot compete with the count-based method and performs even inferior to the lemma and POS-tag based models.

	Method	freq	1	5	10	50	100
nouns	SKIP W1 500	high	0.3760	0.2889	0.2546	0.1907	0.1665
	JBT LMI log	high	**0.4004**	**0.3143**	**0.2776**	**0.2148**	**0.1929**
	SKIP W1 500	mid	0.1990	0.1630	0.1507	0.1308	0.1216
	JBT LMI log	mid	**0.2898**	**0.2214**	**0.1989**	**0.1585**	**0.1451**
	SKIP W1 500	low	0.1420	0.1288	0.1230	0.1061	0.0913
	JBT LMI log	low	**0.2634**	**0.2012**	**0.1815**	**0.1431**	**0.1300**
verbs	SKIP W1 500	high	0.4073	0.3011	0.2656	0.2153	0.1973
	JBT LMI log	high	**0.4948**	**0.3729**	**0.3305**	**0.2660**	**0.2494**
	SKIP W1 500	mid	0.2842	0.2201	0.2012	0.1770	0.1683
	JBT LMI log	mid	**0.3980**	**0.3026**	**0.2699**	**0.2193**	**0.2072**
	SKIP W1 500	low	0.2076	0.1781	0.1714	0.1589	0.1482
	JBT LMI log	low	**0.3214**	**0.2597**	**0.2363**	**0.2007**	**0.1919**

Table 3: WordNet Path scores for semantic models that use dependency parses as context

7 Data Analysis

When examining the most similar words, we detected some further properties of each models. Exemplarily, we show the five most similar terms to the noun "access" using the POS-tagged and lemmatized models in Table 4. First, we observe that not all similar terms are nouns and in addition it seems, that

SKIP W1 500		GloVe W1 500		JBT LMI log	
access#VB	0.73	accessible#JJ	0.80	connection#NN	27.08
accessible#JJ	0.65	provide#VB	0.80	connectivity#NN	22.72
accessibility#RB	0.64	allow#VB	0.78	link#NN	14.62
accessibility#NN	0.64	enable#VB	0.78	exposure#NN	13.58
wifus#NN	0.61	available#JJ	0.75	entry#NN	12.11

Table 4: Most similar words for the noun "access".

in comparison to JBT, SKIP and GloVe favor less frequent words. These effects are explored in the following.

Frequency of Similarities

To explore the frequencies of similar words, we compute the average frequency for the top $N = \{1, 10, 100, 200\}$ most similar words for the sampled candidates. In addition, we use the relative frequency in relation to the frequency of the queried word. Among all frequency bands and for verbs and nouns we observe a consistent pattern, as shown in Figure 1. For nouns, the SKIP and CBOW similar

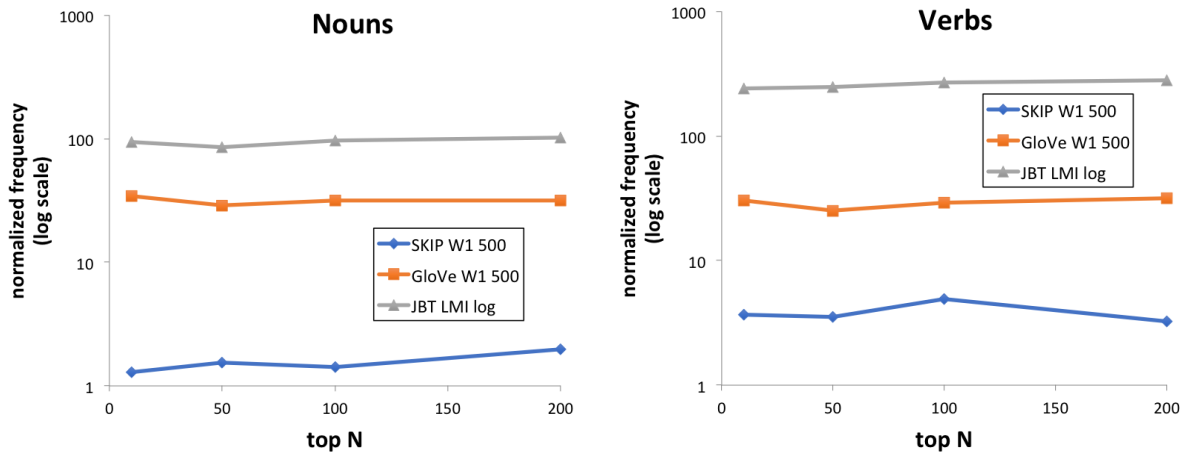

Figure 1: Normalized average frequency for the top N most similar words for 3000 nouns (left) and 3000 verbs (right) for the semantic models computed using lemmas and POS information.

words are on average 3 times more frequent than the queried term. Schnabel et al. (2015) also describe that the frequency of the similarities stay in the same frequency region and attribute this effect to the cosine similarity. Using GloVe the similar nouns are on average 20 times more frequent and for JBT we retrieve words that are on average 100 times more frequent than the queried word. For verbs, we obtain consistently higher average similarities. However, the pattern is similar to the one observed with nouns.

