Multi-Prototype Vector-Space Models of Word Meaning

Joseph Reisinger
Department of Computer Science
The University of Texas at Austin
1 University Station C0500
Austin, TX 78712-0233
joeraii@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
The University of Texas at Austin
1 University Station C0500
Austin, TX 78712-0233
mooney@cs.utexas.edu

Abstract

Current vector-space models of lexical semantics create a single "prototype" vector to represent the meaning of a word. However, due to lexical ambiguity, encoding word meaning with a single vector is problematic. This paper presents a method that uses clustering to produce multiple "sense-specific" vectors for each word. This approach provides a context-dependent vector representation of word meaning that naturally accommodates homonymy and polysemy. Experimental comparisons to human judgements of semantic similarity for both isolated words as well as words in sentential contexts demonstrate the superiority of this approach over both prototype and exemplar based vector-space models.

1 Introduction

Automatically judging the degree of semantic similarity between words is an important task useful in text classification (Baker and McCallum, 1998), information retrieval (Sanderson, 1994), textual entailment, and other language processing tasks. The standard empirical approach to this task exploits the *distributional hypothesis*, i.e. that similar words appear in similar contexts (Curran and Moens, 2002; Lin and Pantel, 2002; Pereira et al., 1993). Traditionally, word types are represented by a single vector of contextual features derived from co-occurrence information, and semantic similarity is computed using some measure of vector distance (Lee, 1999; Lowe, 2001).

However, due to homonymy and polysemy, capturing the semantics of a word with a single vector is problematic. For example, the word *club* is similar to both *bat* and *association,* which are not at all similar to each other. Word meaning violates the triangle inequality when viewed at the level of word types, posing a problem for vector-space models (Tversky and Gati, 1982). A single "prototype" vector is simply incapable of capturing phenomena such as homonymy and polysemy. Also, most vector-space models are context independent, while the meaning of a word clearly depends on context. The word *club* in "The caveman picked up the *club*" is similar to *bat* in "John hit the robber with a *bat*," but not in "The *bat* flew out of the cave."

We present a new resource-lean vector-space model that represents a word's meaning by a *set* of distinct "sense specific" vectors. The similarity of two isolated words A and B is defined as the *minimum* distance between one of A's vectors and one of B's vectors. In addition, a context-dependent meaning for a word is determined by choosing one of the vectors in its set based on minimizing the distance to the vector representing the current context. Consequently, the model supports judging the similarity of both words in isolation and words in context.

The set of vectors for a word is determined by unsupervised *word sense discovery* (WSD) (Schütze, 1998), which clusters the contexts in which a word appears. In previous work, vector-space lexical similarity and word sense discovery have been treated as two separate tasks. This paper shows how they can be combined to create an improved vector-space model of lexical semantics. First, a word's contexts are clustered to produce groups of similar context vectors. An average "prototype" vector is then computed separately for each cluster, producing a set of vectors for each word. Finally, as described above, these cluster vectors can be used to determine the se-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 109–117,
Los Angeles, California, June 2010.

mantic similarity of both isolated words and words in context. The approach is completely modular, and can integrate any clustering method with any traditional vector-space model.

We present experimental comparisons to human judgements of semantic similarity for both isolated words and words in sentential context. The results demonstrate the superiority of a clustered approach over both traditional prototype and exemplar-based vector-space models. For example, given the isolated target word *singer* our method produces the most similar word *vocalist*, while using a single prototype gives *musician*. Given the word *cell* in the context: "The book was published while Piasecki was still in prison, and a copy was delivered to his *cell*." the standard approach produces *protein* while our method yields *incarcerated*.

The remainder of the paper is organized as follows: Section 2 gives relevant background on prototype and exemplar methods for lexical semantics, Section 3 presents our multi-prototype method, Section 4 presents our experimental evaluations, Section 5 discusses future work, and Section 6 concludes.

2 Background

Psychological concept models can be roughly divided into two classes:

1. *Prototype* models represented concepts by an abstract prototypical instance, similar to a cluster centroid in parametric density estimation.

2. *Exemplar* models represent concepts by a concrete set of observed instances, similar to non-parametric approaches to density estimation in statistics (Ashby and Alfonso-Reese, 1995).

