Large-scale evaluation of dependency-based DSMs: Are they worth the effort?

Gabriella Lapesa
Institute for Natural Language Processing
University of Stuttgart
gabriella.lapesa@ims.uni-stuttgart.de

Stefan Evert
Corpus Linguistics Group
FAU Erlangen-Nürnberg
stefan.evert@fau.de

Abstract

This paper presents a large-scale evaluation study of dependency-based distributional semantic models. We evaluate dependency-filtered and dependency-structured DSMs in a number of standard semantic similarity tasks, systematically exploring their parameter space in order to give them a "fair shot" against window-based models. Our results show that properly tuned window-based DSMs still outperform the dependency-based models in most tasks. There appears to be little need for the language-dependent resources and computational cost associated with syntactic analysis.[1]

1 Introduction

Distributional semantic models (DSMs) based on syntactic dependency relations (Padó and Lapata, 2007; Baroni and Lenci, 2010) represent a more linguistically informed version of the widely-used window-based DSMs (Sahlgren, 2006; Bullinaria and Levy, 2007; Bullinaria and Levy, 2012). Both types of DSMs operationalize the meaning of a target word t as a set of co-occurrence patterns extracted from language corpora. While window-based DSMs adopt a surface-oriented perspective (two words co-occur if they appear within a certain span, e.g. of 4 tokens), dependency-based DSMs adopt a *syntactic* perspective on co-occurrence: "nearness" is defined by the presence of a syntactic relation between target and features (e.g. direct object, subject, adjectival modifier), which may also correspond to a path along several edges of a dependency graph. If syntactic relations are only used to determine co-occurrence contexts, we talk of *dependency-filtered* DSMs; if the type of relation is explicitly encoded in the context features (e.g. "subj_dog"), we talk of *dependency-typed* DSMs.

The fortune of syntax-based models in distributional semantics has been mixed. Early work on dependency-filtered (Padó and Lapata, 2007) or dependency-typed (Rothenhäusler and Schütze, 2009; Baroni and Lenci, 2010) DSMs indicated that syntax-based semantic representations are indeed superior. These evaluation studies, however, were restricted to a specific corpus (BNC in Padó and Lapata (2007)) or task (noun clustering in Rothenhäusler and Schütze (2009)), or based on a very specific notion of co-occurrence (Baroni and Lenci, 2010)[2]. Meanwhile, extensive evaluation studies and parameter tuning led to significant improvements in the performance of window-based models (Bullinaria and Levy, 2007; Bullinaria and Levy, 2012; Lapesa and Evert, 2014) to the point that dependency-based DSMs currently hold the state-of-the-art only in very few standard semantic similarity tasks; see Baroni et al. (2014) and Lapesa and Evert (2014) for an overview of the state of the art. Among recent comparative evaluation studies, only Kiela and Clark (2014) attempt a direct comparison between the parameter spaces of window-based and syntax-based DSMs: once again, window-based models are found to perform better (with the exception of models built from the large Google Books N-gram corpus), but the scope of this comparison is rather limited.

The aim of this paper is to establish a fair ground for the comparison between window-based and dependency-based DSMs. To that end, we take as a reference point the large parameter set evaluated by

[1] The analysis presented in this paper is complemented by supplementary materials, which are available for download at http://www.linguistik.fau.de/dsmeval/.

[2] Among the dependency-based DSMs evaluated by Baroni and Lenci (2010), the best performing one relies on type-based co-occurrence: the co-occurrence strength between a target and a context is quantified as the number of different patterns in which they occur.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 394–400,
Valencia, Spain, April 3-7, 2017.

Lapesa and Evert (2014) and Lapesa et al. (2014) for window-based models. We carry out a parallel evaluation for dependency-based DSMs using the same tasks, datasets, parameters – adding some parameters specific to syntax-based models (such as the parser used and the type of allowed dependency relations) – and model selection methodology, allowing for a direct comparison of the results.

