Exemplar-Based Models for Word Meaning In Context

Katrin Erk
Department of Linguistics
University of Texas at Austin
katrin.erk@mail.utexas.edu

Sebastian Padó
Institut für maschinelle Sprachverarbeitung
Stuttgart University
pado@ims.uni-stuttgart.de

Abstract

This paper describes ongoing work on distributional models for word meaning in context. We abandon the usual one-vector-per-word paradigm in favor of an exemplar model that activates only relevant occurrences. On a paraphrasing task, we find that a simple exemplar model outperforms more complex state-of-the-art models.

1 Introduction

Distributional models are a popular framework for representing word meaning. They describe a lemma through a high-dimensional vector that records co-occurrence with context features over a large corpus. Distributional models have been used in many NLP analysis tasks (Salton et al., 1975; McCarthy and Carroll, 2003; Salton et al., 1975), as well as for cognitive modeling (Baroni and Lenci, 2009; Landauer and Dumais, 1997; McDonald and Ramscar, 2001). Among their attractive properties are their simplicity and versatility, as well as the fact that they can be acquired from corpora in an unsupervised manner.

Distributional models are also attractive as a model of word meaning in context, since they do not have to rely on fixed sets of dictionary sense with their well-known problems (Kilgarriff, 1997; McCarthy and Navigli, 2009). Also, they can be used directly for testing paraphrase applicability (Szpektor et al., 2008), a task that has recently become prominent in the context of textual entailment (Bar-Haim et al., 2007). However, polysemy is a fundamental problem for distributional models. Typically, distributional models compute a single "type" vector for a target word, which contains co-occurrence counts for all the occurrences of the target in a large corpus. If the target is polysemous, this vector mixes contextual features for all the senses of the target. For example, among the top 20 features for *coach*, we get *match* and *team* (for the "trainer" sense) as well as *driver* and *car* (for the "bus" sense). This problem has typically been approached by modifying the type vector for a target to better match a given context (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2009).

In the terms of research on human concept representation, which often employs feature vector representations, the use of type vectors can be understood as a *prototype*-based approach, which uses a single vector per category. From this angle, computing prototypes throws away much interesting distributional information. A rival class of models is that of *exemplar* models, which memorize each seen instance of a category and perform categorization by comparing a new stimulus to each remembered exemplar vector.

We can address the polysemy issue through an exemplar model by simply removing all exemplars that are "not relevant" for the present context, or conversely *activating* only the relevant ones. For the *coach* example, in the context of a text about motorways, presumably an instance like "The coach drove a steady 45 mph" would be activated, while "The team lost all games since the new coach arrived" would not.

In this paper, we present an exemplar-based distributional model for modeling word meaning in context, applying the model to the task of deciding paraphrase applicability. With a very simple vector representation and just using activation, we outperform the state-of-the-art prototype models. We perform an in-depth error analysis to identify stable parameters for this class of models.

2 Related Work

Among distributional models of word, there are some approaches that address polysemy, either by inducing a fixed clustering of contexts into senses (Schütze, 1998) or by dynamically modi-

fying a word's type vector according to each given sentence context (Landauer and Dumais, 1997; Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2009). Polysemy-aware approaches also differ in their notion of *context*. Some use a bag-of-words representation of words in the current sentence (Schütze, 1998; Landauer and Dumais, 1997), some make use of syntactic context (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2009). The approach that we present in the current paper computes a representation dynamically for each sentence context, using a simple bag-of-words representation of context.

In cognitive science, prototype models predict degree of category membership through similarity to a single prototype, while exemplar theory represents a concept as a collection of all previously seen exemplars (Murphy, 2002). Griffiths et al. (2007) found that the benefit of exemplars over prototypes grows with the number of available exemplars. The problem of representing meaning in context, which we consider in this paper, is closely related to the problem of *concept combination* in cognitive science, i.e., the derivation of representations for complex concepts (such as "metal spoon") given the representations of base concepts ("metal" and "spoon"). While most approaches to concept combination are based on prototype models, Voorspoels et al. (2009) show superior results for an exemplar model based on exemplar *activation*.

In NLP, exemplar-based (memory-based) models have been applied to many problems (Daelemans et al., 1999). In the current paper, we use an exemplar model for computing distributional representations for word meaning in context, using the context to *activate* relevant exemplars. Comparing representations of context, bag-of-words (BOW) representations are more informative and noisier, while syntax-based representations deliver sparser and less noisy information. Following the hypothesis that richer, topical information is more suitable for exemplar activation, we use BOW representations of sentential context in the current paper.