Tversky and Gati (1982) famously showed that conceptual similarity violates the triangle inequality, lending evidence for exemplar-based models in psychology. Exemplar models have been previously used for lexical semantics problems such as selectional preference (Erk, 2007) and thematic fit (Vandekerckhove et al., 2009). Individual exemplars can be quite noisy and the model can incur high computational overhead at prediction time since naively computing the similarity between two words using each occurrence in a textual corpus as an exemplar requires $O(n^2)$ comparisons. Instead, the standard approach is to compute a single prototype vector for each word from its occurrences.

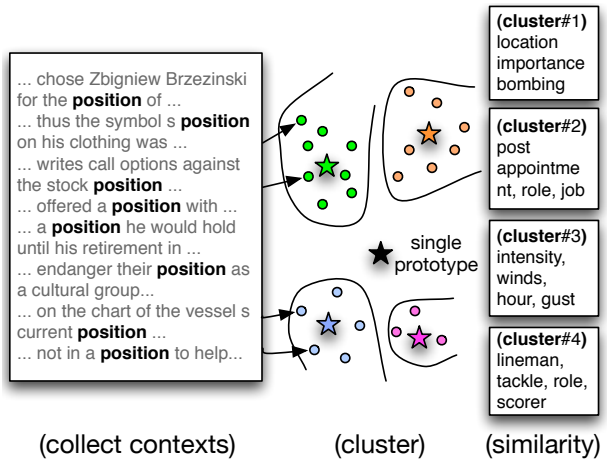

Figure 1: Overview of the multi-prototype approach to near-synonym discovery for a single target word independent of context. Occurrences are clustered and cluster centroids are used as prototype vectors. Note the "hurricane" sense of *position* (cluster 3) is not typically considered appropriate in WSD.

This paper presents a *multi-prototype* vector space model for lexical semantics with a single parameter K (the number of clusters) that generalizes both prototype ($K = 1$) and exemplar ($K = N$, the total number of instances) methods. Such models have been widely studied in the Psychology literature (Griffiths et al., 2007; Love et al., 2004; Rosseel, 2002). By employing multiple prototypes per word, vector space models can account for homonymy, polysemy and thematic variation in word usage. Furthermore, such approaches require only $O(K^2)$ comparisons for computing similarity, yielding potential computational savings over the exemplar approach when $K \ll N$, while reaping many of the same benefits.

Previous work on lexical semantic relatedness has focused on two approaches: (1) mining monolingual or bilingual dictionaries or other pre-existing resources to construct networks of related words (Agirre and Edmond, 2006; Ramage et al., 2009), and (2) using the distributional hypothesis to automatically infer a vector-space prototype of word meaning from large corpora (Agirre et al., 2009; Curran, 2004; Harris, 1954). The former approach tends to have greater precision, but depends on hand-

crafted dictionaries and cannot, in general, model sense frequency (Budanitsky and Hirst, 2006). The latter approach is fundamentally more scalable as it does not rely on specific resources and can model corpus-specific sense distributions. However, the distributional approach can suffer from poor precision, as thematically similar words (e.g., *singer* and *actor*) and antonyms often occur in similar contexts (Lin et al., 2003).

Unsupervised word-sense discovery has been studied by number of researchers (Agirre and Edmond, 2006; Schütze, 1998). Most work has also focused on corpus-based distributional approaches, varying the vector-space representation, e.g. by incorporating syntactic and co-occurrence information from the words surrounding the target term (Pereira et al., 1993; Pantel and Lin, 2002).

3 Multi-Prototype Vector-Space Models

Our approach is similar to standard vector-space models of word meaning, with the addition of a per-word-type clustering step: Occurrences for a specific word type are collected from the corpus and clustered using any appropriate method (§3.1). Similarity between two word types is then computed as a function of their cluster centroids (§3.2), instead of the centroid of all the word's occurrences. Figure 1 gives an overview of this process.

3.1 Clustering Occurrences

Multiple prototypes for each word w are generated by clustering feature vectors $v(c)$ derived from each occurrence $c \in \mathcal{C}(w)$ in a large textual corpus and collecting the resulting cluster centroids $\pi_k(w), k \in [1, K]$. This approach is commonly employed in unsupervised word sense discovery; however, we do not assume that clusters correspond to traditional word senses. Rather, we only rely on clusters to capture meaningful variation in word usage.

