Scaling to Large^3 Data: An efficient and effective method to compute Distributional Thesauri

Martin Riedl and **Chris Biemann**
FG Language Technology
Computer Science Department, Technische Universität Darmstadt
Hochschulstrasse 10, D-64289 Darmstadt, Germany
{riedl,biem}@cs.tu-darmstadt.de

Abstract

We introduce a new highly scalable approach for computing Distributional Thesauri (DTs). By employing pruning techniques and a distributed framework, we make the computation for very large corpora feasible on comparably small computational resources. We demonstrate this by releasing a DT for the whole vocabulary of Google Books syntactic n-grams. Evaluating against lexical resources using two measures, we show that our approach produces higher quality DTs than previous approaches, and is thus preferable in terms of speed and quality for large corpora.

1 Introduction

Using larger data to estimate models for machine learning applications as well as for applications of Natural Language Processing (NLP) has repeatedly shown to be advantageous, see e.g. (Banko and Brill, 2001; Brants et al., 2007). In this work, we tackle the influence of corpus size for building a distributional thesaurus (Lin, 1998). Especially, we shed light on the interaction of similarity measures and corpus size, as well as aspects of scalability.

We shortly introduce the JoBimText framework for distributional semantics and show its scalability for large corpora. For the computation of the data we follow the MapReduce (Dean and Ghemawat, 2004) paradigm. The computation of similarities between terms becomes challenging on large corpora, as both the numbers of terms to be compared and the number of context features increases. This makes standard similarity calculations as proposed in (Lin, 1998; Curran, 2002; Lund and Burgess, 1996; Weeds et al., 2004) computationally infeasible. These approaches first calculate an information measure between each word and the according context and then calculate the similarity between all words, based on the information measure for all shared contexts.

2 Related Work

A variety of approaches to compute DTs have been proposed to tackle issues regarding size and runtime. The reduction of the feature space seems to be one possibility, but still requires the computation of such reduction cf. (Blei et al., 2003; Golub and Kahan, 1965). Other approaches use randomised indexing for storing counts or hashing functions to approximate counts and measures (Gorman and Curran, 2006; Goyal et al., 2010; Sahlgren, 2006). Another possibility is the usage of distributed processing like MapReduce. In (Pantel et al., 2009; Agirre et al., 2009) a DT is computed using MapReduce on 200 quad core nodes (for 5.2 billion sentences) respectively 2000 cores (1.6 Terawords), an amount of hardware only available to commercial search engines. Whereas Agirre uses a χ^2 test to measure the information between terms and context, Pantel uses the Pointwise Mutual Information (PMI). Then, both approaches use the cosine similarity to calculate the similarity between terms. Furthermore, Pantel describes an optimization for the calculation of the cosine similarity. Whereas Pantel and Lin (2002) describe a method for sense clustering, they also use a method to calculate similarities between terms. Here, they propose a pruning scheme similar to ours, but do not explicitly evaluate its effect.

The evaluation of DTs has been performed in extrinsic and intrinsic manner. Extrinsic evaluations have been performed using e.g. DTs for automatic

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 884–890,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

set expansion (Pantel et al., 2009) or phrase polarity identification (Goyal and Daumé, 2011). In this work we will concentrate on intrinsic evaluations: Lin (1997; 1998) introduced two measures using WordNet (Miller, 1995) and Roget's Thesaurus. Using WordNet, he defines context features (synsets a word occurs in Wordnet or subsets when using Roget's Thesaurus) and then builds a gold standard thesaurus using a similarity measure. Then he evaluates his generated Distributional Thesaurus (DT) with respect to the gold standard thesauri. Weeds et al. (2004) evaluate various similarity measures based on 1000 frequent and 1000 infrequent words. Curran (2004) created a gold standard thesaurus by manually extracting entries from several English thesauri for 70 words. His automatically generated DTs are evaluated against this gold standard thesaurus using several measures. We will report on his measure and additionally propose a measure based on WordNet paths.

3 Building a Distributional Thesaurus

Here we present our scalable DT algorithm using the MapReduce paradigm, which is divided into two parts: The holing system and a computational method to calculate distributional similarities. A more detailed description, especially for the MapReduce steps, can be found in (Biemann and Riedl, 2013).