3 Exemplar Activation Models

We now present an exemplar-based model for meaning in context. It assumes that each target lemma is represented by a set of exemplars, where an exemplar is a sentence in which the target occurs, represented as a vector. We use lowercase letters for individual exemplars (vectors), and uppercase letters for sets of exemplars.

Sentential context	Paraphrase
After a fire extinguisher is used, it must always be **returned** for recharging and its use recorded.	bring back (3), take back (2), send back (1), give back (1)
We **return** to the young woman who is reading the Wrigley's wrapping paper.	come back (3), revert (1), revisit (1), go (1)

Table 1: The Lexical Substitution (LexSub) dataset.

We model polysemy by *activating* relevant exemplars of a lemma E in a given sentence context s. (Note that we use E to refer to both a lemma and its exemplar set, and that s can be viewed as just another exemplar vector.) In general, we define *activation* of a set E by exemplar s as

$$act(E, s) = \{e \in E \,|\, \mathrm{sim}(e, s) > \theta(E, s)\}$$

where E is an exemplar set, s is the "point of comparison", sim is some similarity measure such as Cosine or Jaccard, and $\theta(E, s)$ is a threshold. Exemplars belong to the activated set if their similarity to s exceeds $\theta(E, s)$.[1] We explore two variants of activation. In **kNN activation**, the k most similar exemplars to s are activated by setting θ to the similarity of the k-th most similar exemplar. In **q-percentage activation**, we activate the top $q\%$ of E by setting θ to the (100-q)-th percentile of the $sim(e, s)$ distribution. Note that, while in the kNN activation scheme the number of activated exemplars is the same for every lemma, this is not the case for percentage activation: There, a more frequent lemma (i.e., a lemma with more exemplars) will have more exemplars activated.

Exemplar activation for paraphrasing. A paraphrases is typically only applicable to a particular sense of a target word. Table 1 illustrates this on two examples from the Lexical Substitution (LexSub) dataset (McCarthy and Navigli, 2009), both featuring the target *return*. The right column lists appropriate paraphrases of *return* in each context (given by human annotators). [2] We apply the exemplar activation model to the task of predicting paraphrase felicity: Given a target lemma T in a particular sentential context s, and given a list of

[1] In principle, activation could be treated not just as binary inclusion/exclusion, but also as a graded weighting scheme. However, weighting schemes introduce a large number of parameters, which we wanted to avoid.

[2] Each annotator was allowed to give up to three paraphrases per target in context. As a consequence, the number of gold paraphrases per target sentence varies.

potential paraphrases of T, the task is to predict which of the paraphrases are applicable in s.

Previous approaches (Mitchell and Lapata, 2008; Erk and Padó, 2008; Erk and Padó, 2009; Thater et al., 2009) have performed this task by modifying the type vector for T to the context s and then comparing the resulting vector T' to the type vector of a paraphrase candidate P. In our exemplar setting, we select a contextually adequate subset of contexts in which T has been observed, using $T' = act(T, s)$ as a generalized representation of meaning of target T in the context of s.

Previous approaches used all of P as a representation for a paraphrase candidate P. However, P includes also irrelevant exemplars, while for a paraphrase to be judged as good, it is sufficient that one plausible reading exists. Therefore, we use $P' = act(P, s)$ to represent the paraphrase.

4 Experimental Evaluation

Data. We evaluate our model on predicting paraphrases from the Lexical Substitution (LexSub) dataset (McCarthy and Navigli, 2009). This dataset consists of 2000 instances of 200 target words in sentential contexts, with paraphrases for each target word instance generated by up to 6 participants. Paraphrases are ranked by the number of annotators that chose them (cf. Table 1). Following Erk and Padó (2008), we take the list of paraphrase candidates for a target as given (computed by pooling all paraphrases that LexSub annotators proposed for the target) and use the models to rank them for any given sentence context.

As exemplars, we create bag-of-words co-occurrence vectors from the BNC. These vectors represent instances of a target word by the other words in the same sentence, lemmatized and POS-tagged, minus stop words. E.g., if the lemma *gnurge* occurs twice in the BNC, once in the sentence "The dog will gnurge the other dog", and once in "The old windows gnurged", the exemplar set for *gnurge* contains the vectors *[dog-n: 2, other-a:1]* and *[old-a: 1, window-n: 1]*. For exemplar similarity, we use the standard Cosine similarity, and for the similarity of two exemplar sets, the Cosine of their centroids.

