CogALex-V Shared Task: Mach5
A traditional DSM approach to semantic relatedness

Stefan Evert
Corpus Linguistics Group
Friedrich-Alexander-Universität Erlangen-Nürnberg
Bismarckstr. 6, 91054 Erlangen, Germany
stefan.evert@fau.de

Abstract

This contribution provides a strong baseline result for the CogALex-V shared task using a traditional "count"-type DSM (placed in rank 2 out of 7 in subtask 1 and rank 3 out of 6 in subtask 2). Parameter tuning experiments reveal some surprising effects and suggest that the use of random word pairs as negative examples may be problematic, guiding the parameter optimization in an undesirable direction.

1 Introduction

It is generally assumed that traditional "count"-type distributional semantic models (DSM) are good at identifying attributionally similar words, but cannot distinguish between different semantic relations (e.g. synonyms, antonyms, hypernyms) and work poorly for other forms of semantic relatedness such as meronymy (Baroni and Lenci, 2011). Moreover, DSMs based on syntactic dependency relations are supposed to achieve better results than window-based models (Padó and Lapata, 2007). The goal of the present paper is to test how well traditional DSMs can be tuned to identify different types of semantic relations in the CogALex-V shared task. It can thus be seen as a strong baseline against which more specialized approaches can be compared. The system developed here is nicknamed マッハ号, or Mach5[1] in English.

According to the distributional hypothesis (Harris, 1954), semantically related words should have a smaller distance in a distributional space than unrelated words, especially if they are attributionally similar. This suggests a simple strategy for the identification of semantically related words in subtask 1: candidate pairs are predicted to be related if their distributional distance is below a specified threshold value θ. The choice of θ determines the trade-off between precision and recall as visualized in the left panel of Fig. 1, where the thin dotted line shows precision (P) and the thin dashed line shows recall (R) for different values of θ. The optimal threshold $\theta^* = 80.7°$ – indicated by a circle and a thin vertical line – is chosen to maximize F_1-score, the harmonic mean of precision and recall, which is also the main evaluation criterion in the CogALex-V task. In this example, the DSM achieves $P = 76.27\%$, $R = 74.38\%$ and $F_1 = 75.31\%$ on the training data.

DSM distances cannot be used in the same way to discriminate between semantic relations in subtask 2 because antonyms, synonyms, hypernyms, etc. will all be relatively close in semantic space and their distance distributions are similar (Baroni and Lenci, 2011; Santus et al., 2015). Therefore, Mach5 implements a simple machine learning approach for this subtask, as described in Sec. 3. Parameters of the underlying DSM are tuned based on the overall identification of semantically related words (Sec. 2).

2 The Mach5 DSM

The Mach5 distributional model is based on ENCOW 2014, a large English Web corpus (Schäfer and Bildhauer, 2012) with a size of approx. 9.5 billion tokens after sentence deduplication. A particular

[1] https://en.wikipedia.org/wiki/Mach_Five

Proceedings of the Workshop on Cognitive Aspects of the Lexicon,
pages 92–97, Osaka, Japan, December 11-17 2016.

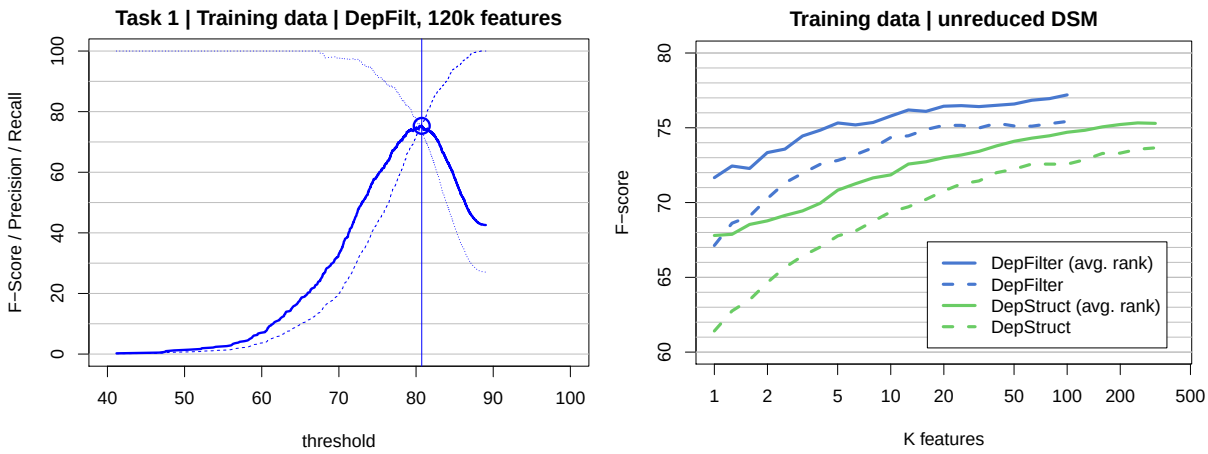

Figure 1: Left panel: Application of DSM distances (angles) to the identification of semantically related words. Right panel: Performance in subtask 1 depending on the number of feature dimensions used.

