Derivational Smoothing for Syntactic Distributional Semantics

Sebastian Padó* **Jan Šnajder†** **Britta Zeller***
*Heidelberg University, Institut für Computerlinguistik
69120 Heidelberg, Germany
†University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{pado, zeller}@cl.uni-heidelberg.de jan.snajder@fer.hr

Abstract

Syntax-based vector spaces are used widely in lexical semantics and are more versatile than word-based spaces (Baroni and Lenci, 2010). However, they are also sparse, with resulting reliability and coverage problems. We address this problem by *derivational smoothing*, which uses knowledge about derivationally related words (*oldish* → *old*) to improve semantic similarity estimates. We develop a set of derivational smoothing methods and evaluate them on two lexical semantics tasks in German. Even for models built from very large corpora, simple derivational smoothing can improve coverage considerably.

1 Introduction

Distributional semantics (Turney and Pantel, 2010) builds on the assumption that the semantic similarity of words is strongly correlated to the overlap between their linguistic contexts. This hypothesis can be used to construct context vectors for words directly from large text corpora in an unsupervised manner. Such vector spaces have been applied successfully to many problems in NLP (see Turney and Pantel (2010) or Erk (2012) for current overviews).

Most distributional models in computational lexical semantics are either (a) *bag-of-words* models, where the context features are words within a surface window around the target word, or (b) *syntactic* models, where context features are typically pairs of dependency relations and context words.

The advantage of syntactic models is that they incorporate a richer, structured notion of context. This makes them more versatile; the Distributional Memory framework by Baroni and Lenci (2010) is applicable to a wide range of tasks. It is also able – at least in principle – to capture more fine-grained types of semantic similarity such as predicate-argument plausibility (Erk et al., 2010). At the same time, syntactic spaces are much more prone to sparsity problems, as their contexts are sparser. This leads to reliability and coverage problems.

In this paper, we propose a novel strategy for combating sparsity in syntactic vector spaces, *derivational smoothing*. It follows the intuition that derivationally related words (*feed – feeder*, *blocked – blockage*) are, as a rule, semantically highly similar. Consequently, knowledge about derivationally related words can be used as a "back off" for sparse vectors in syntactic spaces. For example, the pair *oldish – ancient* should receive a high semantic similarity, but in practice, the vector for *oldish* will be very sparse, which makes this result uncertain. Knowing that *oldish* is derivationally related to *old* allows us to use the much less sparse vector for *old* as a proxy for *oldish*.

We present a set of general methods for smoothing vector similarity computations given a resource that groups words into derivational families (equivalence classes) and evaluate these methods on German for two distributional tasks (similarity prediction and synonym choice). We find that even for syntactic models built from very large corpora, a simple derivational resource that groups words on morphological grounds can improve the results.

2 Related Work

Smoothing techniques – either statistical, distributional, or knowledge-based – are widely applied in all areas of NLP. Many of the methods were first applied in Language Modeling to deal with unseen n-grams (Chen and Goodman, 1999; Dagan et al., 1999). Query expansion methods in Information Retrieval are also prominent cases of smoothing that addresses the lexical mismatch between query and document (Voorhees, 1994; Gonzalo et al., 1998; Navigli and Velardi, 2003). In lexical semantics, smoothing is often achieved by backing

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 731–735,
Sofia, Bulgaria, August 4-9 2013.

off from words to semantic classes, either adopted from a resource such as WordNet (Resnik, 1996) or induced from data (Pantel and Lin, 2002; Wang et al., 2005; Erk et al., 2010). Similarly, distributional features support generalization in Named Entity Recognition (Finkel et al., 2005).

Although distributional information is often used for smoothing, to our knowledge there is little work on smoothing distributional models themselves. We see two main precursor studies for our work. Bergsma et al. (2008) build models of selectional preferences that include morphological features such as capitalization and the presence of digits. However, their approach is task-specific and requires a (semi-)supervised setting. Allan and Kumaran (2003) make use of morphology by building language models for stemming-based equivalence classes. Our approach also uses morphological processing, albeit more precise than stemming.