Evaluation. The model's prediction for an item is a list of paraphrases ranked by their predicted goodness of fit. To evaluate them against a weighted list of gold paraphrases, we follow Thater et al. (2009) in using Generalized Average Precision (GAP), which interpolates the precision values of top-n prediction lists for increasing n. Let $G = \langle q_1, \ldots, q_m \rangle$ be the list of gold paraphrases with gold weights $\langle y_1, \ldots, y_m \rangle$. Let $P = \langle p_1, \ldots, p_n \rangle$ be the list of model predictions as ranked by the model, and let $\langle x_1, \ldots, x_n \rangle$ be the *gold* weights associated with them (assume $x_i = 0$ if $p_i \notin G$), where $G \subseteq P$. Let $I(x_i) = 1$ if $p_i \in G$, and zero otherwise. We write $\overline{x_i} = \frac{1}{i} \sum_{k=1}^{i} x_k$ for the average gold weight of the first i model predictions, and analogously $\overline{y_i}$. Then

$$GAP(P, G) = \frac{1}{\sum_{j=1}^{m} I(y_j)\overline{y_j}} \sum_{i=1}^{n} I(x_i)\overline{x_i}$$

Since the model may rank multiple paraphrases the same, we average over 10 random permutations of equally ranked paraphrases. We report mean GAP over all items in the dataset.

para-	*actT*		*actP*	
meter	kNN	perc.	kNN	perc.
10	36.1	35.5	36.5	**38.6**
20	**36.2**	35.2	36.2	37.9
30	36.1	35.3	35.8	37.8
40	36.0	35.3	35.8	37.7
50	35.9	35.1	35.9	37.5
60	36.0	35.0	36.1	37.5
70	35.9	34.8	36.1	37.5
80	36.0	34.7	36.0	37.4
90	35.9	34.5	35.9	37.3
no act.	34.6		35.7	
random BL	28.5			

Table 2: Activation of T or P individually on the full LexSub dataset (GAP evaluation)

Results and Discussion. We first computed two models that activate either the paraphrase or the target, but not both. Model 1, *actT*, activates only the target, using the complete P as paraphrase, and ranking paraphrases by $sim(P, act(T, s))$. Model 2, *actP*, activates only the paraphrase, using s as the target word, ranking by $sim(act(P, s), s)$.

The results for these models are shown in Table 2, with both kNN and percentage activation: kNN activation with a parameter of 10 means that the 10 closest neighbors were activated, while percentage with a parameter of 10 means that the closest 10% of the exemplars were used. Note first that we computed a random baseline (last row) with a GAP of 28.5. The second-to-last row ("no activation") shows two more informed baselines.

The *actT* "no act" result (34.6) corresponds to a prototype-based model that ranks paraphrase candidates by the distance between their type vectors and the target's type vector. Virtually all exemplar models outperform this prototype model. Note also that both *actT* and *actP* show the best results for small values of the activation parameter. This indicates paraphrases can be judged on the basis of a rather small number of exemplars. Nevertheless, *actT* and *actP* differ with regard to the details of their optimal activation. For *actT*, a small absolute number of activated exemplars (here, 20) works best , while *actP* yields the best results for a small percentage of paraphrase exemplars. This can be explained by the different functions played by *actT* and *actP* (cf. Section 3): Activation of the paraphrase must allow a guess about whether there is reasonable interpretation of P in the context s. This appears to require a reasonably-sized sample from P. In contrast, target activation merely has to counteract the sparsity of s, and activation of too many exemplars from T leads to oversmoothing.

We obtained significances by computing 95% and 99% confidence intervals with bootstrap resampling. As a rule of thumb, we find that 0.4% difference in GAP corresponds to a significant difference at the 95% level, and 0.7% difference in GAP to significance at the 99% level. The four activation methods (i.e., columns in Table 2) are significantly different from each other, with the exception of the pair *actT*/kNN and *actP*/kNN (n.s.), so that we get the following order:

$$\textit{actP}/\text{perc} > \textit{actP}/\text{kNN} \approx \textit{actT}/\text{kNN} > \textit{actT}/\text{perc}$$

where $>$ means "significantly outperforms". In particular, the best method (*actT*/kNN) outperforms all other methods at $p<0.01$. Here, the best parameter setting (10% activation) is also significantly better than the next-one one (20% activation). With the exception of *actT*/perc, all activation methods significantly outperform the best baseline (*actP*, no activation).