advantage of this huge corpus is its full coverage of the CogALex-V training and test sets, so that no special handling of unseen words is required. Both a dependency-filtered and a dependency-structured DSM were compiled from syntactic dependencies obtained with the robust C&C parser (Curran et al., 2007). The target vocabulary of 26,450 lemmas extends the vocabulary of Distributional Memory (Baroni and Lenci, 2010) with all words in the training and test sets of the shared task. Similar to the gold standard, the DSM uses lemmatized words (from TreeTagger) but does not distinguish between homonyms with different parts of speech (such as *clear*$_{\text{ADJ}}$ and *clear*$_{\text{VERB}}$). The 120,000 most frequent lemmas were extracted as features for the dependency-filtered model (henceforth DepFilt); the 300,000 most frequent relation-lemma combinations (e.g. OBJ=*cat*) were extracted as features for the dependency-structured model (henceforth DepStruct).

Some basic parameters were set according to the recommendations of Lapesa and Evert (2014): sparse (i.e. non-negative) simple log-likelihood (simple-ll) is used as an association measure for feature weighting and an additional log transformation is applied to the simple-ll scores. The models use angular distance (equivalent to cosine similarity) and explore logarithmic neighbour rank as an index of semantic (dis)similarity. Other parameters are tuned incrementally on the training data, as described in the following subsections. The main tuning criterion is the F_1 score achieved by an optimal cutoff threshold on the training data of subtask 1.

2.1 Feature selection

A first step is to determine how many feature dimensions are required in order to achieve good results and whether the dependency-filtered or the dependency-structured model is superior. The right panel of Fig. 1 plots F_1-scores in subtask 1 against the number of most frequent features and the other parameters. The graphs show clearly that more features produce better results and that further improvements may be expected from an even larger number of features, especially for DepStruct. Average logarithmic neighbour rank (solid lines) as an index of relatedness outperforms angular distance (dashed lines) by a large margin (forward and backward rank fall somewhere in between and have been omitted for clarity). DepFilt (blue, best $F_1 = 77.2\%$) is also considerably better than DepStruct (green, best $F_1 = 75.3\%$), even with a much smaller number of features.

Some authors suggest that medium-frequency features are the most informative for DSMs (Kiela and Clark, 2014), which motivates experiments with feature windows of 10,000–50,000 features in different frequency ranges. Fig. 2 shows the starting point of the feature window (i.e. the number of most frequent features skipped) on the x-axis and different window sizes in different colours.

For DepFilt (left panel), excluding the most frequent dimensions has a strong negative effect on angular distance. However, neighbour rank improves up to $F_1 = 78.03\%$ if the first 20,000–50,000 features are skipped, and it deteriorates much more slowly afterwards. The number of features in the window

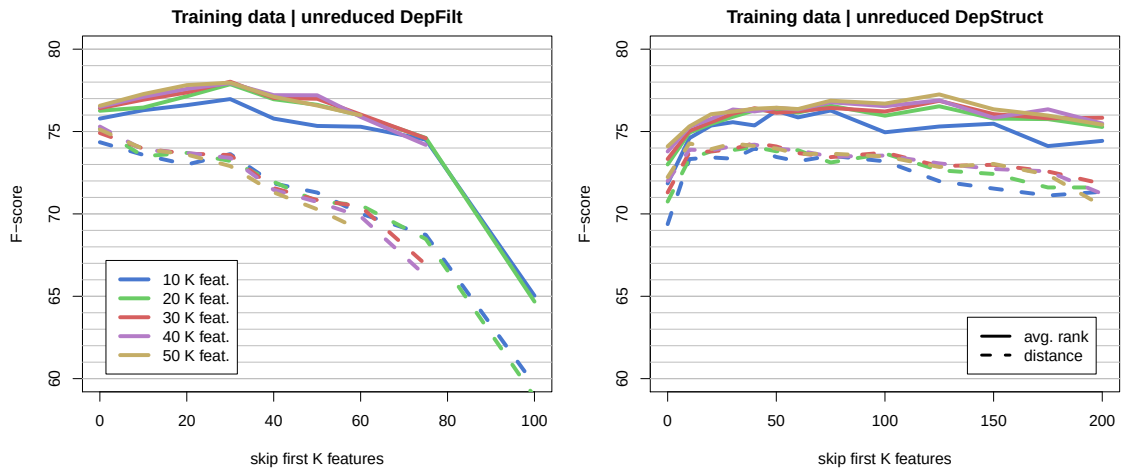

Figure 2: Performance of different feature windows in subtask 1 for DepFilt (left) and DepStruct (right). The x-axis shows how many thousands (K) of features are skipped.

