Ranking Paraphrases in Context

Stefan Thater
Universität des Saarlandes
stth@coli.uni-sb.de

Georgiana Dinu
Universität des Saarlandes
dinu@coli.uni-sb.de

Manfred Pinkal
Universität des Saarlandes
pinkal@coli.uni-sb.de

Abstract

We present a vector space model that supports the computation of appropriate vector representations for words in context, and apply it to a paraphrase ranking task. An evaluation on the SemEval 2007 lexical substitution task data shows promising results: the model significantly outperforms a current state of the art model, and our treatment of context is effective.

1 Introduction

Knowledge about paraphrases is of central importance to textual inference modeling. Systems which support automatic extraction of large repositories of paraphrase or inference rules like Lin and Pantel (2001) or Szpektor et al. (2004) thus form first-class candidate resources to be leveraged for NLP tasks like question answering, information extraction, or summarization, and the meta-task of recognizing textual entailment.

Existing knowledge bases still suffer a number of limitations, making their use in applications challenging. One of the most serious problems is insensitivity to context. Natural-language inference is highly context-sensitive, the applicability of inference rules depending on word sense and even finer grained contextual distinctions in usage (Szpektor et al., 2007). Application of a rule like "*X shed Y* $\Leftrightarrow$ *X throw Y*" is appropriate in a sentence like "a mouse study sheds light on the mixed results," but not in sentences like "the economy seems to be shedding fewer jobs" or "cats do not shed the virus to other cats." Systems like the above-mentioned ones base the extraction of inference rules on distributional similarity of words rather than word senses, and apply unconditionally whenever one side of the rule matches on the word level, which may lead to considerable precision problems (Geffet and Dagan, 2005) .

Some approaches address the problem of context sensitivity by deriving inference rules whose argument slots bear selectional preference information (Pantel et al., 2007; Basili et al., 2007). A different line of accounting for contextual variation has been taken by Mitchell and Lapata (2008), who propose a compositional approach, "contextualizing" the vector-space meaning representation of predicates by combining the distributional properties of the predicate with those of its arguments. A related approach has been proposed by Erk and Padó (2008), who integrate selectional preferences into the compositional picture. In this paper, we propose a context-sensitive vector-space approach which draws some important ideas from Erk and Pado's paper ("E&P" in the following), but implements them in a different, more effective way: An evaluation on the SemEval 2007 lexical substitution task data shows that our model significantly outperforms E&P in terms of average precision.

Plan of the paper. Section 2 presents our model and briefly relates it to previous work. Section 3 describes the evaluation of our model on the lexical substitution task data. Section 4 concludes.

2 A model for meaning in context

We propose a dependency-based model whose dimensions reflect dependency relations, and distinguish two kinds or layers of lexical meaning: *argument meaning* and *predicate meaning.* The argument meaning of a word w is a vector representing frequencies of all pairs (w', r') of predicate expressions w' and dependency relations r' such that w' stands in relation r' to w. Intuitively, argument meaning is similar to E&P's "inverse selectional preferences." Argument meanings are used for two purposes in our model: (i) to construct predicate meanings, and (ii) to contextually constrain them.

For technical convenience, we will use a definitional variant of argument meaning, by indexing it with an "incoming" relation, which allows predicate and argument meaning to be treated technically as vectors of the same type. Assuming a set

Proceedings of the 2009 Workshop on Applied Textual Inference, ACL-IJCNLP 2009, pages 44–47,
Suntec, Singapore, 6 August 2009.

R of role labels and a set W of words, we represent both predicate and argument meaning as vectors in a vector space V with a basis $\{e_i\}_{i \in R \times R \times W}$, i.e., a vector space whose dimensions correspond to triples of two role labels and a word. The argument meaning $v_r(w)$ of a word w is defined as follows:

$$v_r(w) = \sum_{w' \in W, r' \in R} f(w', r', w) \cdot e_{(r,r',w')}, \quad (1)$$

where r is the "incoming" relation, and $f(w', r', w)$ denotes the frequency of w occurring in relation r' to w' in a collection of dependency trees. To obtain predicate meaning $v_P(w)$, we count the occurrences of argument words w' standing in relation r to w, and compute the predicate meaning as the sum of the argument meanings $v_r(w')$, weighted by these co-occurrence frequencies:

$$v_P(w) = \sum_{r \in R, w' \in W} f(w, r, w') \cdot v_r(w') \quad (2)$$

That is, the meaning of a predicate is modelled by a vector representing "second order" co-coccurrence frequencies with other predicates.

In general, words have both a "downward looking" predicate meaning and an "upward looking" argument meaning. In our study, only one of them will be relevant, since we will restrict ourselves to local predicate-argument structures with verbal heads and nominal arguments.

Computing meaning in context. Vectors representing predicate meaning are derived by collecting co-occurrence frequencies for all uses of the predicate, possibly resulting in vector representations in which different meanings of the predicate are combined. Given an instance of a predicate w that has arguments $w_1, \ldots, w_k$, we can now contextually constrain the predicate meaning of w by the argument meanings of its arguments. Here, we propose to simple "restrict" the predicate meaning to those dimensions that have a non-zero value in at least one of its argument meanings. More formally, we write $v_{|v'}$ to denote a vector that is identical to v for all components that have a non-zero value in v', zero otherwise. We compute *predicate meaning in context* as follows:

$$v_P(w)_{|\sum_{1 \leq i \leq k} v_{r_i}(w_i)}, \quad (3)$$

where r_i is the argument position filled by w_i.