Based on these observations, we computed a third model, *actTP*, that activates both T (by kNN) and P (by percentage), ranking paraphrases by $sim(act(P,s), act(T,s))$. Table 3 shows the results. We find the overall best model at a similar location in parameter space as for *actT* and *actP* (cf. Table 2), namely by setting the activation parameters to small values. The sensitivity of the parameters changes considerably, though. When we fix the *actP* activation level, we find comparatively large performance differences between the T activation settings k=5 and k=10 (highly significant for 10% *actP*, and significant for 20% and 30% *actP*). On the other hand, when we fix the *actT* activation level, changes in *actP* activation generally have an insignificant impact.

P activation (%) $\Rightarrow$	10	20	30
T activation (kNN) $\Downarrow$			
5	**38.2**	38.1	38.1
10	37.6	37.8	37.7
20	37.3	37.4	37.3
40	37.2	37.2	36.1

Table 3: Joint activation of P and T on the full LexSub dataset (GAP evaluation)

Somewhat disappointingly, we are not able to surpass the best result for *actP* alone. This indicates that – at least in the current vector space – the sparsity of s is less of a problem than the "dilution" of s that we face when we representing the target word by exemplars of T close to s. Note, however, that the numerically worse performance of the best *actTP* model is still not significantly different from the best *actP* model.

Influence of POS and frequency. An analysis of the results by target part-of-speech showed that the globally optimal parameters also yield the best results for individual POS, even though there are substantial differences among POS. For *actT*, the best results emerge for all POS with kNN activation with k between 10 and 30. For k=20, we obtain a GAP of 35.3 (verbs), 38.2 (nouns), and 35.1 (adjectives). For *actP*, the best parameter for all POS was activation of 10%, with GAPs of 36.9 (verbs), 41.4 (nouns), and 37.5 (adjectives). Interestingly, the results for *actTP* (verbs: 38.4, nouns: 40.6, adjectives: 36.9) are better than *actP* for verbs, but worse for nouns and adjectives, which indicates that the sparsity problem might be more prominent than for the other POS. In all three models, we found a clear effect of target and paraphrase frequency, with deteriorating performance for the highest-frequency targets as well as for the lemmas with the highest average paraphrase frequency.

Comparison to other models. Many of the other models are syntax-based and are therefore only applicable to a subset of the LexSub data. We have re-evaluated our exemplar models on the subsets we used in Erk and Padó (2008, EP08, 367

	Models		
	EP08	EP09	TDP09
EP08 dataset	27.4	NA	NA
EP09 dataset	NA	32.2	36.5
	actT	*actP*	*actTP*
EP08 dataset	36.5	38.0	**39.9**
EP09 dataset	39.1	**39.9**	39.6

Table 4: Comparison to other models on two subsets of LexSub (GAP evaluation)

datapoints) and Erk and Padó (2009, EP09, 100 datapoints). The second set was also used by Thater et al. (2009, TDP09). The results in Table 4 compare these models against our best previous exemplar models and show that our models outperform these models across the board. [3] Due to the small sizes of these datasets, statistical significance is more difficult to attain. On EP09, the differences among our models are not significant, but the difference between them and the original EP09 model is.[4] On EP08, all differences are significant except for *actP* vs. *actTP*.

We note that both the EP08 and the EP09 datasets appear to be simpler to model than the complete Lexical Substitution dataset, at least by our exemplar-based models. This underscores an old insight: namely, that direct syntactic neighbors, such as arguments and modifiers, provide strong clues as to word sense.

5 Conclusions and Outlook

This paper reports on work in progress on an exemplar activation model as an alternative to one-vector-per-word approaches to word meaning in context. Exemplar activation is very effective in handling polysemy, even with a very simple (and sparse) bag-of-words vector representation. On both the EP08 and EP09 datasets, our models surpass more complex prototype-based approaches (Tab. 4). It is also noteworthy that the exemplar activation models work best when few exemplars are used, which bodes well for their efficiency.

