Topic models for meaning similarity in context

Georgiana Dinu
Dept. of Computational Linguistics
Saarland University
dinu@coli.uni-sb.de

Mirella Lapata
School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

Recent work on distributional methods for similarity focuses on using the context in which a target word occurs to derive context-sensitive similarity computations. In this paper we present a method for computing similarity which builds vector representations for words *in context* by modeling senses as latent variables in a large corpus. We apply this to the Lexical Substitution Task and we show that our model significantly outperforms typical distributional methods.

1 Introduction

Distributional methods for word similarity ((Landauer and Dumais, 1997), (Schuetze, 1998)) are based on co-occurrence statistics extracted from large amounts of text. Typically, each word is assigned a representation as a point in a high-dimensional space, where the dimensions represent contextual features such as co-occurring words. Following this, meaning relatedness scores are computed by using various similarity measures on the vector representations.

One of the major issues that all distributional methods have to face is sense ambiguity. Since vector representations reflect *mixtures of uses* additional methods have to be employed in order to capture specific meanings of a word in context. Consider the occurrence of verb *shed* in the following SemEval 2007 Lexical Substitution Task (McCarthy and Navigli, 2007) example:

Cats in the latent phase only have the virus internally , but feel normal and do not ***shed*** *the virus to other cats and the environment .*

Human participants in this task provided words such as *transmit* and *spread* as good substitutes for *shed* in this context, however a vector space representation of *shed* will not capture this infrequent sense.

For these reasons, recent work on distributional methods for similarity such as (Mitchell and Lapata, 2008) (Erk and Padó, 2008) (Thater et al., 2009) focuses on using the context in which a target word occurs to derive context-sensitive similarity computations.

In this paper we present a method for computing similarity which builds vector representations for words *in context*. Most distributional methods so far extract representations from large texts, and only as a follow-on step they either 1) alter these in order to reflect a *disambiguated* word (such as (Erk and Padó, 2008)) or 2) directly asses the appropriateness of a similarity judgment, given a specific context (such as (Pantel et al., 2007)). Our approach differs from this as we assume ambiguity of words at the, initial, acquisition step, by encoding senses of words as a hidden variable in the text we process.

In this paper we focus on a particular distributional representation inspired by (Lin and Pantel, 2001a) and induce context-sensitive similarity between phrases represented as paths in dependency graphs. It is inspired by recent work on topic models and it deals with sense-ambiguity in a natural manner by modeling senses as latent variables in a large corpus. We apply this to the Lexical Substitution Task and we show that our model outperforms the (Lin and Pantel, 2001a) method by inducing context-appropriate similarity judgments.

Coling 2008: Poster Volume, pages 250–258,
Beijing, August 2010

2 Related work

Discovery of Inference Rules from Text (DIRT) A popular distributional method for meaning relatedness is the DIRT algorithm for extracting inference rules (Lin and Pantel, 2001a). In this algorithm a $pattern$ is a noun-ending path in a dependency graph and the goal is to acquire pairs of patterns for which entailment holds (in at least one direction) such as *(X solve Y, X find solution to Y)*.

The method can be seen a particular instance of a vector space. Each pattern is represented by the sets of its left hand side (X) and right hand side (Y) noun fillers in a large corpus. Two patterns are compared in the X-filler space, and correspondingly in the Y-filler space by using the Lin similarity measure:

$$\mathrm{sim}_{\mathrm{Lin}}(v, w) = \frac{\sum_{i \in I(v) \cap I(w)} (v_i + w_i)}{\sum_{i \in I(v)} v_i + \sum_{l \in I(w)} w_i}$$

where values in v and w are point-wise mutual information, and $I(\cdot)$ gives the indices of positive values in a vector.

The final similarity score between two patterns is obtained by multiplying the X and Y similarity scores. Table 1 shows a fragment of a DIRT-like vector space.

	..	*case*	*problem*	..
(*X solve Y,* ***Y***)	..	6.1	4.4	..
(*X settle Y,* ***Y***)	..	5.2	5.9	..