Keeping the same POS

Next, we examine the stability of the most similar terms in respect to the POS of the candidate term. For this we use the lemmatized and dependency-based models in order to determine the percentage of similar words that keep the same POS-tag. This reveals how good the most similar words stay in their same grammatical function and is e.g. relevant when trying to replace unknown words in machine translation or for POS-tagging and dependency parsing, where the grammatical function should be the same. We show the ratio of all 3000 selected nouns for the top $N = \{1, 10, 50, 100\}$ first entries in Table 5. Using the lemmatized models, we obtain the highest POS consistency among the similar terms using JBT, followed by GloVe and SKIP after a large margin. The dependency parses-based models show a

context	Method	1	10	50	100
lemmas	SKIP W1 500	0.6077	0.5550	0.5060	0.4834
	GloVe W1 500	0.5137	0.5382	0.5446	0.5408
	JBT LMI log	**0.9997**	**0.8969**	**0.8762**	**0.8650**
dependencies	SKIP W1 500	0.9703	**0.9687**	**0.9559**	**0.9450**
	JBT LMI log	**1.0000**	0.9486	0.9258	0.9139

Table 5: Percentage of the top N most similar terms for nouns that keep the same POS-tag

different trend: here the SKIP model pertains mostly in the same POS class and yields higher scores than the JBT approach.

8 Conclusion

In this paper, we have shown the differences between efficiently computable semantic methods of three different classes: sparse count based, dense count-based and dense prediction-based models. For global similarity ranking, we advise using the SKIP or CBOW method when processing raw and lemmatized text, which obtain the best overall results on SimLex and SimVerb. In general, we observe performance increases when using lemmatized text rather than raw text. Using dependency parses, only the JBT model improves and yields the best result for verbs. Using SKIP with dependency parse context no improvements are gained and the performance is mostly worse than using raw text. Based on the local similarity ranking, we recommend using the JBT model, which yields the best overall performance both for nouns and verbs. In addition, using dependency parses as context results in further improvements. When requiring more than the top 50 most similar terms for query term, we would not advise using the dense vector representations, as both GloVe and word2vec perform poorly. Based on tasks where words in text should be replaced with words of the same grammatical function (e.g. lexical substitution, machine translation) using either JBT with all context or SKIP using dependency parses is advised, as word and lemma based GloVe and SKIP favor similarities to words of another POS. Furthermore, SKIP and CBOW favor to extract similar terms of the same frequency as the queried word, whereas similar words obtained with JBT are on average 176 times more frequent. For tasks like text simplification however, providing more frequent words is favored as frequent words are more likely to be known.

In future work, we would like to evaluate further methods like Random Indexing, SVD-based methods, and DM (Padó and Lapata, 2007) and enhance the evaluation by extrinsic ones. In addition, we want to conceive a method that integrates the advantages of all discussed methods.

References

Banko, M. and E. Brill (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, pp. 26–33.

Baroni, M., G. Dinu, and G. 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 (Volume 1: Long Papers)*, Baltimore, MA, USA, pp. 238–247.

Biemann, C. and M. Riedl (2013). Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling 1*(1), 55–95.

Claveau, V. and E. Kijak (2016). Direct vs. indirect evaluation of distributional thesauri. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 1837–1848.

de Marneffe, M.-C., B. Maccartney, and C. D. Manning (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, LREC 2006, Genova, Italy, pp. 449–454.

Evert, S. (2005). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph. D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Ferrone, L. and F. M. Zanzotto (2017). Symbolic, distributed and distributional representations for natural language processing in the era of deep learning: a survey. *CoRR abs/1702.00764*.

Gerz, D., I. Vulić, F. Hill, R. Reichart, and A. Korhonen (2016). SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 2173–2182.

Gorman, J. and J. R. Curran (2006). Scaling distributional similarity to large corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL 2006, Sydney, Australia, pp. 361–368.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Norwell, MA, USA: Kluwer Academic Publishers.

Harris, Z. S. (1951). *Methods in Structural Linguistics*. Chicago: University of Chicago Press.

Hill, F., R. Reichart, and A. Korhonen (2015). SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics 41*(4), 665–695.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, ACL 1990, Pittsburgh, PA, USA, pp. 268–275.

Kiela, D., F. Hill, and S. Clark (2015, September). Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 2044–2048.

Kilgarriff, A., P. Rychlý, and D. T. Pavel Smrz (2004). The sketch engine. In *Proceedings of the 11th EURALEX International Congress*, Lorient, France, pp. 105–115.

Levy, O. and Y. Goldberg (2014a). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, MD, USA, pp. 302–308.

Levy, O. and Y. Goldberg (2014b). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, NIPS 2014, pp. 2177–2185.

Levy, O., Y. Goldberg, and I. Dagan (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics 3*, 211–225.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL 1998/EACL 1997, Madrid, Spain, pp. 64–71.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, COLING 1998, Montreal, Quebec, Canada, pp. 768–774.

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

Melamud, O., D. McClosky, S. Patwardhan, and M. Bansal (2016). The role of context types and dimensionality in learning word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-2016, San Diego, CA, USA, pp. 1030–1040.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Machine Learning*, ICLR 2013, Scottsdale, AZ, USA, pp. 1310–1318.

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

Padró, M., M. Idiart, A. Villavicencio, and C. Ramisch (2014). Nothing like good old frequency: Studying context filters for distributional thesauri. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2014, Doha, Qatar, pp. 419–424.

Pedersen, T., S. Patwardhan, and J. Michelizzi (2004). WordNet::Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, Boston, MA, USA, pp. 38–41.

Pennington, J., R. Socher, and C. D. Manning (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2014, Doha, Quatar, pp. 1532–1543.

Riedl, M. (2016). *Unsupervised Methods for Learning and Using Semantics of Natural Language*. Ph. D. thesis, Technische Universität Darmstadt, Germany.

Riedl, M. and C. Biemann (2013). Scaling to large^3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2013, Seattle, WA, USA, pp. 884–890.

Rothe, S. and H. Schütze (2015). AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, ACL 2015, Beijing, China, pp. 1793 – 1803.

Schnabel, T., I. Labutov, D. Mimno, and T. Joachims (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 298–307.

Turney, P. D. and P. Pantel (2010, January). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research 37*(1), 141–188.