3 A Resource for German Derivation

Derivational morphology describes the process of building new (derived) words from other (basis) words. Derived words can, but do not have to, share the part-of-speech (POS) with their basis (*old*$_A$ → *oldish*$_A$ vs. *warm*$_A$ → *warm*$_V$, *warmth*$_N$). Words can be grouped into *derivational families* by forming the transitive closure over individual derivation relations. The words in these families are typically semantically similar, although the exact degree depends on the type of relation and idiosyncratic factors (*book*$_N$ → *bookish*$_A$, Lieber (2009)).

For German, there are several resources with derivational information. We use version 1.3 of DERIVBASE (Zeller et al., 2013),[1] a freely available resource that groups over 280,000 verbs, nouns, and adjectives into more than 17,000 non-singleton derivational families. It has a precision of 84% and a recall of 71%. Its higher coverage compared to CELEX (Baayen et al., 1996) and IMSLEX (Fitschen, 2004) makes it particularly suitable for the use in smoothing, where the resource should include low-frequency lemmas.

The following example illustrates a family that covers three POSes as well as a word with a predominant metaphorical reading (*to kneel* → *to beg*):

> knieen$_V$ (*to kneel*$_V$), beknieen$_V$ (to beg$_V$), Kniende$_N$ (kneeling_person$_N$), kniend$_A$ (kneeling$_A$), Knie$_{Nn}$ (knee$_N$)

[1] Downloadable from: `http://goo.gl/7KG2U`

Using derivational knowledge for smoothing raises the question of how semantically similar the lemmas within a family really are. Fortunately, DERIVBASE provides information that can be used in this manner. It was constructed with hand-written derivation rules, employing string transformation functions that map basis lemmas onto derived lemmas. For example, a suffixation rule using the affix "heit" generates the derivation *dunkel – Dunkelheit* (*dark*$_A$ – *darkness*$_N$). Since derivational families are defined as transitive closures, each pair of words in a family is connected by a derivation path. Because the rules do not have a perfect precision, our confidence in pairs of words decreases the longer the derivation path between them. To reflect this, we assign each pair a *confidence* of $1/n$, where n is the length of the shortest path between the lemmas. For example, *bekleiden (enrobe*$_V$*)* is connected to *Verkleidung (disguise*$_N$*)* through three steps via the lemmas *kleiden (dress*$_V$*)* and *verkleiden (disguise*$_V$*)* and is assigned the confidence 1/3.

4 Models for Derivational Smoothing

Derivational smoothing exploits the fact that derivationally related words are also semantically related, by combining and/or comparing distributional representations of derivationally related words. The definition of a derivational smoothing algorithm consists of two parts: a *trigger* and a *scheme*.

Notation. Given a word w, we use $\mathbf{w}$ to denote its distributional vector and $\mathcal{D}(w)$ to denote the set of vectors for the derivational family of w. We assume that $\mathbf{w} \in \mathcal{D}(w)$. For words that have no derivations in DERIVBASE, $\mathcal{D}(w)$ is a singleton set, $\mathcal{D}(w) = \{\mathbf{w}\}$. Let $\alpha(w, w')$ denote the confidence of the pair (w, w'), as explained in Section 3.

Smoothing trigger. As discussed above, there is no guarantee for high semantic similarity within a derivational family. For this reason, smoothing may also drown out information. In this paper, we report on two triggers: *smooth always* always performs smoothing; *smooth if sim=0* smooths only when the unsmoothed similarity $sim(\mathbf{w}_1, \mathbf{w}_2)$ is zero or unknown (due to w_1 or w_2 not being in the model).

Smoothing scheme. We present three smoothing schemes, all of which apply to the level of complete families. The first two schemes are *exemplar-based* schemes, which define the smoothed similarity for a word pair as a function of the pairwise similarities between all words of the two derivational families.