Table 1: DIRT-like vector representation in the **Y**-filler space. The values represent mutual information.

Further on, this similarity method is used for the task of paraphrasing. A total set of patterns is extracted from a large corpus and each of them can be paraphrased by returning its most similar patterns, according to the similarity score. Although relatively accurate[1], it has been noted (Lin and Pantel, 2001b) that the paraphrases extracted this way reflect, as expected, various meanings, and that a context-sensitive representation would be appropriate.

Context-sensitive extensions of DIRT (Pantel et al., 2007) and (Basili et al., 2007) focus on making DIRT rules context-sensitive by attaching appropriate semantic classes to the X and Y slots of an inference rule. For this purpose, the initial step in their methods is to acquire an inference rule database, using the DIRT algorithm. Following this, given an inference rule, they identify semantic classes for the X and Y fillers which make the application of the rule appropriate. For this (Pantel et al., 2007) build a set of semantic classes using WordNet in one case and CBC clustering algorithm in the other; for each rule, they use the overlap of the fillers found in the input corpus as an indicator of the correct semantic classes. The same idea is used in (Basili et al., 2007) where, this time, the X and Y fillers are clustered for each rule individually; these nouns are clustered using an LSA-vector representation extracted from a large corpus.

(Connor and Roth, 2007) take a slightly different approach as they attempt to classify the context of a rule as appropriate or not, again using the overlap of fillers as an indicator. They all show improvement over DIRT by evaluating on occurrences of rules in context which are annotated as correct/incorrect by human participants. On a common data set (Pantel et al., 2007) and (Basili et al., 2007) achieve significant improvements over DIRT at 95% confidence level when employing the clustering methods. (Szpektor et al., 2008) propose a general framework for these methods and show that some of these settings obtain significant (level 0.01) improvements over the DIRT algorithm on data derived from the ACE 2005 event detection task.

Related work on topic models Topic models have been previously used for semantic tasks. Work such as (Cai et al., 2007) or (Boyd-Graber et al., 2007) use the document-level topics extracted with Latent Dirichlet Allocation (LDA) as indicators of meanings for word sense disambiguation. More related to our work are (Brody and Lapata, 2009) or (Toutanova and Johnson, 2008) who use LDA-based models which induce latent variables from task-specific data rather than from simple documents.

[1] Precision is estimated to lie around 50% for the most confident paraphrases

X solve Y
we-X:122, country-X:89, government-X:82, it-X:69,..., problem-Y:1088, issue-Y:134, crisis-Y:99, dispute-Y:78,...

Table 2: Fragments of the *document* associated to *X solve Y*. *we-X: 122* indicates that *X solve Y* occurs 122 times with we as an X filler.

(Brody and Lapata, 2009) apply such a model for word sense induction on a set of 35 target nouns. They assume senses as latent variables and context features as observations; unlike our model they induce local senses specific to every target word by estimating separate models with the final goal of explicitly inducing word senses.

(Toutanova and Johnson, 2008) use an LDA-based model for semi-supervised part-of-speech tagging. They build a word context model in which each token involves: generating a distribution over tags, sampling a tag, and finally generating context words according to a tag-specific word distribution (context words are observations). Their model achieves highest performance when combined with a ambiguity class component which uses a dictionary for possible tags of target words.

Both these papers show improvements over state-of-the-art systems for their tasks.

3 Generative model for similarity in context

We develop a method for computing similarity of patterns in context, i.e. patterns with instantiated X and Y values. We do not enhance the representation of an inference rule with sense (context-appropriateness) information but rather focus on the task of assigning similarity scores to such pairs of *instantiated* patterns. Unlike previous work, we do not employ any other additional resources, investigating this way whether structurally richer information can be learned from the same input co-occurrence matrix as the original DIRT method.

Our model, as well as the DIRT algorithm, uses *context* information extracted from large corpora to learn similarities between *patterns*; however ideally we would like to learn contextual preferences (or, in general, some form of sense-disambiguation) for these patterns. This is achieved in our model by assuming an intermediate layer consisting of *meanings* (senses): the *context* surrounding a pattern is indicative of *meanings*, and preference for some *meanings* gives the characterization of a *pattern*.