3.1 Holing System

The holing operation splits an observation (e.g. a dependency relation) into a pair of two parts: a term and a context feature. This captures their first-order relationship. These pairs are subsequently used for the computation of the similarities between terms, leading to a second-order relation. The representation can be formalized by the pair <x,y> where x is the term and y represents the context feature. The position of x in y is denoted by the hole symbol '@'. As an example the dependency relation *(nsub;gave$_2$;I$_1$)* could be transferred to *<gave$_2$,(nsub;@;I$_1$)>* and *<I$_1$,(nsub;gave$_2$;@)>*. This representation scheme is more generic then the schemes introduced in (Lin, 1998; Curran, 2002), as it allows to characterise pairs by several holes, which could be used to learn analogies, cf. (Turney and Littman, 2005).

3.2 Distributional Similarity

First, we count the frequency for each first-order relation and remove all features that occur with more than w terms, as these context features tend to be too general to characterise the similarity between other words (Rychlý and Kilgarriff, 2007; Goyal et al., 2010, cmp.). From this. we calculate a significance score for all first-order relations. For this work, we implemented two different significance measures: Pointwise Mutual Information (PMI): $PMI(term, feature) = \log_2(\frac{f(term,feature)}{f(term)f(feature)})$ (Church and Hanks, 1990) and Lexicographer's Mutual Information (LMI): $LMI(term, feature) = f(term, feature)\log_2(\frac{f(term,feature)}{f(term)f(feature)})$ (Evert, 2005).

We then prune all negatively correlated pairs ($s<0$). The maximum number of context features per term are defined with p, as we argue that it is sufficient to keep only the p most salient (ordered descending by their significance score) context features per term. Features of low saliency generally should not contribute much to the similarity of terms and also could lead to spurious similarity scores. Afterwards, all terms are aggregated by their features, which allows us to compute similarity scores between all terms that share at least one such feature.

Whereas the method introduced by (Pantel and Lin, 2002) is very similar to the one proposed in this paper (the similarity between terms is calculated solely by the number of features two terms share), they use PMI to rank features and do not use pruning to scale to large corpora, as they use a rather small corpus. Additionally, they do not evaluate the effect of such pruning.

In contrast to the best measures proposed by Lin (1998; Curran (2002; Pantel et al. (2009; Goyal et al. (2010) we do not calculate any information measure using frequencies of features and terms (we use significance ranking instead), as shown in Table 1.

Additionally, we avoid any similarity measurement using the information measure, as also done in these approaches, to calculate the similarity over the feature counts of each term: we merely count how many salient features two terms share. All these constraints makes this approach more scalable to larger corpora, as we do not need to know the full list of

Information Measures	
Lin's formula	$I(term, feature) = lin(term, feature) = \log \frac{f(term,feature)*f(relation(feature))}{\sum(f(word,relation(feature))f(word)}$
Curran's TTest	$I(term, feature) = ttest(term, feature) = \frac{p(term,feature)-p(feature)*p(term)}{\sqrt{p(feature)*p(term)}}$
Similarity Measures	
Lin's formula	$sim(t_1, t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)}(I(t_1,f)+I(t_2,f))}{\sum_{f \in features(t_1)} I(t_1,f)+\sum_{f \in features(w_2)} I(w_2,f)}$
Curran's Dice	$sim(t_1, t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)} min(I(t_1,f),I(t_2,f))}{\sum_{f \in features(t_1) \cap features(t_2)}(I(t_1,f)+I(t_2,f))}$ with $I(t, f) > 0$
Our Measure	$sim(t_1, t_2) = \sum_{f \in features(t_1) \cap features(t_2)} 1$ with $s > 0$

Table 1: Similarity measures used for computing the distributional similarity between terms.

features for a term pair at any time. While our computations might seem simplistic, we demonstrate its adequacy for large corpora in Section 5.

4 Evaluation

The evaluation is performed using a recent dump of English Wikipedia, containing 36 million sentences and a newspaper corpus, compiled from 120 million sentences (about 2 Gigawords) from Leipzig Corpora Collection (Richter et al., 2006) and the Gigaword corpus (Parker et al., 2011). The DTs are based on collapsed dependencies from the Stanford Parser (Marneffe et al., 2006) in the holing operation. For all DTs we use the pruning parameters s=0, p=1000 and w=1000. In a final evaluation, we use the syntactic n-grams built from Google Books (Goldberg and Orwant, 2013).

To show the impact of corpus size, we downsampled our corpora to 10 million, 1 million and 100,000 sentences. We compare our results against DTs calculated using Lin's (Lin, 1998) measure and the best measure proposed by Curran (2002) (see Table 1).