The first one, *maxSim*, checks for particularly similar words in the families:

$$maxSim(w_1, w_2) = \max_{\substack{\mathbf{w}'_1 \in \mathcal{D}(w_1) \\ \mathbf{w}'_2 \in \mathcal{D}(w_2)}} sim(\mathbf{w}'_1, \mathbf{w}'_2)$$

The second one, *avgSim*, computes the average pairwise similarity (N is the number of pairs):

$$avgSim(w_1, w_2) = \frac{1}{N} \sum_{\substack{\mathbf{w}'_1 \in \mathcal{D}(w_1) \\ \mathbf{w}'_2 \in \mathcal{D}(w_2)}} sim(\mathbf{w}'_1, \mathbf{w}'_2)$$

The third scheme, *centSim*, is *prototype-based*. It computes a centroid vector for each derivational family, which can be thought of as a representation for the concept(s) that it expresses:

$$centSim(w_1, w_2) = sim\big(\mathbf{c}(\mathcal{D}(w_1)), \mathbf{c}(\mathcal{D}(w_2))\big)$$

where $\mathbf{c}(\mathcal{D}(w)) = \sum_{\mathbf{w}' \in \mathcal{D}(w)} \alpha(w, w') \cdot \mathbf{w}'$ is the confidence-weighted centroid vector. *centSim* is similar to *avgSim*. It is more efficient to calculate and effectively introduces a kind or regularization, where outliers in either family have less impact on the overall result.

These models only represents a sample of possible derivational smoothing methods. We performed a number of additional experiments (POS-restricted smoothing, word-based, and pair-based smoothing triggers), but they did not yield any improvements over the simpler models we present here.

5 Experimental Evaluation

Syntactic Distributional Model. The syntactic distributional model that we use represents target words by pairs of dependency relations and context words. More specifically, we use the $W \times LW$ matricization of DM.DE, the German version (Padó and Utt, 2012) of Distributional Memory (Baroni and Lenci, 2010). DM.DE was created on the basis of the 884M-token SDEWAC web corpus (Faaß et al., 2010), lemmatized, tagged, and parsed with the German MATE toolkit (Bohnet, 2010).

Experiments. We evaluate the impact of smoothing on two standard tasks from lexical semantics. The first task is predicting semantic similarity. We lemmatized and POS-tagged the German GUR350 dataset (Zesch et al., 2007), a set of 350 word pairs with human similarity judgments, created analogously to the well-known Rubenstein and Goodenough (1965) dataset for English.[2] We predict semantic similarity as cosine similarity. We make a prediction for a word pair if both words are represented in the semantic space and their vectors have a non-zero similarity.

The second task is synonym choice on the German version of the Reader's Digest WordPower dataset (Wallace and Wallace, 2005).[2] This dataset, which we also lemmatized and POS-tagged, consists of 984 target words with four synonym candidates each (including phrases), one of which is correct. Again, we compute semantic similarity as the cosine between target and a candidate vector and pick the highest-similarity candidate as synonym. For phrases, we compute the maximum pairwise word similarity. We make a prediction for an item if the target as well as at least one candidate are represented in the semantic space and their vectors have a non-zero similarity.

We expect differences between the two tasks with regard to derivational smoothing, since the words within derivational families are generally related but often not synonymous (cf. the example in Section 3). Thus, semantic similarity judgments should profit more easily from derivational smoothing than synonym choice.

Baseline. Our baseline is a standard bag-of-words vector space (BOW), which represents target words by the words occurring in their context. We use standard parameters ($\pm$10 word window, 8.000 most frequent verb, noun, and adjective lemmas). The model was created from the same corpus as DM.DE. We also applied derivational smoothing to this model, but did not obtain improvements.