For this we use a generative model inspired by Latent Dirichlet Allocation (Blei et al., 2003) (Griffiths and Steyvers, 2004) which is successfully employed for modeling collections of documents and the underlying topics which occur in them. The statistical model is characterized by the following distributions:

$$\begin{array}{rl} w_i|z_i, \phi^{z_i} & Discrete(\phi^{z_i}) \\ \phi^z & Dirichlet(\beta) \\ z_i|\theta^p & Discrete(\theta^p) \\ \theta^p & Dirichlet(\alpha) \end{array}$$

θ^p is the distribution over meanings associated to a pattern p and ϕ^z is the distribution over words associated to a meaning z. The occurrence of each filler word w_i with a pattern p, is then generated by sampling 1) a meaning conditioned on the meaning distribution associated to p: $z_i|\theta^p$ and 2) a word conditioned on the word distribution associated to the meaning z_i: $w_i|z_i, \phi^{z_i}$. θ^p and ϕ^z are assumed to be Dirichlet distributions with parameters α and β.

The set of context words (X and Y fillers) occurring with a pattern p form the *document* (in LDA terms) associated to a pattern p. Table 2 lists a fragment of the document associated to pattern *X solve Y*. These are built simply by listing for each pattern, occurrence counts with specific filler words. Since we want our model to differentiate between X and Y fillers, words occurring as fillers are made disjoint by adding a corresponding suffix.

The total set of such *documents* extracted from a large corpus is then used for estimating the model. We use Gibbs sampling[2] and the result is a set of samples from $P(z|w)$ (i.e. meaning assignments for each occurring filler word) from which θ^p (pattern-meaning distributions) and ϕ^z(meaning-word distributions) can be estimated.

Our model has the advantage that, once these

[2] *http://gibbslda.sourceforge.net/*

distributions are estimated, given a pattern p and a context w_n, *in-context* vector representations can be built in a straightforward manner.

Meaning representation in-context Let K be the assumed number of meanings, $(z_1, ..., z_K)$. We associate to a pattern in context (p,w_n), the ***K***-dimensional vector containing for each meaning z_i ($i : 1..K$), the probability of z_i, conditioned on pattern p and context word w_n:

$$vec(p, w_n) = (P(z_1|w_n, p), ..., P(z_K|w_n, p)) \tag{1}$$

where,

$$P(z_i|w_n, p) = \frac{P(z_i, p)P(w_n|z_i)}{\Sigma_{i=1}^{K} P(z_i, p)P(w_n|z_i)} \tag{2}$$

This is the probability that w_n is generated by meaning z_i conditioned on p, therefore, the probability that pattern p has meaning z_i in context w_n, exactly the concept we want to model.

Meaning representation out-of-context We can also associate to pattern p an *out-of-context* vector representation: the ***K***-dimensional vector representing its distribution over meanings:

$$vec(p) = (P(z_1|p), ..., P(z_K|p)) \tag{3}$$

This can be seen as a dimensionality reduction method, since we bring vector representations to a lower dimensional space over (ideally) meaningful concepts.

From the generative model we obtain the desired distributions $P(z_i|p) = \theta_i^p$ and $P(w_n|z_i) = \phi_n^{z_i}$.[3]

Computing similarity between patterns The similarity between patterns occurring with X and Y filler-words is computed following (Lin and Pantel, 2001a) by multiplying the similarities obtained separately in the X and Y spaces.:

$$\begin{aligned} sim((w_{X1}, p_1, w_{Y1})(w_{X2}, p_2, w_{Y2})) = \\ sim(vec(p_1, w_{X1}), vec(p_2, w_{X2}))* \\ sim(vec(p_1, w_{Y1}), vec(p_2, w_{Y2})) \end{aligned} \tag{4}$$

$we \xleftarrow{subj} make \xrightarrow{dobj} statement$	
$we \xleftarrow{subj} give \xrightarrow{dobj} statement$	*good*
$we \xleftarrow{subj} prepare \xrightarrow{dobj} statement$	*bad*