We address the question of whether dependency-based models can significantly improve DSM performance if the parameters are properly set, and whether the degree of the improvement justifies the increased complexity of the extraction process. In either case, a more thorough understanding of the parameter space will be beneficial for applications that prefer dependency-based DSMs on general grounds, e.g. because of an integration with syntactic structure (Erk et al., 2010). While the evaluation reported here does not encompass predict-type models, we believe that our findings also apply to the usefulness of dependency information in neural word embeddings (Levy and Goldberg, 2014).

2 Evaluation setting

Tasks & Datasets Our evaluation covers all tasks and datasets used by Lapesa and Evert (2014) and Lapesa et al. (2014). For space reasons, we present detailed results for one representative dataset from each task[3]: the **TOEFL synonym test** dataset (Landauer and Dumais, 1997) for the multiple-choice synonymy task (performance: accuracy); the **Generalized Event Knowledge** (McRae and Matzuki, 2009) dataset (GEK), a collection of 402 triples (target, consistent prime, inconsistent prime), for the multiple-choice semantic priming task (performance: accuracy)[4]; the **WordSim-353** (WS353) dataset, which contains 353 noun pairs with similarity/relatedness ratings (Finkelstein et al., 2002) for the task of predicting human similarity ratings (performance: Pearson's *r*); and the **Almuhareb-Poesio** (AP) dataset, containing 402 nouns grouped into 21 semantic classes (Almuhareb, 2006) for the noun clustering task (performance: cluster purity[5]).

DSM parameters We employ a large vocabulary of target words (27,522 lemma types), based on the vocabulary of Distributional Memory (Baroni and Lenci, 2010) and extended to cover all items in our datasets. After extracting dependency paths from the source corpora, the DSMs were compiled using the UCS toolkit[6] and the `wordspace` package for R (Evert, 2014). We evaluate the following parameters:

Source corpus (abbreviated in the plots as *corpus*): BNC[7], WaCkypedia_EN, and ukWaC[8];

Format of dependency relations (*dep.style*): Basic vs. collapsed with propagation of conjuncts (De Marneffe et al., 2006; De Marneffe and Manning, 2008);

Annotation pipeline (*parser*): TreeTagger (Schmid, 1995) and MALT parser (Nivre, 2003) vs. bidirectional POS tagger and Neural Network parser of Stanford CoreNLP (Chen and Manning, 2014);

Path length (*path.length*): we include paths with a maximum length of 1, 2, 3, 4 or 5 edges;

Type of dependency relations (*dep.type*): paths composed only of core dependencies (main actants of the sentence) vs. paths that also allow external dependencies (inter-clausal relations and conjuncts);

Threshold for context selection (*orig.dim*): we select the 5k, 10k, 20k, 50k, or 100k most frequent context dimensions;

Score for feature weighting (*score*): frequency, tf.idf, Dice coefficient, simple log-likelihood, Mutual Information (MI), t-score, or z-score;[9]

Feature transformation (*transformation*): an additional square root, sigmoid (`tanh`), or logarithmic transformation applied to feature scores vs. no transformation;

Number of latent SVD dimensions (*red.dim*): we project vectors into 1000 dimensions using randomized SVD (Halko et al., 2009), then select the first 100, 300, 500, 700, or 900 latent dimensions;

Number of skipped SVD dimensions (*dim.skip*): exclude the first 0, 50 or 100 latent

[3] If more than one dataset was available for a task, we preferred larger datasets (for which results are more reliable). Results for all datasets will be made available in the supplementary materials.

[4] In contrast to the paradigmatic relation targeted by TOEFL (i.e., synonymy), the GEK dataset focuses on relatedness of a more syntagmatic nature. See Lapesa et al. (2014) for more details on this dataset.

[5] Based on k-medoids clustering (Kaufman and Rousseeuw, 1990, Ch. 2) with standard parameter settings.

[6] http://www.collocations.de/software.html

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

[8] Both ukWaC and WaCkypedia_EN are available from http://wacky.sslmit.unibo.it/doku.php?id=corpora.

[9] All methods use sparse non-negative variants; e.g. our MI corresponds to positive pointwise MI (PPMI).

dimensions (e.g., those with the highest singular values); previous work on window-based DSMs (Bullinaria and Levy, 2012; Lapesa and Evert, 2014; Lapesa et al., 2014) showed that model performance improves when the initial components of the reduced matrix (i.e., those with the highest variance) are discarded.