seems to make little difference, especially for neighbour rank where as few as 20,000 features are sufficient.[2] The observations for DepStruct (right panel) are similar, but even more striking. Angular distance improves considerably if up to 50,000 high-frequency features are skipped, then declines only slowly. For neighbour rank, performance continues to improve and achieves $F_1 = 77.25\%$ when more than 100,000 features are skipped; in other words, a relatively small window of lower-frequency features seems to yield the best results. Contrary to what Fig. 1 suggested, DepFilt and DepStruct now achieve similar F_1-scores with a suitably chosen window of less than 50,000 words; both results are better than those reported above for the full feature sets.

For further experiments involving dimensionality reduction, somewhat larger feature windows are selected because the latent dimensions might be able to exploit shared information unlike the unreduced models evaluated here. DepFilt uses feature ranks 20,000–70,000 (with $1292 \leq f \leq 12720$) to achieve $F_1 = 77.25\%$ in subtask 1; DepStruct uses feature ranks 50,000–150,000 (with $1796 \leq f \leq 11040$) to achieve $F_1 = 76.52\%$.[3]

2.2 Dimensionality reduction by SVD

Most evaluation studies find that dimensionality reduction, which is traditionally carried out by an efficient sparse truncated singular value decomposition (SVD), improves DSM representations. Lapesa and Evert (2014) report consistently better results across a wide range of evaluation tasks and parameter settings. The following experiments explore how many latent dimensions are required and whether skipping the first latent dimensions is beneficial (Bullinaria and Levy, 2012; Lapesa and Evert, 2014). In addition, we look at a parameter that has only recently become popular: Caron's (2001) power scaling coefficient P for the SVD dimensions.[4] Bullinaria and Levy (2012) report a substantial improvement in model performance if P is set close to 0, especially for the TOEFL synonym task.

The DepFilt and DepStruct vectors selected in Sec. 2.1 are normalized according to the Euclidean norm, then SVD is applied to project them into 1000 latent dimensions for each model. The left panel of Fig. 3 shows that power scaling with $P < 1$ leads to a substantial improvement. For both models, $P = 0$ is a nearly optimal and theoretically motivated choice. It is particularly fascinating that power scaling evens out most of the differences between angular distance and neighbour rank as well as between DepStruct and DepFilt, with DepStruct performing slightly better now. Fixing $P = 0$, additional

[2] It is interesting to note that Schütze (1998) and several other early papers use 20,000 feature dimensions.

[3] In a 100-million-word corpus like the British National Corpus, this would correspond to frequencies between approx. 10 and 150 occurrences, i.e. a range of words that are normally excluded from distributional models.

[4] $P = 1$ corresponds to standard SVD, $P > 1$ gives more weight to the first latent dimensions (capturing the strongest correlation patterns), and $P < 1$ equalizes the dimensons. In particular, for $P = 0$ each latent dimension makes the same average contribution to distances between the word vectors (under certain additional circumstances).