Table 3: Development set: good/bad substitutes for $we \xleftarrow{subj} make \xrightarrow{dobj} statement$

Out-of-context similarity is defined in a straightforward manner:

$$sim(p_1, p_2) = sim(vec(p_1,), vec(p_2)) \tag{5}$$

4 Evaluation setup

In this paper we evaluate our model on computing similarities between pairs of the type $(X, pattern, Y), (X, pattern', Y)$ where two different patterns are compared in identical contexts. For this we use the Semeval Lexical Substitution dataset, which requires human participants to provide substitutes for a set of target words occurring in different contexts. This section describes the evaluation methodology for this data as well as the automatically generated data set we use for development.

Development set For finding good model parameters, we use the SemCor corpus providing text in which all content words are tagged with WordNet 1.6 senses. We used this data in the following manner: We parse the text using Stanford parser and extract occurrences of triples *(X, pattern, Y)*. Given these triples we generate *good* and *bad* substitutes for them: the *good* substitutes are generated by replacing the words occurring in the patterns with sense-appropriate synonyms, while *bad* ones are obtained by substitution with synonyms corresponding to the rest of the senses (the wrong senses). The synonyms are extracted from WordNet 1.6 synsets using the sense annotation present in the text.

For evaluation we feed the models pairs of instantiated patterns. One of them is the original phrase encountered in the data, and the other one is a *good/bad* substitute for it. Table 3 shows an example of the data.

We evaluate the output of a system by requiring that, for each instance, every good substitute is scored more similar to the original phrase than

[3] For similarity in context, we use the conditional $P(z_i|p)$ instead of the joint $P(z_i, p)$ which is computationally equivalent for the paraphrasing setting.

every bad substitute. This leads to an accuracy score which can be compared against a random baseline of 50%.

The data set obtained is far from being a very reliable resource for the task of lexical substitution, however this method of generating data has the advantage of producing a large number of instances which can be easily acquired from any sense-annotated data set. In our experiments we use the Brown2 fragment from which we extract over 3000 instances of patterns in context.

Lexical substitution task The Lexical Substitution Task (McCarthy and Navigli, 2007) presents 5 annotators with a set of target words, each in different context sentences. The task requires the participants to provide appropriate substitute words for each occurrence of the target words.

We use this data similarly to (Erk and Padó, 2008) and (Thater et al., 2009) and for each target word, we pool together all the substitutes given for *all* context sentences. Similarly to the SemCor data, we do not use the entire sentence as a context as we extract only *patterns* containing target words together with their X and Y fillers. The models assign similarity scores to each candidate by comparing them to the pattern occurring in the original sentence. A ranked list of candidates is obtained which in turn is compared with the substitutes provided by the participants. Table 4 gives an example of this data set (for each substitute we list the number of participants providing it).

To evaluate the performance of a model we employ two similarity measures, which capture different aspects of the task. Kendall τ rank coefficient measures the correlation between two ranks; since the gold ranking is usually only a partial order, we use τ_b which makes adjustments for ties. We employ a second evaluation measure: Generalized Average Precision (Kishida, 2005). This is a measure inspired from information retrieval and has been previously used for evaluating this task (Thater et al., 2009). It evaluates a system on its ability to retrieve correct substitutes using the gold ranking together with the associated confidence scores. The confidence scores are in turn determined by the number of people providing each substitute.

pattern	*human substitutes*
$study \xleftarrow{subj} shed \xrightarrow{dobj} light$	throw 3, reveal 2, shine 1
$cat \xleftarrow{subj} shed \xrightarrow{dobj} virus$	spread 2, pass 2, transmit 2, emit 1

Table 4: Lexical substitution data set: target verb *shed*

5 Experiments

5.1 Model selection

The data we use to estimate our models is extracted from a GigaWord fragment containing approximately 100 million tokens. We parse the text with Stanford dependency parser to obtain dependency graphs from which we extract paths together with counts of their left and right fillers. We extract paths containing at most four words, including the two noun anchors. Furthermore we impose a frequency threshold on patterns and words, leading us to a collection of ≈80 000 paths, with filler nouns over a vocabulary of ≈40 000 words.