Our experiments employ a *mixture of von Mises-Fisher distributions* (movMF) clustering method with first-order unigram contexts (Banerjee et al., 2005). Feature vectors $v(c)$ are composed of individual features $I(c, f)$, taken as all unigrams occurring $f \in \mathcal{F}$ in a 10-word window around w.

Like spherical k-means (Dhillon and Modha, 2001), movMF models semantic relatedness using cosine similarity, a standard measure of textual similarity. However, movMF introduces an additional per-cluster *concentration* parameter controlling its semantic breadth, allowing it to more accurately model non-uniformities in the distribution of cluster sizes. Based on preliminary experiments comparing various clustering methods, we found movMF gave the best results.

3.2 Measuring Semantic Similarity

The similarity between two words in a multi-prototype model can be computed straightforwardly, requiring only simple modifications to standard distributional similarity methods such as those presented by Curran (2004). Given words w and w', we define two *noncontextual clustered similarity metrics* to measure similarity of isolated words:

$$\text{AvgSim}(w, w') \stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} d(\pi_k(w), \pi_j(w'))$$

$$\text{MaxSim}(w, w') \stackrel{\text{def}}{=} \max_{1 \le j \le K, 1 \le k \le K} d(\pi_k(w), \pi_j(w'))$$

where $d(\cdot, \cdot)$ is a standard distributional similarity measure. In AvgSim, word similarity is computed as the average similarity of all pairs of prototype vectors; In MaxSim the similarity is the maximum over all pairwise prototype similarities. All results reported in this paper use *cosine* similarity, [1]

$$\text{Cos}(w, w') = \frac{\sum_{f \in \mathcal{F}} I(w, f) \cdot I(w', f)}{\sqrt{\sum_{f \in \mathcal{F}} I(w, f)^2} \sqrt{\sum_{f \in \mathcal{F}} I(w', f)^2}}$$

We compare across two different feature functions *tf-idf* weighting and χ^2 weighting, chosen due to their ubiquity in the literature (Agirre et al., 2009; Curran, 2004).

In AvgSim, all prototype pairs contribute equally to the similarity computation, thus two words are judged as similar if many of their senses are similar. MaxSim, on the other hand, only requires a single pair of prototypes to be close for the words to be judged similar. Thus, MaxSim models the similarity of words that share only a single sense (e.g. *bat* and *club*) at the cost of lower robustness to noisy clusters that might be introduced when K is large.

When contextual information is available, AvgSim and MaxSim can be modified to produce

[1] The main results also hold for *weighted Jaccard* similarity.

more precise similarity computations:

$$\text{AvgSimC}(w, w') \stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} d_{c,w,k} d_{c',w',j} d(\pi_k(w), \pi_j(w'))$$

$$\text{MaxSimC}(w, w') \stackrel{\text{def}}{=} d(\hat{\pi}(w), \hat{\pi}(w'))$$

where $d_{c,w,k} \stackrel{\text{def}}{=} d(v(c), \pi_k(w))$ is the likelihood of context c belonging to cluster $\pi_k(w)$, and $\hat{\pi}(w) \stackrel{\text{def}}{=} \pi_{\arg\max_{1 \le k \le K} d_{c,w,k}}(w)$, the maximum likelihood cluster for w in context c. Thus, AvgSimC corresponds to *soft cluster assignment*, weighting each similarity term in AvgSim by the likelihood of the word contexts appearing in their respective clusters. MaxSimC corresponds to *hard assignment*, using only the most probable cluster assignment. Note that AvgSim and MaxSim can be thought of as special cases of AvgSimC and MaxSimC with uniform weight to each cluster; hence AvgSimC and MaxSimC can be used to compare words in context to isolated words as well.

4 Experimental Evaluation

4.1 Corpora

We employed two corpora to train our models:

1. A snapshot of English Wikipedia taken on Sept. 29th, 2009. Wikitext markup is removed, as are articles with fewer than 100 words, leaving 2.8M articles with a total of 2.05B words.

2. The third edition English Gigaword corpus, with articles containing fewer than 100 words removed, leaving 6.6M articles and 3.9B words (Graff, 2003).

Wikipedia covers a wider range of sense distributions, whereas Gigaword contains only newswire text and tends to employ fewer senses of most ambiguous words. Our method outperforms baseline methods even on Gigaword, indicating its advantages even when the corpus covers few senses.