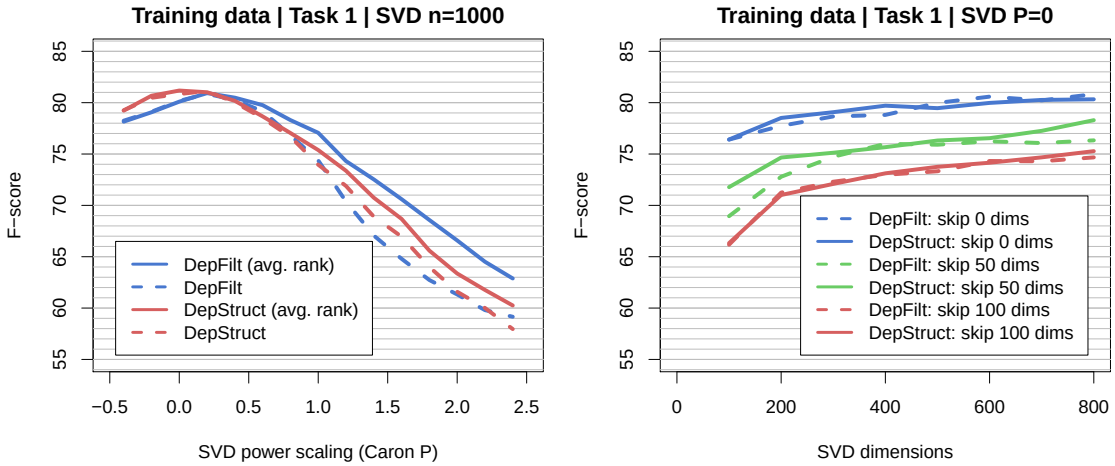

Figure 3: Effect of power scaling on SVD dimensions (Caron's P, left panel) and of further truncation of SVD as well skipping the first latent dimensions (for $P = 0$, right panel).

experiments show that skipping the first SVD dimensions does not lead to a further improvement, but rather to a considerably decrease in quality (left panel of Fig. 3). This surprising effect appears to be caused by feature equalization: without power scaling (i.e. $P = 1$), F_1 improves when skipping up to 50 SVD dimensions (not shown). Further truncation of the SVD to less than 600 dimensions also decreases quality, but performance seems to stabilize if at least 600 dimensions are used (blue lines in right panel).

The final Mach5 DSMs are based on the first 600 SVD dimensions with Caron's $P = 0$, equalizing the relative importance of the latent dimensions. Computationally cheaper distance values are used as an index of semantic relatedness, since they perform only marginally worse than average neighbour rank.

3 The Mach5 system

Run 1 of the Mach5 system only uses distance information from the DepFilt and DepStruct DSMs tuned in Sec. 2. For subtask 1, an optimal cutoff threshold is determined to maximize F_1 on the training data (DepFilt: $86.4°$, DepStruct: $87.0°$). For subtask 2, a SVM classifier with RBF kernel is applied to six-dimensional feature vectors containing angular distance as well as forward and backward neighbour rank from both DepFilt and DepStruct. Metaparameters (C and class weights) were tuned manually by cross-validation of weighted average F_1-scores on the training data.[5] Evaluation results on the training and test sets are shown in Table 1.

Run 2 explores the possibility that different types of semantic relations might be encoded in different SVD dimensions, which can be exploited by changing the weights of the dimensions when computing semantic distances. As a computationally efficient approximation, we apply a linear SVM classifier to feature vectors containing the contribution $(x_i - y_i)^2$ of each latent dimension i to the Euclidean distance between the pre-normalized vectors $\mathbf{x}$ and $\mathbf{y}$ of a word pair.[6] Features from DepFilt and DepStruct are concatenated for a total of 1,200 feature dimensions. Both subtasks can be approached in this way, training either a binary (substak 1) or a five-way (subtask 2) SVM classifier. Again, metaparameters were manually tuned on the training data. It turned out to be crucial to set the cost parameter to a low value $C \leq .01$ in order to ensure strong regularization and avoid overfitting.

For the official submission, the runs performing best on the training data were selected, shown in bold in Table 1. Competition results are thus $F_1 = 77.88\%$ in subtask 1 and $\bar{F}_1 = 29.59\%$ in subtask 2. All models were implemented in R using the `wordspace` package for distributional semantics (Evert, 2014)

[5] Cross-validation uses a round-robin scheme grouped by target word in order to avoid item-specific learning. Without this precaution, cross-validated performance on the training data might be highly optimistic in some cases. For example, a simple round-robin scheme yielded $\bar{F}_1 = 39.88\%$ for run 2 in subtask 2, while the more realistic grouped cross-validation yields $\bar{F}_1 = 32.37\%$. Differences are much smaller for the simpler models of run 1.

[6] These features gave slightly better performance than contribution $x_i y_i$ to cosine similarity.

Run		Subtask 1		Subtask 2	
		Train F_1	Test F_1	Train $\bar{F}_1$	Test $\bar{F}_1$
run 1	DepFilt	**80.59**	**77.88**	—	—
	DepStruct	79.98	76.80	—	—
	both	—	—	26.47	23.76
run 2	both	78.12	72.76	**32.37**	**29.59**
run 3	both	*80.88*	*78.93*	—	*31.97*

Table 1: Evaluation results of different Mach5 runs on the training data (10-fold cross-validation) and test data, using the official F_1-scores in subtask 1 and weighted average $\bar{F}_1$ across the four semantic relations in subtask 2. Runs selected for the competition are shown in bold font, the best results obtained in follow-up experiments are shown in italics.

and the LibSVM classifier from package `e1071`. Mach5 can be downloaded as an R script together with the original co-occurrence data from `http://www.collocations.de/data/#mach5`.