Our evaluation is performed using the same 1000 frequent and 1000 infrequent nouns as previously employed by Weeds et al. (2004). We create a gold standard, by extracting reasonable entries of these 2000 nouns using Roget's 1911 thesaurus, Moby Thesaurus, Merriam Webster's Thesaurus, the Big Huge Thesaurus and the OpenOffice Thesaurus and employ the inverse ranking measure (Curran, 2002) to evaluate the DTs.

Furthermore, we introduce a WordNet-based method. To calculate the similarity between two terms, we use the WordNet::Similarity path (Pedersen et al., 2004) measure. While its absolute scores are hard to interpret due to inhomogenity in the granularity of WordNet, they are well-suited for relative comparison. The score between two terms is inversely proportional to the shortest path between all the synsets of both terms. The highest possible score is one, if two terms share a synset. We compare the average score of the top five (or ten) entries in the DT for each of the 2000 selected words for our comparison.

5 Results

First, we inspect the results of Curran's measure using the Wikipedia and newspaper corpus for the frequent nouns, shown in Figure 1.

Both graphs show the inverse ranking score against the size of the corpus. Our method scores consistently higher when using LMI instead of PMI for ranking the features per term. The PMI measure declines when the corpus becomes larger. This can be attributed to the fact that PMI favors term-context pairs involving rare contexts (Bordag, 2008). Computing similarities between terms should not be performed on the basis of rare contexts, as these do not generalize well because of their sparseness.

All other measures improve with larger corpora. It is surprising that recent works use PMI to calculate similarities between terms (Goyal et al., 2010; Pantel et al., 2009), who, however evaluate their approach only with respect to their own implementation or extrinsically, and do not prune on saliency. Apart from the PMI measure, Curran's measure leads to the weakest results. We could not confirm that his measure outperforms Lin's measure as stated in (Curran, 2002)[1]. An explanation for this results

[1]Regarding Curran's Dice formula, it is not clear whether to use the intersection or the union of the features. We use an intersection, as it is unclear how to interpret the minimum function otherwise, and the alternatives performed worse.

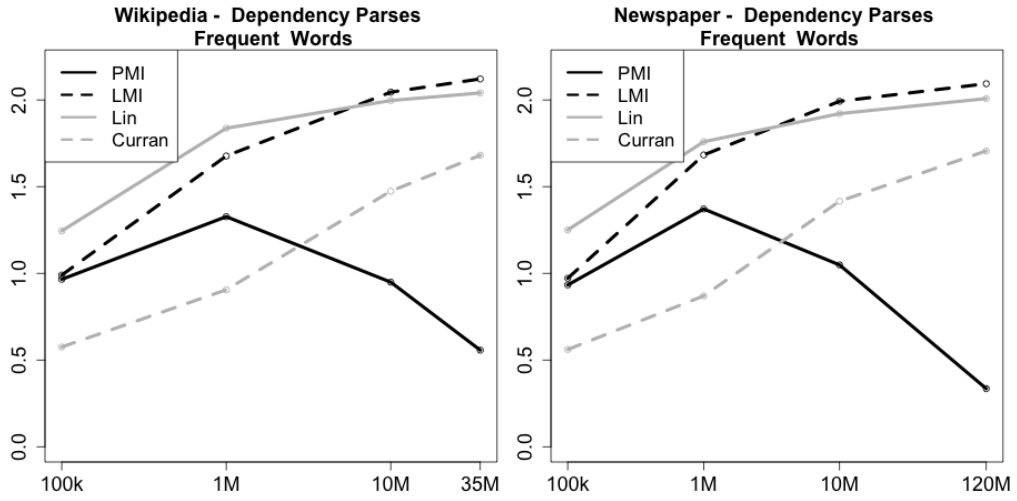

Figure 1: Inverse ranking for 1000 frequent nouns (Wikipedia left, Newspaper right) for different sized corpora. The 4 lines represent the scores of following DTs: our method using LMI (dashed black line) and the PMI significance measure (solid black line) and Curran's (dash bray line) and Lin's measure (solid tray line).

might be the use of a different parser, very few test words and also a different gold standard thesaurus in his evaluation. Comparing our method using LMI to Lin's method, we achieve lower scores with our method using small corpora, but surpass Lin's measure from 10 million sentences onwards.

Next, we show the results of the WordNet evaluation measure in Figure 2. Comparing the top 10 (upper) to the top 5 words (lower) used for the evaluation, we can observe higher scores for the top 5 words, which validates the ranking. These results are highly correlated to the results achieved with the inverse ranking measure. This is a positive result, as the WordNet measure can be performed automatically using a single public resource[2]. In Figure 3, we show results for the 1000 infrequent nouns using the inverse ranking (upper) and the WordNet measure (lower).