4.2 Judging Semantic Similarity

To evaluate the quality of various models, we first compared their lexical similarity measurements to human similarity judgements from the WordSim-353 data set (Finkelstein et al., 2001). This test corpus contains multiple human judgements on 353 word pairs, covering both monosemous and polysemous words, each rated on a 1–10 integer scale. Spearman's rank correlation (ρ) with average human judgements (Agirre et al., 2009) was used to measure the quality of various models.

Figure 2 plots Spearman's ρ on WordSim-353 against the number of clusters (K) for Wikipedia and Gigaword corpora, using pruned *tf-idf* and χ^2 features.[2] In general pruned *tf-idf* features yield higher correlation than χ^2 features. Using AvgSim, the multi-prototype approach ($K > 1$) yields higher correlation than the single-prototype approach ($K = 1$) across all corpora and feature types, achieving state-of-the-art results with pruned *tf-idf* features. This result is statistically significant in all cases for *tf-idf* and for $K \in [2, 10]$ on Wikipedia and $K > 4$ on Gigaword for χ^2 features.[3] MaxSim yields similar performance when $K < 10$ but performance degrades as K increases.

It is possible to circumvent the model-selection problem (choosing the best value of K) by simply combining the prototypes from clusterings of different sizes. This approach represents words using both semantically broad and semantically tight prototypes, similar to hierarchical clustering. Table 1 and Figure 2 (squares) show the result of such a *combined* approach, where the prototypes for clusterings of size 2-5, 10, 20, 50, and 100 are unioned to form a single large prototype set. In general, this approach works about as well as picking the optimal value of K, even outperforming the single best cluster size for Wikipedia.

Finally, we also compared our method to a pure exemplar approach, averaging similarity across all occurrence pairs.[4] Table 1 summarizes the results. The exemplar approach yields significantly higher correlation than the single prototype approach in all cases except Gigaword with *tf-idf* features ($p < 0.05$). Furthermore, it performs significantly *worse*

[2] (**Feature pruning**) We find that results using *tf-idf* features are extremely sensitive to feature pruning while χ^2 features are more robust. In all experiments we prune *tf-idf* features by their overall weight, taking the top 5000. This setting was found to optimize the performance of the single-prototype approach.

[3] Significance is calculated using the large-sample approximation of the Spearman rank *test*; ($p < 0.05$).

[4] Averaging across all pairs was found to yield higher correlation than averaging over the most similar pairs.

Spearman's ρ	prototype	exemplar	multi-prototype (AvgSim)			combined
			$K = 5$	$K = 20$	$K = 50$	
Wikipedia *tf-idf*	0.53±0.02	0.60±0.06	0.69±0.02	0.76±0.01	0.76±0.01	0.77±0.01
Wikipedia χ^2	0.54±0.03	0.65±0.07	0.58±0.02	0.56±0.02	0.52±0.03	0.59±0.04
Gigaword *tf-idf*	0.49±0.02	0.48±0.10	0.64±0.02	0.61±0.02	0.61±0.02	0.62±0.02
Gigaword χ^2	0.25±0.03	0.41±0.14	0.32±0.03	0.35±0.03	0.33±0.03	0.34±0.03

Table 1: Spearman correlation on the WordSim-353 dataset broken down by corpus and feature type.

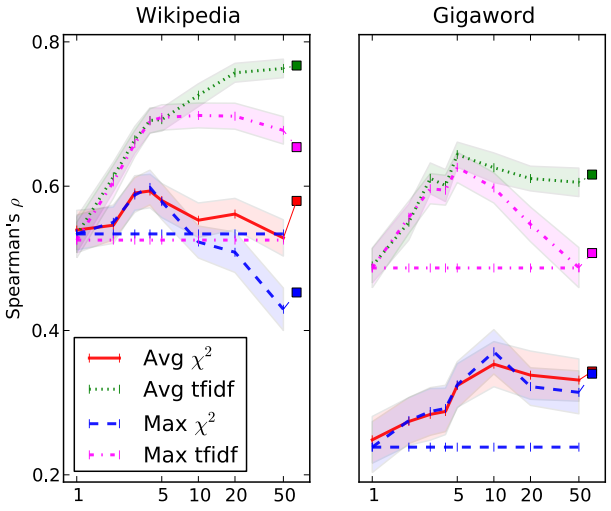

Figure 2: WordSim-353 rank correlation vs. number of clusters (log scale) for both the Wikipedia (left) and Gigaword (right) corpora. Horizontal bars show the performance of single-prototype. Squares indicate performance when combining across clusterings. Error bars depict 95% confidence intervals using the Spearman test. Squares indicate performance when combining across clusterings.