Distance metric (*metric*): cosine distance (i.e. the angle between vectors) vs. Manhattan distance;

Index of distributional relatedness (*rel.index*): the semantic relatedness of words *a* and *b* in a DSM is quantified either by their metric distance $d(a, b)$ or by neighbor rank (rank of *b* among the neighbors of *a* for TOEFL and GEK, mean of $\log \mathrm{rank}(a, b)$ and $\log \mathrm{rank}(b, a)$ for WS353 and AP).

Among the evaluated parameters, *parser*, *dep.type* and *dep.style* are specific to dependency-based DSMs. *Path.length* is the dependency-based equivalent of window size in a bag-of-words DSM. The comparison between *filtered vs. typed* DSMs can be considered roughly equivalent to the comparison between undirected and directed windows in a bag-of-words DSM. All the other parameters are shared with window-based DSMs.

Evaluation methodology We tested all possible combinations of the parameters described above, resulting in a total of 806400 runs per model class (filtered vs. typed), which were generated and evaluated on a large HPC cluster within approximately 6 weeks. To meaningfully interpret the evaluation results, we apply a model selection methodology that is sensitive to parameter interactions and robust to overfitting. Following Lapesa and Evert (2013), we analyze the influence of individual parameters and their interactions using general linear models with performance (accuracy, correlation, purity) as a dependent variable and the model parameters as independent variables, including all two-way interactions. Analysis of variance – which is straightforward for our full factorial design – is used to quantify the impact of each parameter or interaction. Robust optimal parameter settings are identified with the help of effect displays (Fox, 2003), which show the partial effect of one or two parameters by marginalizing over all other parameters. Unlike coefficient estimates, they allow an intuitive interpretation of the effect sizes of categorical variables irrespective of the dummy coding scheme used.

3 Results

As model runs without dimensionality reduction performed consistently worse than the corresponding SVD-reduced runs, we only report results for the latter in this paper.

Impact of parameters We use a feature ablation approach to assess which parameters have the strongest impact on model performance. The ablation value of a parameter is the proportion of variance accounted for by the parameter together with all its interactions (corresponding to the reduction in adjusted R^2 of the model fit if the parameter were left out). Figures 1 and 2 visualize the feature ablation values of all evaluated parameters in the dependency-filtered and dependency-typed setting, respectively. Table 1 shows R^2 for the full model as well as all major interactions (partial $R^2 > 1\%$).

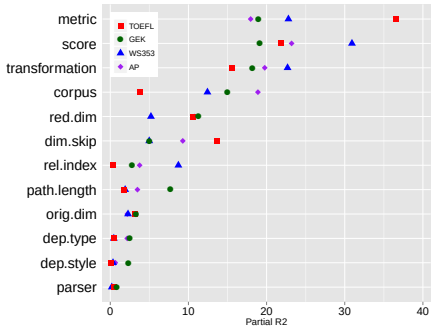

Figure 1: Feature ablation (dependency-filtered)

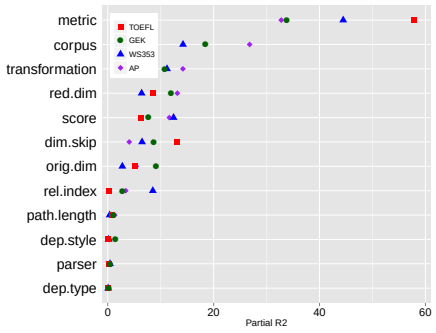

Figure 2: Feature ablation (dependency-typed)

	Filtered				Typed			
	T	G	W	A	T	G	W	A
Full model	88	83	88	83	89	84	90	88
score × transf	8.3	7.8	11.2	8.6	2.4	3.5	5.0	5.7
score × metric	1.3	1.5	1.5	1.8	–	–	–	–
corpus × metric	–	–	–	–	–	–	1.0	4.6
metric × red.dim	–	2.5	1.4	–	–	2.0	1.3	4.7
metric × dim.skip	4.0	1.0	1.1	3.4	4.9	1.6	2.2	1.2
metric × orig.dim	1.0	2.0	1.2	–	3.3	6.6	2.0	2.3