We estimate a total number of 20 models. We set $\beta = 0.01$ as previous work (Wang et al., 2009) reports good results with this value. For parameter α we test 4 settings: $\alpha_1 = \frac{2}{K}$ and $\alpha_4 = \frac{50}{K}$ which are reported in the literature as good ((Porteous et al., 2008) and (Griffiths and Steyvers, 2004)), as well as 2 intermediate values: $\alpha_2 = \frac{5}{K}$ and $\alpha_3 = \frac{10}{K}$. We test a set of 5 K values: $\{800, 1000, 1200, 1400, 1600\}$. These are chosen to be large since they represent the global set of meanings shared by all the patterns in the collection.

As vector similarity measure we test scalar product (sp), which in our model is interpreted as the probability that two patterns share a common meaning. Additionally we test cosine (cos) similarity and inverse Jensen-Shannon (JS) divergence, which is a popular measure for comparing probability distributions:

$$JSD(v, w) = \frac{1}{2}KLD(v|m) + \frac{1}{2}KLD(w|m)$$

with $m = \frac{1}{2}(v + w)$ and KLD the standard Kullback-Leibler divergence: $KLD(v|w) = \Sigma_i v_i log(\frac{v_i}{w_i})$.

We perform both in-context (using eq. (4)) as well as out-of-context computations (eq. (5)). Similarly to previous work (Erk and Padó, 2008), we observe that comparing a contextualized representation against a non-contextualized one brings significant improvements over comparing two representations in context. We assume this is specific to the type of data we work with, in which two patterns are compared in an identical context, rather than across different contexts; we therefore compute context-sensitive similarities by contextualizing just the target word.

Number of topics Although the parameters cover relatively large ranges the models perform surprisingly similar across different α and K values, as well as across all three similarity measures. For sp similarity, the accuracy scores we obtain are in the range [56.5-59.5] with a average deviation from the mean of just 0.8%; similar figures are obtained using the other similarity measures. Figure 1 plots the average of the accuracy scores using sp as similarity measure, across different number of topics. A small preference for higher K values is observed, all models performing consistently good at 1200, 1400 and 1600 topics.

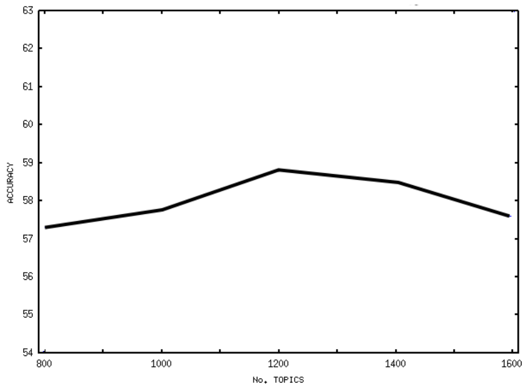

Figure 1: Average accuracy across the 5 K values.

Mixture models This leads us to attempting a very simple mixture model, which computes the similarity score between two patterns as the average similarity obtained across a number of models. For each α setting, we mix models across the three best topic numbers: $\{1200, 1400, 1600\}$. In Figure 2 we plot this mixture model together with the three single ones, at each α value. It can be noticed that the mixture model improves over all three single models for three out of the four α values.

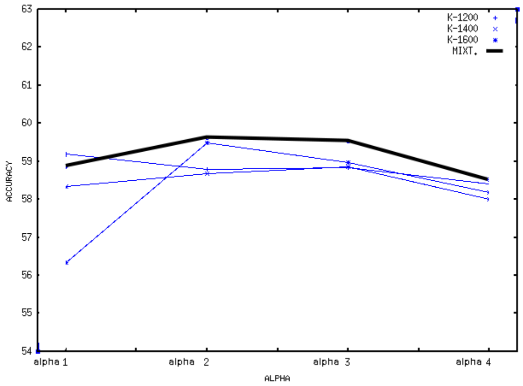

Figure 2: Mixture model $\{1200, 1400, 1600\}$ (bold) vs. the three individual models, across the 4 α values.