Parameters. To reduce the effect of noise and provide a more fine-grained control over the effect of context, we can choose different thresholds for function f in the computation of predicate and argument meaning. In Section 3, we obtain best results if we consider only dependency relations that occur at least 6 times in the British National Corpus (BNC) for the computation of predicate meaning, and relations occurring at least 15 times for the computation of argument meanings when predicate meaning is contextually constrained.

target	*subject*	*object*	*paraphrases*
shed	study	light	throw 3, reveal 2, shine 1
shed	cat	virus	spread 2, pass 2, emit 1, transmit 2
shed	you	blood	lose 3, spill 1, give 1

Table 1: Lexical substitution task data set

Related work. Our model is similar to the structured vector space model proposed by Erk and Padó (2008) in that the representation of predicate meaning is based on dependency relations, and that "inverse selectional preferences" play an important role. However, inverse selectional preferences are used in E&P's model mainly to compute meaning in context, while they are directly "built into" the vectors representing predicate meaning in our model.

3 Evaluation

We evaluate our model on a paraphrase ranking task on a subset of the SemEval 2007 lexical substitution task (McCarthy and Navigli, 2007) data, and compare it to a random baseline and E&P's state of the art model.

Dataset. The lexical substitution task dataset contains 10 instances for 44 target verbs in different sentential contexts. Systems that participated in the task had to generate paraphrases for each of these instances, which are evaluated against a gold standard containing up to 9 possible paraphrases for individual instances. Following Erk and Padó (2008), we use the data in a different fashion: we pool paraphrases for all instances of a verb in all contexts, and use the models to rank these paraphrase candidates in specific contexts.

Table 1 shows three instances of the target verb *shed* together with its paraphrases in the gold standard as an expample. The paraphrases are attached with weights, which correspond to the number of times they have been given by different annotators.

To allow for a comparision with E&P's model, we follow Erk and Padó (2008) and extract only sentences from the dataset containing target verbs

with overtly realized subject and object, and remove instances from the dataset for which the target verb or one of its arguments is not in the BNC. We obtain a set of 162 instances for 34 different verbs. We also remove paraphrases that are not in the BNC. On average, target verbs have 20.5 paraphrase candidates, 3.9 of which are correct in specific contexts.

Experimental setup. We parse the BNC using MiniPar (Lin, 1993) and extract co-occurrence frequencies, considering only dependency relations for the most frequent 2000 verbs. We don't use raw frequency counts directly but reweight the vectors by pointwise mutual information.

To rank paraphrases in context, we compute contextually constrained vectors for the verb in the input sentence and all its paraphrase candidates by taking the corresponding predicate vectors and restricting them to the argument meanings of the argument head nouns in the input sentence. The restricted vectors for the paraphrase candidates are then ranked by comparing them to the restricted vector of the input verb using cosine similarity.

In order to compare our model with state of the art, we reimplement E&P's structured vector space model. We filter stop words, and compute lexical vectors in a "syntactic" space using the most frequent 2000 words from the BNC as basis. We also consider a variant in which the basis corresponds to words indexed by their grammatical roles. We choose parameters that Erk and Padó (2009) report to perform best, and use the method described in Erk and Padó (2009) to compute vectors in context.

Evaluation metrics. As scoring methods, we use both "precision out of ten" (P_{oot}), which was originally used in the lexical substitution task and also used by E&P, and *generalized average precision* (Kishida, 2005), a variant of *average precision* which is frequently used in information extraction tasks and has also been used in the PASCAL RTE challenges (Dagan et al., 2006).

P_{oot} can be defined as follows:

$$P_{oot} = \frac{\Sigma_{s \in M \cap G} f(s)}{\Sigma_{s \in G} f(s)},$$

where M is the list of 10 paraphrase candidates top-ranked by the model, G is the corresponding annotated gold data, and $f(s)$ is the weight of the individual paraphrases. Here, P_{oot} is computed for each target instance separately; below, we report the average over all instances.

Model	P_{oot}	*GAP*
Random baseline	54.25	26.03
E&P (target only)	64.61 (63.31)	29.95 (32.02)
E&P (add, object only)	**66.20** (62.90)	29.93 (31.54)
E&P (min, both)	64.86 (59.62)	**32.22** (31.28)
TDP	63.32	**36.54**
TDP (target only)	62.60	33.04

Table 2: Results

Generalized average precision (*GAP*) is a more precise measure than P_{oot}: Applied to a ranking task with about 20 candidates, P_{oot} just gives the percentage of good candidates found in the upper half of the proposed ranking. Average precision is sensitive to the relative position of correct and incorrect candidates in the ranking, *GAP* moreover rewards the correct order of positive cases w.r.t. their gold standard weight.