Table 1: R^2 of full model and major interactions for T[OEFL], G[EK], W[S353] and A[P] datasets

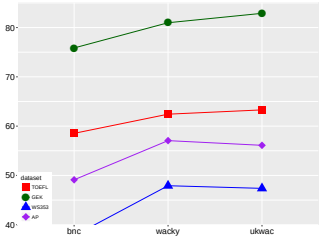

Figure 3: Corpus (filt)

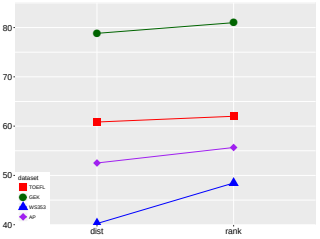

Figure 4: Rel. index (filt)

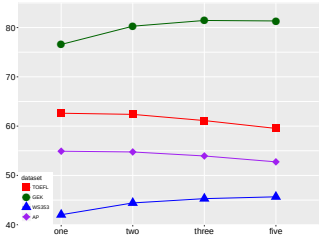

Figure 5: Path length (filt)

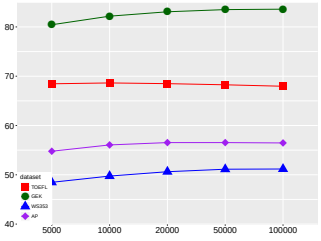

Figure 6: Context dim. (filt)

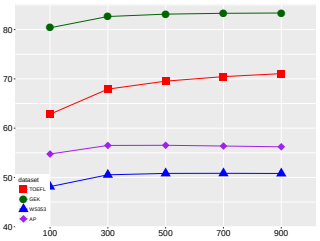

Figure 7: Red. SVD dim. (filt)

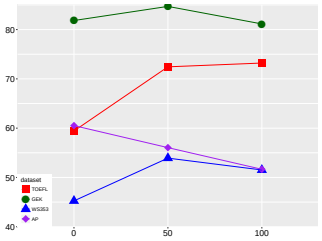

Figure 8: Skip SVD dim. (filt)

Parameters can be divided into three groups. First, a group of parameters with a **strong** impact on model performance, which is dominated by *metric* in both settings. *Metric* also has strong interactions with many other parameters. Further parameters in this group are *score* and *transformation*, again with a strong interaction across all datasets and both settings (Lapesa and Evert (2014) found this interaction to be the strongest also for window-based DSMs), as well as *corpus*. Second, a group of parameters with an **intermediate** impact includes the two SVD-related parameters (*red.dim* and *dim.skip*) and, to a lesser extent, the number of context dimensions (*orig.dim*) and the relatedness index (*rel.index*). *Path.length* only affects dependency-filtered models on the GEK dataset (that directly involves syntagmatic relatedness) and, but to a lesser extent, on AP (which encodes cohyponymy). It is almost irrelevant in a dependency-typed setting. This is probably due to the fact that direct dependency relations already capture the "core" of the semantic space and the information contributed by longer paths is neutralized by the additional noise. Third, a group of **irrelevant** parameters, which comprises the details of the dependency scheme (*dep.style* and *dep.type*) as well as the *parser* used.