In-context vs. out-of-context computations Further on we compare *in-context* versus *out-of-context* computations. The similarity measures exhibit significant differences in regard to this aspect. In Figure 3 we plot *in-context* vs. *out-of-context* computations using scalar product (left) and JS (right) with the mixture model previously defined, plotted at different α values. For *sp* *in-context* computations significantly outperform *out-of-context* ones and the two intermediate alpha values seem to be the best. However for JS similarity the *out-of-context* computations are significantly better and a clear preference for smaller α values can be observed.

Finally, on the test data, we use the following models (where $GM_{mixt/sing,sim}$ stands for a *mixture* or *single* model with similarity measure *sim*):

- $GM_{mixt,sp/cos}$
  mixt({1200, 1400, 1600}x{α_2, α_3})
- $GM_{mixt,js}$
  mixt({1200, 1400, 1600}x{α_1, α_2})
- $GM_{sing,sp}$: (1600, α2)
- $GM_{sing,cos/js}$: (1200, α1)

The mixture models are build based on the observations previously made while the single mod-

Model	*In-context*	*Out-of-context*
$GM_{mixt,sp}$	59.89	58.68
$GM_{mixt,cos}$	59.50	58.67
$GM_{mixt,js}$	59.73	60.68
$GM_{sing,sp}$	59.48	58.86
$GM_{sing,cos}$	59.43	57.87
$GM_{sing,js}$	58.65	59.36

Table 5: Accuracy results on development set

els are the best performing ones, for each similarity measure. The accuracy scores obtained with these models are given in Table 5. Mixture models generally outperform single ones and *in-context* computations outperform *out-of-context* ones for sp and cos. The best results on the development set are however achieved by *out-of-context* models using JS as similarity measure.

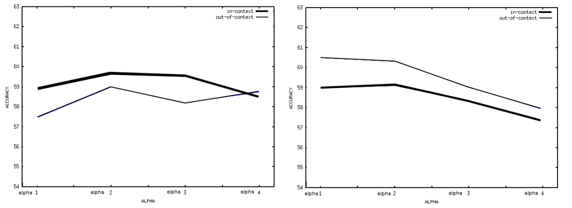

Figure 3: *In-context* (bold) vs. *out-of-context* computations across the 4 α values using scalar product (left) and JS (right)

5.2 Results

Table 6 shows the results for the Lexical Substitution data set. We use the subset of the data containing sentences in which the target word is part of a syntactic path which is present in the total collection of patterns. This leads to a set containing 165 instances of patterns in context, most of these containing target verbs.

Since sp and cos measures perform very similarly we only list results with cosine similarity measure. In addition to the models with settings determined on the development set, we also test a very simple mixture model: $GM_{mixt-all,sim}$. This simply averages over *all* 20 configurations and its purpose is to investigate the necessity of a carefully selected mixture model.

It can be noticed that all GM mixture models outperform DIRT, which is reflected in both similarity measures. Notably the very simple model which averages all the configurations implemented is surprisingly performant. Using randomized significance testing we obtained that $GM_{mixt,cos}$ is significantly better than DIRT at p level 1e-03 on both GAP and τ_b. $GM_{mixt-all,cos}$ outperforms DIRT at level 0.05.

Model	τ_b	GAP
Random	0.0	34.91
DIRT	14.53	48.06
$\mathbf{GM_{mixt,cos}}$	**22.35**	**52.04**
$GM_{mixt,js}$	18.17	50.80
$\mathbf{GM_{mixt-all,cos}}$	**20.42**	**51.13**
$GM_{mixt-all,js}$	19.03	51.15
$GM_{sing,cos}$	15.10	48.20
$GM_{sing,js}$	14.17	47.97

Table 6: Results on Lexical Substitution data

In terms of similarity measures, the observations made on the development set hold, as for the *in-context* computations cos and sp outperform JS. However, unlike on the development data, the single models perform much worse than the mixture ones which can indicate that the development set is not perfectly suited for choosing model parameters.