than combined multi-prototype for *tf-idf* features, and does not differ significantly for χ^2 features. Overall this result indicates that multi-prototype performs at least as well as exemplar in the worst case, and significantly outperforms when using the best feature representation / corpus pair.

4.3 Predicting Near-Synonyms

We next evaluated the multi-prototype approach on its ability to determine the most closely related words for a given target word (using the Wikipedia corpus with *tf-idf* features). The top k most similar words were computed for each prototype of each target word. Using a forced-choice setup, human subjects were asked to evaluate the quality of these *near synonyms* relative to those produced by a single prototype. Participants on Amazon's Mechanical Turk[5] (Snow et al., 2008) were asked to choose between two possible alternatives (one from a prototype model and one from a multi-prototype model) as being most similar to a given target word. The target words were presented either in isolation or in a sentential context randomly selected from the corpus. Table 2 lists the ambiguous words used for this task. They are grouped into homonyms (words with very distinct senses) and polysemes (words with related senses). All words were chosen such that their usages occur within the same part of speech.

homonymous
carrier, crane, cell, company, issue, interest, match, media, nature, party, practice, plant, racket, recess, reservation, rock, space, value
polysemous
cause, chance, journal, market, network, policy, power, production, series, trading, train

Table 2: Words used in predicting near synonyms.

In the non-contextual task, 79 unique raters completed 7,620 comparisons of which 72 were discarded due to poor performance on a known test set.[6] In the contextual task, 127 raters completed 9,930 comparisons of which 87 were discarded.

For the non-contextual case, Figure 3 left plots the fraction of raters preferring the multi-prototype prediction (using AvgSim) over that of a single prototype as the number of clusters is varied. When asked to choose between the single best word for

[5] http://mturk.com

[6] (**Rater reliability**) The reliability of Mechanical Turk raters is quite variable, so we computed an accuracy score for each rater by including a control question with a known correct answer in each HIT. Control questions were generated by selecting a random word from WordNet 3.0 and including as possible choices a word in the same synset (correct answer) and a word in a synset with a high path distance (incorrect answer). Raters who got less than 50% of these control questions correct, or spent too little time on the HIT were discarded.