Best parameter values In this section, we identify the best parameter settings by inspecting partial effect plots. We focus on dependency-filtered models because they consistently achieve better results and only discuss the dependency-typed ones when the best parameters are different. As for window-based DSMs, the Manhattan *metric* always performs much worse than cosine distance; the different behaviour of the two metrics also accounts for most of the interactions listed in table 1. We therefore exclude runs with Manhattan metric from further analysis and the effect plots below. The two bigger *corpora* are always a better choice (figure 3), with a preference for ukWaC in the multiple choice tasks. *Neighbor rank* (figure 4) outperforms distance, but the increased computational cost may only be justified for AP and WS353; the effect is much stronger for unreduced models in all tasks. As far as *path length* (figure 5) is concerned, datasets containing syntagmatic (GEK) or non-attributional relatedness (WS353) need longer paths to reach optimal performance. While the TOEFL task only requires 5k *context dimensions* (figure 6), more dimensions are necessary for AP and WS353 (20k and 50k) and even more for GEK (100k). Performance in all tasks improves with an increasing number of *reduced dimensions*, but 300 appear to be sufficient for AP and WS353 (figure 7); *skipping* the first 50 latent dimensions is beneficial for all tasks except AP (figure 8). The strong interaction between *score* and *transformation*, displayed in figure 12 for AP dataset and in figure 13 for GEK, indicates a preference for simple log-likelihood with log transformation or MI without any transformation (similar tendencies to AP hold for the remaining datasets). Parameters which are not explanatory can be set to the most "economic" value: MALT for *parser*, basic for *dependency style*, and core for *dependency type*.

Let us now briefly turn to dependency-typed

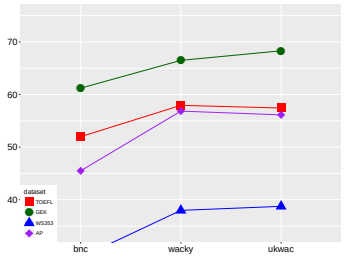

Figure 9: Corpus (typed)

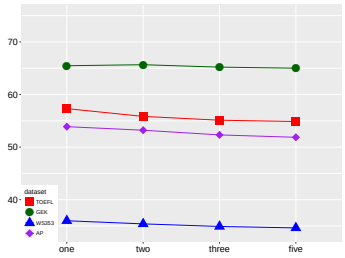

Figure 10: Path length (typed)

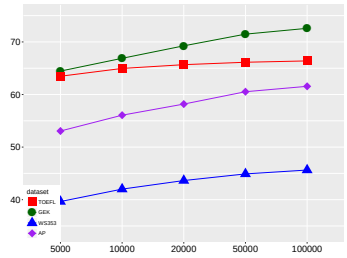

Figure 11: Context dim. (typed)

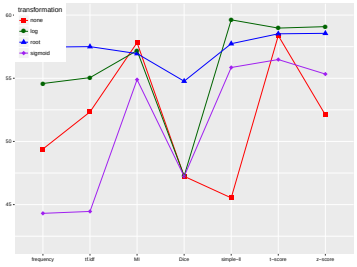

Figure 12: AP: Score × Transformation (filt)

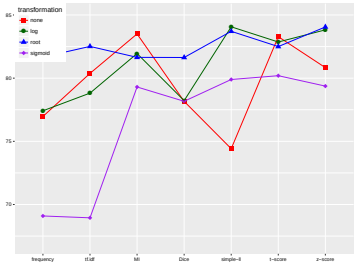

Figure 13: GEK: Score × Transformation (filt)

models. Preference for *corpus* remains on bigger corpora (figure 9). Figure 10 reveals that longer paths are detrimental (only exception being GEK's minor improvement with paths of length two). Figure 11 shows that the highest number of *context dimensions* (100k) is necessary for all tasks.

	corpus	path	o.dim	r.dim	d.sk	b.set	b.bow	soa
Dependency filtered								
TOEFL	ukwac	1	5k	900	100	85	92.5	100
GEK	ukwac	3	100k	700	50	92.6	97.0	–
WS	wacky	5	50k	300	50	0.67	0.68	0.81
AP	wacky	1	20k	300	0	69.6	69.0	79.0
Dependency typed								
	corpus	path	o.dim	r.dim	d.sk	b.set	b.bow	soa
TOEFL	wacky	1	100k	900	100	81.2	92.5	100
GEK	ukwac	2	100k	900	50	86.8	97.0	–
WS	ukwac	1	100k	700	50	0.62	0.68	0.81
AP	wacky	1	100k	300	0	71.9	69.0	79.0

Table 2: Best parameter settings for each task, compared with window-based DSM and state-of-the-art