Out-of-context computations for all models and all similarity measures are significantly outperformed, leading to scores in ranges [11-14] τ_b and [45-48] GAP.

In Table 7 we list the rankings produced by three models for the target word $shed$ in context $virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$. As it can be observed, the model performing context-sensitive computations $GM_{mixt,cos}$-*in-context* returns a better ranking in comparison to the $DIRT$ and $GM_{mixt,cos}$-*out-of-context* models.

6 Conclusion

We have addressed the task of computing meaning similarity in context using distributional methods. The specific representation we use follows (Lin and Pantel, 2001a): we extract *patterns* (paths in dependency trees which connect two nouns) and we use the co-occurrence with these nouns to build high-dimensional vectors. Using this data

$virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$

$GM_{mixt,cos}$	$GM_{mixt,cos}$	*DIRT*	GOLD	
in-context	*out-of-context*			
lose	*lose*	*drop*	pass	2
drop	*drop*	*lose*	spread	2
transmit	*relinquish*	*give*	transmit	2
spread	*reveal*	**transmit**		
pass	**pass**	**spread**		
relinquish	*throw*	*reveal*		
reveal	**spread**	*relinquish*		
throw	**transmit**	*throw*		
give	*give*	**pass**		

Table 7: Ranks returned for $virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$

we develop a principled method to induce context-sensitive representations by modeling the *meaning* of a pattern as a latent variable in the input corpus. We apply this model to the task of Lexical Substitution and we show it allows the computation of context-sensitive similarities; it significantly outperforms the original method, while using the exact same input data.

In future work, we plan to use our model for generating paraphrases for patterns occurring in context, a scenario closer to real applications than *out-of-context* paraphrasing.

Finally, a formulation of our model in a typical bag-of-words semantic space for word similarity can be employed in a wider range of applications and will allow comparison with other methods for building context-sensitive vector representations.

7 Acknowledgments

This work was partially supported by DFG (IRTG 715).

References

Basili, Roberto, Diego De Cao, Paolo Marocco, and Marco Pennacchiotti. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *In Proceedings of RANLP 2007*, Borovets, Bulgaria.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Boyd-Graber, Jordan, David M. Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *Empirical Methods in Natural Language Processing*.

Brody, Samuel and Mirella Lapata. 2009. Bayesian word sense induction. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111, Morristown, NJ, USA. Association for Computational Linguistics.

Cai, Jun Fu, Wee Sun Lee, and Yee Whye Teh. 2007. Nus-ml:improving word sense disambiguation using topic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 249–252, Prague, Czech Republic, June. Association for Computational Linguistics.

Connor, Michael and Dan Roth. 2007. Context sensitive paraphrasing with a global unsupervised classifier. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 104–115, Berlin, Heidelberg. Springer-Verlag.

Erk, Katrin and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP 2008*, Waikiki, Honolulu, Hawaii.

Griffiths, T. L. and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April.

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

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Lin, Dekang and Patrick Pantel. 2001a. DIRT – Discovery of Inference Rules from Text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA.

Lin, Dekang and Patrick Pantel. 2001b. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360.

McCarthy, D. and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, pages 48–53, Prague.

Mitchell, Jeff and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio.

Pantel, Patrick, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York.

Porteous, Ian, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, New York, NY, USA. ACM.

Schuetze, Hinrich. 1998. Automatic word sense discrimination. *Journal of Computational Linguistics*, 24:97–123.

Szpektor, Idan, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. 2008. Contextual preferences. In *Proceedings of ACL-08: HLT*, pages 683–691, Columbus, Ohio, June. Association for Computational Linguistics.

Thater, Stefan, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of TextInfer ACL 2009*.

Toutanova, Kristina and Mark Johnson. 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging. In Platt, J.C., D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528. MIT Press, Cambridge, MA.

Wang, Yi, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. 2009. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Proc. of 5th International Conference on Algorithmic Aspects in Information and Management.*