4 Discussion

The optimal cutoff angles determined for subtask 1 are surprisingly high – close to orthogonality – which suggests a possible problem with the use of random word pairs as negative examples in the gold standard (and many other DSM evaluation tasks that also use random word pairs as a control). As a consequence, the parameter tuning in Sec. 2 was guided towards recognizing random word pairs rather than clearly defined semantic relations. The distribution of DSM distances for different semantic relations in the left panel of Fig. 4 supports this interpretation: the distances between semantically related words spread over a wide range and can become very large (sometimes even above 90°), while most random word combinations are almost precisely orthogonal. For a DSM with conventional state-of-the-art parameter settings[7] (right panel of Fig. 4), the distribution shows a much larger spread of the random word pairs.

It seems plausible that this "conventional" DSM may contain useful information for the discrimination between different semantic relations, while the Mach5 DSM has been tuned to identify random word pairs as accurately as possible. Therefore, a combined approach was implemented after the competition as run 3 of the Mach5 system. It uses an SVM classifier with RBF kernel, based on the six-dimensional features vectors from run 1, for distinguishing between related and unrelated word pairs in subtask 1. For the discrimination between semantic relations, a linear SVM classifier is trained only on related word pairs, using partial Euclidean distances from the conventional DepFilt model and partial inner products from the conventional DepStruct model as features (similar to run 2). In subtask 2, the first (binary) classifier identifies RANDOM pairs, while the second (four-way) classifier selects a relation label for the remaining word pairs. As can be seen from the bottom row of Table 1, run 3 performs noticeably better than the submitted system,[8] although it would still have ranked in second and third place, respectively, in the competition. These results provide additional support for the hypothesis that the wide-spread use of random word pairs as negative examples poses the risk of misguiding DSM parameter tuning.

References

Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–712.

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*, pages 1–10, Edinburgh, UK.

[7] DepFilt with 50,000 features (except for the 150 most frequent lemmas), reduced to first 600 SVD dimensions without power scaling ($P = 1$); DepStruct with 100,000 features (except for 400 most frequent ones) and same SVD projection.

[8] $F_1 = 78.93\%$ vs. 77.88% in subtask 1 and $\bar{F}_1 = 31.97\%$ vs. 29.59% in subtask 2

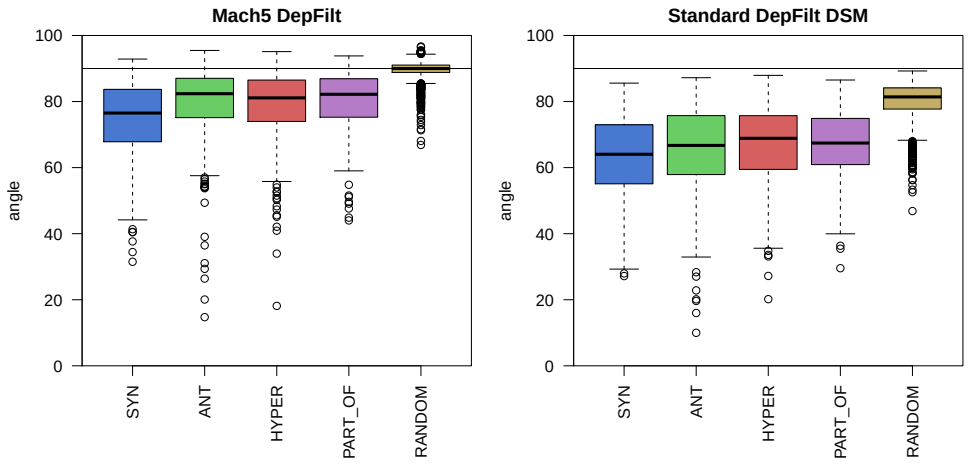

Figure 4: Distribution of DSM distances across the four semantic relations and the random controls in the test data for CogALex-V subtask 2, comparing the tuned Mach5 model (left panel) against a dependency-filtered DSM with conventional state-of-the-art parameter settings (right panel).

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(3):890–907.

John Caron. 2001. Experiments with LSA scoring: Optimal rank and basis. In Michael W. Berry, editor, *Computational Information Retrieval*, pages 157–169. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, pages 33–36, Prague, Czech Republic.

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, August.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162. Reprinted in Harris (1970, 775–794).

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden.

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.

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

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC '12)*, pages 486–493, Istanbul, Turkey. ELRA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.