Best settings Table 2 reports the robustly optimal parameter settings for dependency-filtered and dependency-based models[10] and their perfomance (*b.set*). For comparison, we also show the performance of the optimized window-based DSM from Lapesa and Evert (2014) or Lapesa et al. (2014) (*b.bow*), and the state of the art for the task (*soa*). Table 3 reports the parameter values of general settings for the dependency filtered (*Filter*) and typed (*Typed*) models and their performance on the four datasets.

	corpus	path	o.dim	r.dim	d.sk	T	G	W	A
Filter	ukwac	2	50k	700	50	86.2	90.1	0.67	65.4
Typed	ukwac	1	100k	900	50	77.5	82.1	0.62	69.4

Table 3: General best settings (filtered and typed)

4 Conclusion

We presented the results of a large-scale evaluation study of syntax-based DSMs. We show that, even after extensive parameter tuning, syntax-based DSMs outperform comparable window-based models only in one task out of four (noun clustering). We found many similarities to window-based DSMs: a significant core of the parameter space (metric, score, transformation, relatedness index) is common to both types of models, in terms of their impact on performance as well as the best parameter values; path length trades off between paradigmatic similarity and non-attributional relatedness, in the same way window-size does; most tasks require more SVD dimensions than are commonly used, and synonymy is better modeled by discarding the first SVD dimensions. It is left for future work to establish to what extent our conclusions generalize to different languages[11] and to more linguistically challenging tasks (e.g., prediction of thematic fit ratings).

[10] Common parameters: parser: MALT; dep.style: basic; dep.type: core; score: simple log-likelihood; transformation: log; metric: cosine; rel.index: rank.

[11] For example, DSM evaluation on German reveals a mixed picture: on the one hand, Bott and Schulte im Walde (2015) found no advantage for syntax-based models over bag-of-words ones in a quite linguistic task: the prediction of particle verb compositionality; on the other, Utt and Padó (2014) did find advantages in the use of syntactic information in the German counterparts of TOEFL and WS353.

Acknowledgements

We are grateful to the three anonymous reviewers, whose comments helped improve our paper. Gabriella Lapesa's research is funded by the DFG Collaborative Research Centre SFB 732.

References

Abdulrahman Almuhareb. 2006. *Attributes in Lexical Acquisition*. Ph.D. thesis, University of Essex.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):1–49.

Marco Baroni, Georgiana Dinu, and Germán 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)*, pages 238–247.

Stefan Bott and Sabine Schulte im Walde. 2015. Exploiting fine-grained syntactic transfer features to predict the compositionality of german particle verbs. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 34–39.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming and SVD. *Behavior Research Methods*, 44:890–907.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP 2014*.

Marie-Catherine De Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual (revised for the Stanford parser v3.5.1 in February 2015). Technical report, Stanford University.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*, pages 449–454.

Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.

Stefan Evert. 2014. Distributional semantics in R with the wordspace package. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 110–114, Dublin, Ireland.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

John Fox. 2003. Effect displays in R for generalised linear models. *Journal of Statistical Software*, 8(15):1–27.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. Technical Report 2009-05, ACM, California Institute of Technology.

Leonard Kaufman and Peter J. Rousseeuw. 1990. *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons.

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of EACL 2014, Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Gabriella Lapesa and Stefan Evert. 2013. Evaluating neighbor rank and distance measures as predictors of semantic priming. In *Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2013)*, pages 66–74, Sofia, Bulgaria.

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 (TACL)*, 2:531–545.

Gabriella Lapesa, Stefan Evert, and Sabine Schulte im Walde. 2014. Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 160–170, Dublin, Ireland.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 302–308.

Ken McRae and Kazunaga Matzuki. 2009. People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and Linguistics Compass*, 3(6):1417–1429.

J. Nivre. 2003. Efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, pages 149–160.

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

Klaus Rothenhäusler and Hinrich Schütze. 2009. Unsupervised classification with dependency based word spaces. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics*, pages 17–24, Athens, Greece.

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, University of Stockolm.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

Jason Utt and Sebastian Padó. 2014. Crosslingual and multilingual construction of syntax-based vector space models. *Transactions of the Association of Computational Linguistics*, 2:245–258.