We found that the best target representations result from activating a low absolute number of exemplars. Paraphrase representations are best activated with a percentage-based threshold. Overall, we found that paraphrase activation had a much larger impact on performance than target activation, and that drawing on target exemplars other than s to represent the target meaning in context improved over using s itself only for verbs (Tab. 3). This suggests the possibility of considering T's activated paraphrase candidates as the representation of T in the context s, rather than some vector of T itself, in the spirit of Kintsch (2001).

While it is encouraging that the best parameter settings involved the activation of only few exemplars, computation with exemplar models still requires the management of large numbers of vectors. The computational overhead can be reduced by using data structures that cut down on the number of vector comparisons, or by decreasing vector dimensionality (Gorman and Curran, 2006). We will experiment with those methods to determine the tradeoff of runtime and accuracy for this task.

Another area of future work is to move beyond bag-of-words context: It is known from WSD that syntactic and bag-of-words contexts provide complementary information (Florian et al., 2002; Szpektor et al., 2008), and we hope that they can be integrated in a more sophisticated exemplar model.

Finally, we will to explore task-based evaluations. Relation extraction and textual entailment in particular are tasks where similar models have been used before (Szpektor et al., 2008).

Acknowledgements. This work was supported in part by National Science Foundation grant IIS-0845925, and by a Morris Memorial Grant from the New York Community Trust.

[3] Since our models had the advantage of being tuned on the dataset, we also report the range of results across the parameters we tested. On the EP08 dataset, we obtained 33.1–36.5 for *actT*; 33.3–38.0 for *actP*; 37.7-39.9 for *actTP*. On the EP09 dataset, the numbers were 35.8–39.1 for *actT*; 38.1–39.9 for *actP*; 37.2–39.8 for *actTP*.

[4] We did not have access to the TDP09 predictions to do significance testing.

References

R. Bar-Haim, I. Dagan, I. Greental, and E. Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *Proceedings of AAAI*, pages 871–876, Vancouver, BC.

M. Baroni and A. Lenci. 2009. One distributional memory, many semantic spaces. In *Proceedings of the EACL Workshop on Geometrical Models of Natural Language Semantics*, Athens, Greece.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1/3):11–43. Special Issue on Natural Language Learning.

K. Erk and S. Padó. 2008. A structured vector space

model for word meaning in context. In *Proceedings of EMNLP*, pages 897–906, Honolulu, HI.

K. Erk and S. Padó. 2009. Paraphrase assessment in structured vector space: Exploring parameters and datasets. In *Proceedings of the EACL Workshop on Geometrical Models of Natural Language Semantics*, Athens, Greece.

R. Florian, S. Cucerzan, C. Schafer, and D. Yarowsky. 2002. Combining classifiers for word sense disambiguation. *Journal of Natural Language Engineering*, 8(4):327–341.

J. Gorman and J. R. Curran. 2006. Scaling distributional similarity to large corpora. In *Proceedings of ACL*, pages 361–368, Sydney.

T. Griffiths, K. Canini, A. Sanborn, and D. J. Navarro. 2007. Unifying rational models of categorization via the hierarchical Dirichlet process. In *Proceedings of CogSci*, pages 323–328, Nashville, TN.

A. Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

W. Kintsch. 2001. Predication. *Cognitive Science*, 25:173–202.

T. Landauer and S. Dumais. 1997. A solution to Platos problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

D. McCarthy and J. Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

D. McCarthy and R. Navigli. 2009. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159. Special Issue on Computational Semantic Analysis of Language: SemEval-2007 and Beyond.

S. McDonald and M. Ramscar. 2001. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of CogSci*, pages 611–616.

J. Mitchell and M. Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, pages 236–244, Columbus, OH.

G. L. Murphy. 2002. *The Big Book of Concepts*. MIT Press.

G Salton, A Wang, and C Yang. 1975. A vector-space model for information retrieval. *Journal of the American Society for Information Science*, 18:613–620.

H. Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

I. Szpektor, I. Dagan, R. Bar-Haim, and J. Goldberger. 2008. Contextual preferences. In *Proceedings of ACL*, pages 683–691, Columbus, OH.

S. Thater, G. Dinu, and M. Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the ACL Workshop on Applied Textual Inference*, pages 44–47, Singapore.

W. Voorspoels, W. Vanpaemel, and G. Storms. 2009. The role of extensional information in conceptual combination. In *Proceedings of CogSci*.