We can see that our method using PMI does not decline for larger corpora, as the limit on first-order features is not reached and frequent features are still being used. Comparing our LMI DT is en par with Lin's measure for 10 million sentences, and makes better use of large data when using the complete dataset. Again, the inverse ranking and the WordNet Path measure are highly correlated.

[2]Building a gold standard thesaurus following Curran (2002) needs access to all the used thesauri. Whereas for some, programming interfaces exist, often with limited access and licence restrictions, others have to be extracted manually.

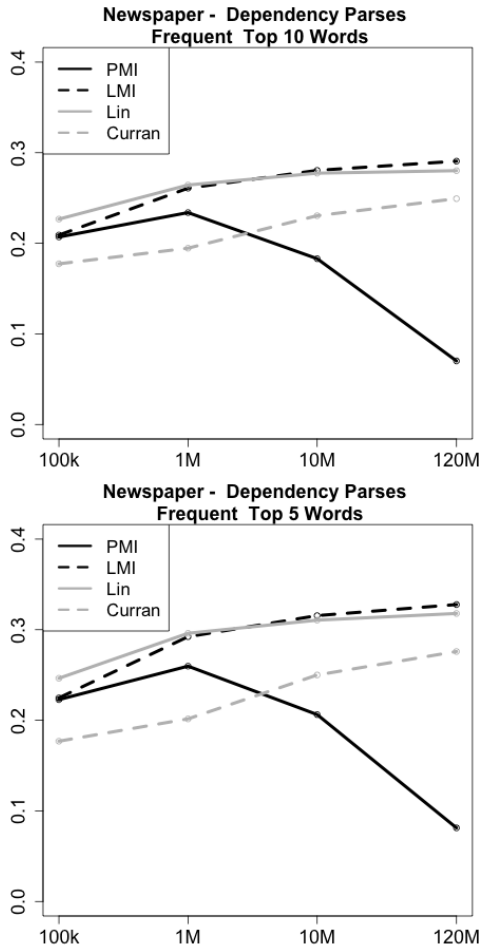

Figure 2: Results, using the WordNet:Path measure for frequent nouns using the newspaper corpus.

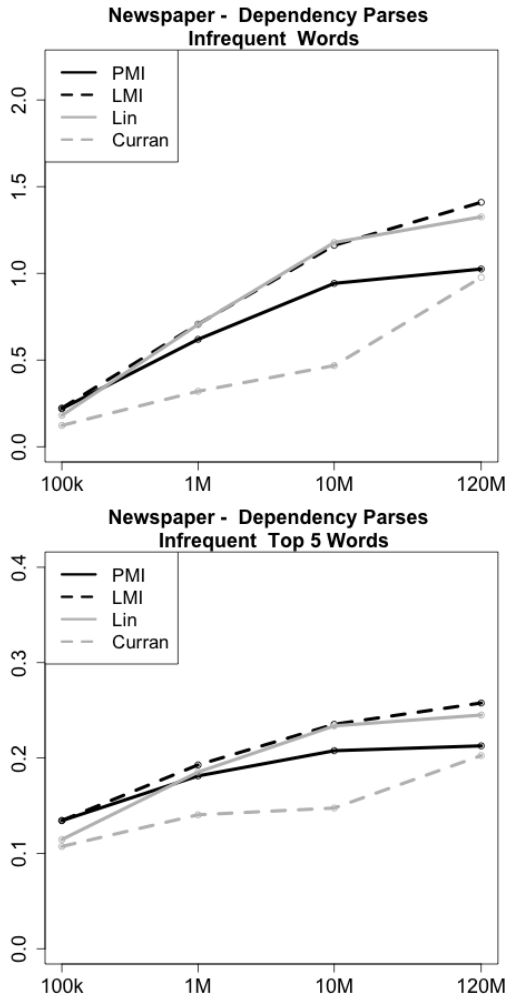

Figure 3: WordNet::Path results for 1000 infrequent nouns

The results shown here validate our pruning approach. Whereas Lin and Curran propose approaches to filter features that have low word feature scores, they do not remove features that occur with too many words, which is done in this work. Using these pruning steps, a simplistic similarity measure does not only lead to reduced computation times, but also to better results, when using larger corpora.