Evaluation. To analyze the impact of smoothing, we evaluate the coverage of models and the quality of their predictions separately. In both tasks, coverage is the percentage of items for which we make a prediction. We measure quality of the semantic similarity task as the Pearson correlation between the model predictions and the human judgments for covered items (Zesch et al., 2007). For synonym choice, we follow the method established by Mohammad et al. (2007), measuring accuracy over covered items, with partial credit for ties.

Results for Semantic Similarity. Table 1 shows the results for the first task. The unsmoothed DM.DE model attains a correlation of $r = 0.44$ and a coverage of 58.9%. Smoothing increases the coverage substantially to 88%. Additionally, conservative, prototype-based smoothing (if $sim = 0$)

[2] Downloadable from: `http://goo.gl/bFokI`

Smoothing trigger	Smoothing scheme	r	Cov %
DM.DE, unsmoothed		.44	58.9
DM.DE, smooth always	avgSim	.30	88.0
	maxSim	.43	88.0
	centSim	.44	88.0
DM.DE, smooth if $sim = 0$	avgSim	.43	88.0
	maxSim	.42	88.0
	centSim	**.47**	88.0
BOW baseline		.36	94.9

Table 1: Results on the semantic similarity task (r: Pearson correlation, Cov: Coverage)

Smoothing trigger	Smoothing scheme	Acc %	Cov %
DM.DE, unsmoothed (Padó & Utt 2012)		53.7	80.8
DM.DE, smooth always	avgSim	46.0	86.6
	maxSim	50.3	86.6
	centSim	49.1	86.6
DM.DE, smooth if $sim = 0$	avgSim	52.6	86.6
	maxSim	51.2	86.6
	centSim	51.3	86.6
BOW "baseline"		**56.9**	98.5

Table 2: Results on the synonym choice task (Acc: Accuracy, Cov: Coverage)

increases correlation somewhat to $r = 0.47$. The difference to the unsmoothed model is not significant at $p = 0.05$ according to Fisher's (1925) method of comparing correlation coefficients.

The bag-of-words baseline (BOW) has a greater coverage than DM.DE models, but at the cost of lower correlation across the board. The only DM.DE model that performs worse than the BOW baseline is the non-conservative avgSim (average similarity) scheme. We attribute this weak performance to the presence of many pairwise zero similarities in the data, which makes the avgSim predictions unreliable.

To our knowledge, there are no previous published papers on distributional approaches to modeling this dataset. The best previous result is a GermaNet/Wikipedia-based model by Zesch et al. (2007). It reports a higher correlation ($r = 0.59$) but a very low coverage at 33.1%.

Results for Synonym Choice. The results for the second task are shown in Table 2. The unsmoothed model achieves an accuracy of 53.7% and a coverage of 80.8%, as reported by Padó and Utt (2012). Smoothing increases the coverage by almost 6% to 86.6% (for example, a question item for *inferior* becomes covered after backing off from the target to *Inferiorität* (*inferiority*)). All smoothed models show a loss in accuracy, albeit small. The best model is again a conservative smoothing model ($sim = 0$) with a loss of 1.1% accuracy. Using bootstrap resampling (Efron and Tibshirani, 1993), we established that the difference to the unsmoothed DM.DE model is not significant at $p < 0.05$. This time, the avgSim (average similarity) smoothing scheme performs best, with the prototype-based scheme in second place. Thus, the results for synonym choice are less clear-cut: derivational smoothing can trade accuracy against coverage but does not lead to a clear improvement. What is more, the BOW "baseline" significantly outperforms all syntactic models, smoothed and unsmoothed, with an almost perfect coverage combined with a higher accuracy.