We define average precision first:

$$AP = \frac{\Sigma_{i=1}^{n} x_i \, p_i}{R} \qquad p_i = \frac{\Sigma_{k=1}^{i} x_k}{i}$$

where x_i is a binary variable indicating whether the ith item as ranked by the model is in the gold standard or not, R is the size of the gold standard, and n the number of paraphrase candidates to be ranked. If we take x_i to be the gold standard weight of the ith item or zero if it is not in the gold standard, we can define *generalized average precision* as follows:

$$GAP = \frac{\Sigma_{i=1}^{n} I(x_i) \, p_i}{R'} \qquad R' = \Sigma_{i=1}^{R} I(y_i) \bar{y}_i$$

where $I(x_i) = 1$ if x_i is larger than zero, zero otherwise, and $\bar{y}_i$ is the average weight of the ideal ranked list $y_1, \ldots, y_i$ of paraphrases in the gold standard.

Results and discussion. Table 2 shows the results of our experiments for two variants of our model ("TDP"), and compares them to a random baseline and three instantiations (in two variants) of E&P's model. The "target only" models don't use context information, i.e., paraphrases are ranked by cosine similarity of predicate meaning only. The other models take context into account. The "min" E&P model takes the component-wise minimum to combine a lexical vector with context vectors and considers both subject and object as context; it is the best performing model in Erk and Padó (2009). The "add" model uses vector addition and considers only objects as context; it is the best-performing

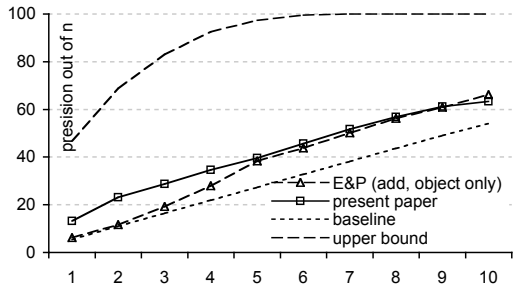

Figure 1: "Precision out of n" for $1 \leq n \leq 10$.

model (in terms of P_{oot}) for our dataset. The numbers in brackets refer to variants of the E&P models in which the basis corresponds to words indexed by their syntactic roles. Note that the results for the E&P models are better than the results published in Erk and Padó (2009), which might be due to slightly different datasets or lists of stop-words.

As can be seen, our model performs $> 10\%$ better than the random baseline. It performs $> 4\%$ better than the "min" E&P model and $> 6\%$ better then the "add" model in terms of *GAP* if we use a vectors space with words as basis. For the variants of the E&P models in which the basis corresponds to words indexed by their syntactic role, we obtain different results, but our model is still $> 4\%$ better than these variants. We can also see that our treatment of context is effective, leading to a $> 3\%$ increase of *GAP*. A stratified shuffling-based randomization test (Yeh, 2000) shows that the differences are statistically significant ($p < 0.05$).

In terms of P_{oot}, the "add" E&P model performs better than our model, which might look surprising, given its low *GAP* score. Fig. 1 gives a more fine-grained comparison between the two models. It displays the "precision out of n" of the two models for varying n. As can be seen, our model performs better for all $n < 10$, and much better than the baseline and E&P for $n \leq 4$.

4 Conclusion

In this paper, we have proposed a dependency-based context-sensitive vector-space approach that supports the computation of adequate vector-based representations of predicate meaning in context. An evaluation on a paraphrase ranking task using a subset of the SemEval 2007 lexical substitution task data shows promising results: our model performs significantly better than a current state of the art system (Erk and Padó, 2008), and our treatment of context is effective.

Since the dataset we used for the evaluation is relatively small, there is a potential danger for overfitting, and it remains to be seen whether the results carry over to larger datasets. First experiments indicate that this is actually the case.

We expect that our approach can be generalized to arrive at a general compositional model, which would allow to compute contextually appropriate meaning representations for complex relational expressions rather than single lexical predicates.

Acknowledgements. We thank Katrin Erk and Sebastian Padó for help and critical comments.

References

R. Basili, D. De Cao, P. Marocco, and M. Pennacchiotti. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *Proc. of RANLP 2007*.

I. Dagan, O. Glickman, and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges*, volume 3944. Springer.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. In *Proc. of EMNLP*.

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

M. Geffet and I. Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proc. of the ACL*.

K. Kishida. 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. *NII Technical Report*.

D. Lin and P. Pantel. 2001. DIRT – Discovery of Inference Rules from Text. In *Proc. of the ACM Conference on Knowledge Discovery and Data Mining*, San Francisco.

D. Lin. 1993. Principle-based parsing without overgeneration. In *Proc. of ACL*, Columbus.

D. McCarthy and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proc. of SemEval*, Prague.

J. Mitchell and M. Lapata. 2008. Vector-based models of semantic composition. In *Proc. of ACL-08: HLT*, Columbus.

P. Pantel, R. Bhagat, B. Coppola, T. Chklovski, and E. Hovy. 2007. ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007*, Rochester.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proc. of EMNLP*, Barcellona.

I. Szpektor, E. Shnarch, and I. Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proc. of ACL*.

A. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proc. of COLING*.