5.1 Using a large[3] corpus

We demonstrate the scalability of our method using the very large Google Books dataset (Goldberg and Orwant, 2013), consisting of dependencies extracted from 17.6 billion sentences. The evaluation results, using different measures, are given in Table 2.

Comparing the results for the Google Books DT to the ones achieved using Wikipedia and the newspaper, we can observe a boost in the performance, both for the inverse ranking and the WordNet measures. Additionally, we show results for the P@1 measure, which indicates the percentage of entries, whose first entry is in the gold standard thesaurus. Remarkably, we get a P@1 against our gold standard thesaurus of 76% for frequent and 64% for infrequent nouns using the Google Books DT.

	Corpus	Inv.	P@1	Path@5	Path@10
frequent nouns	Newspaper	2.0935	0.709	0.3277	0.2906
	Wikipedia	2.1213	0.703	0.3365	0.2968
	Google Books	2.3171	0.764	0.3712	0.3217
infrequent nouns	Newspaper	1.4097	0.516	0.2577	0.2269
	Wikipedia	1.3832	0.514	0.2565	0.2265
	Google Books	1.8125	0.641	0.2989	0.2565

Table 2: Comparing results for different corpora.

The most computation time was needed for the dependency parsing and took two weeks on a small cluster (64 cores on 8 nodes) for the 120 million Newspaper sentences. The DT for the Google Books was calculated in under 30 hours on a Hadoop cluster (192 cores on 16 nodes) and could be calculated within 10 hours for the Newspaper corpus. The computation of a DT using this huge corpus would be intractable with standard vector-based measurements. Even computing Lin's and Curran's vector-based similarity measure for the whole vocabulary of the newspaper corpus was not possible with our Hadoop cluster, as too much memory would have been required and thus we computed similarities only for the 2000 test nouns on a server with 92GB of main memory.

6 Conclusion

We have introduced a highly scalable approach to DT computation and showed its adequacy for very large corpora. Evaluating against thesauri and WordNet, we demonstrated that our similarity measure yields better-quality DTs and scales to corpora of billions of sentences, even on comparably small compute clusters. We achieve this by a number of pruning operations, and distributed processing. The framework and the DTs for Google Books, Newspaper and Wikipedia are available online[3] under the ASL 2.0 licence.

[3] https://sf.net/projects/jobimtext/

Acknowledgments

This work has been supported by the Hessian research excellence program "Landes-Offensive zur Entwicklung Wissenschaftlich-konomischer Exzellenz" (LOEWE) as part of the research center "Digital Humanities". We would also thank the anonymous reviewers for their comments, which greatly helped to improve the paper.

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*, NAACL '09, pages 19–27, Boulder, Colorado, USA.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 26–33, Toulouse, France.

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

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Stefan Bordag. 2008. A comparison of co-occurrence and similarity measures as simulations of context. In *CICLing'08 Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, pages 52–63, Haifa, Israel.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

James R. Curran. 2002. Ensemble methods for automatic thesaurus extraction. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 222–229, Philadelphia, PA, USA.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of Operating Systems, Desing & Implementation (OSDI) '04*, pages 137–150, San Francisco, CA, USA.

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

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247, Atlanta, Georgia, USA.

Gene H. Golub and William M. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *J. Soc. Indust. Appl. Math.: Ser. B, Numer. Anal.*, 2:205–224.

James Gorman and James 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-44, pages 361–368, Sydney, Australia.

Amit Goyal and Hal Daumé, III. 2011. Generating semantic orientation lexicon using large data and thesaurus. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11, pages 37–43, Portland, Oregon, USA.

Amit Goyal, Jagadeesh Jagarlamudi, Hal Daumé, III, and Suresh Venkatasubramanian. 2010. Sketch techniques for scaling distributional similarity to the web. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '10, pages 51–56, Uppsala, Sweden.

Dekang Lin. 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 '98, pages 64–71, Madrid, Spain.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 768–774, Montreal, Quebec, Canada.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28(2):203–208.

Marie-Catherine De Marneffe, Bill Maccartney, and Christopher 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.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 613–619, Edmonton, Alberta, Canada.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 938–947, Singapore.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia, PA, USA.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Boston, Massachusetts, USA.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the leipzig corpora collection. In *Proceedings of the IS-LTC 2006*, Ljubljana, Slovenia.

Pavel Rychlý and Adam Kilgarriff. 2007. An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 41–44, Prague, Czech Republic.

Magnus Sahlgren. 2006. *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* Ph.D. thesis, Stockholm University.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, pages 1015–1021, Geneva, Switzerland.