6 Conclusions and Outlook

In this paper, we have introduced derivational smoothing, a novel strategy to combating sparsity in syntactic vector spaces by comparing and combining the vectors of morphologically related lemmas. The only information strictly necessary for the methods we propose is a grouping of lemmas into derivationally related classes. We have demonstrated that derivational smoothing improves two tasks, increasing coverage substantially and also leading to a numerically higher correlation in the semantic similarity task, even for vectors created from a very large corpus. We obtained the best results for a conservative approach, smoothing only zero similarities. This also explains our failure to improve less sparse word-based models, where very few pairs are assigned a similarity of zero. A comparison of prototype- and exemplar-based schemes did not yield a clear winner. The estimation of generic semantic similarity can profit more from derivational smoothing than the induction of specific lexical relations.

In future work, we plan to work on other evaluation tasks, application to other languages, and more sophisticated smoothing schemes.

Acknowledgments. Authors 1 and 3 were supported by the EC project EXCITEMENT (FP7 ICT-287923). Author 2 was supported by the Croatian Science Foundation (project 02.03/162: "Derivational Semantic Models for Information Retrieval"). We thank Jason Utt for his support and expertise.

References

James Allan and Giridhar Kumaran. 2003. Stemming in the Language Modeling Framework. In *Proceedings of SIGIR*, pages 455–456.

Harald R. Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. *The CELEX Lexical Database. Release 2. LDC96L14*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.

Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A General Framework for Corpus-based Semantics. *Computational Linguistics*, 36(4):673–721.

Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative Learning of Selectional Preference from Unlabeled Text. In *Proceedings of EMNLP*, pages 59–68, Honolulu, Hawaii.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Stanley F. Chen and Joshua Goodman. 1999. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language*, 13(4):359–394.

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning*, 34(1–3):43–69.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.

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.

Katrin Erk. 2012. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.

Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and Application of a Gold Standard for Morphological Analysis: SMOR in Validation. In *Proceedings of LREC-2010*, pages 803–810.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 363–370.

Ronald Aylmer Fisher. 1925. *Statistical methods for research workers*. Oliver and Boyd, Edinburgh.

Arne Fitschen. 2004. *Ein computerlinguistisches Lexikon als komplexes System*. Ph.D. thesis, IMS, Universität Stuttgart.

Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan M. Cigarrán. 1998. Indexing with WordNet Synsets Can Improve Text Retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montréal, Canada.

Rochelle Lieber. 2009. *Morphology and Lexical Semantics*. Cambridge University Press.

Saif Mohammad, Iryna Gurevych, Graeme Hirst, and Torsten Zesch. 2007. Cross-Lingual Distributional Profiles of Concepts for Measuring Semantic Distance. In *Proceedings of the 2007 Joint Conference on EMNLP and CoNLL*, pages 571–580, Prague, Czech Republic.

Roberto Navigli and Paola Velardi. 2003. An Analysis of Ontology-based Query Expansion Strategies. In *Workshop on Adaptive Text Extraction and Mining*, Dubrovnik, Croatia.

Sebastian Padó and Jason Utt. 2012. A Distributional Memory for German. In *Proceedings of KONVENS 2012 workshop on lexical-semantic resources and applications*, pages 462–470, Vienna, Austria.

Patrick Pantel and Dekang Lin. 2002. Discovering Word Senses from Text. In *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619.

Philip Resnik. 1996. Selectional Constraints: An Information-theoretic Model and its Computational Realization. *Cognition*, 61(1-2):127–159.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Ellen M. Voorhees. 1994. Query Expansion Using Lexical-semantic Relations. In *Proceedings of SIGIR*, pages 61–69.

DeWitt Wallace and Lila Acheson Wallace. 2005. *Reader's Digest, das Beste für Deutschland*. Verlag Das Beste, Stuttgart.

Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2005. Strictly Lexical Dependency Parsing. In *Proceedings of IWPT*, pages 152–159.

Britta Zeller, Jan Šnajder, and Sebastian Padó. 2013. DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of ACL*, Sofia, Bulgaria.

Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. 2007. Comparing Wikipedia and German Wordnet by Evaluating Semantic Relatedness on Multiple Datasets. In *Proceedings of NAACL/HLT*, pages 205–208, Rochester, NY.