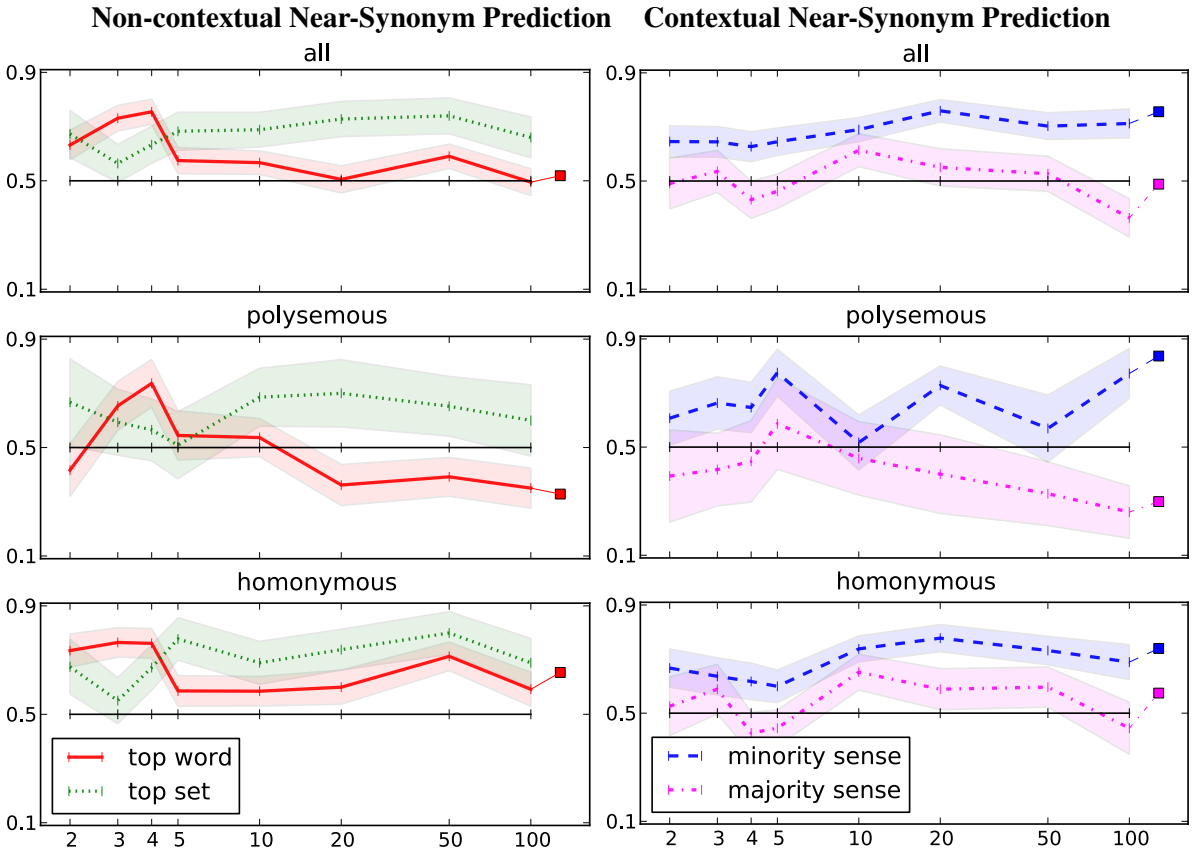

Figure 3: (**left**) Near-synonym evaluation for isolated words showing fraction of raters preferring multi-prototype results vs. number of clusters. Colored squares indicate performance when combining across clusterings. 95% confidence intervals computed using the Wald test. (**right**) Near-synonym evaluation for words in a sentential context chosen either from the minority sense or the majority sense.

each method (**top word**), the multi-prototype prediction is chosen significantly more frequently (i.e. the result is above 0.5) when the number of clusters is small, but the two methods perform similarly for larger numbers of clusters (Wald test, $\alpha = 0.05$.) Clustering more accurately identifies homonyms' clearly distinct senses and produces prototypes that better capture the different uses of these words. As a result, compared to using a single prototype, our approach produces better near-synonyms for homonyms compared to polysemes. However, given the right number of clusters, it also produces better results for polysemous words.

The near-synonym prediction task highlights one of the weaknesses of the multi-prototype approach: as the number of clusters increases, the number of occurrences assigned to each cluster decreases, increasing noise and resulting in some poor prototypes that mainly cover outliers. The word similarity task is somewhat robust to this phenomenon, but synonym prediction is more affected since only the top predicted choice is used. When raters are forced to chose between the top *three* predictions for each method (presented as **top set** in Figure 3 left), the effect of this noise is reduced and the multi-prototype approach remains dominant even for a large number of clusters. This indicates that although more clusters can capture finer-grained sense distinctions, they also can introduce noise.

When presented with words in context (Figure 3 right),[7] raters found no significant difference in the two methods for words used in their majority sense.[8] However, when a minority sense is pre-

[7]Results for the multi-prototype method are generated using AvgSimC (soft assignment) as this was found to significantly outperform MaxSimC.

[8]Sense frequency determined using Google; senses labeled manually by trained human evaluators.

sented (e.g. the "prison" sense of *cell*), raters prefer the choice predicted by the multi-prototype approach. This result is to be expected since the single prototype mainly reflects the majority sense, preventing it from predicting appropriate synonyms for a minority sense. Also, once again, the performance of the multi-prototype approach is better for homonyms than polysemes.

4.4 Predicting Variation in Human Ratings

Variance in pairwise prototype distances can help explain the variance in human similarity judgements for a given word pair. We evaluate this hypothesis empirically on WordSim-353 by computing the Spearman correlation between the *variance* of the per-cluster similarity computations, $\mathbb{V}[D]$, $D \stackrel{\text{def}}{=} \{d(\pi_k(w), \pi_j(w')) : 1 \leq k, j \leq K\}$, and the variance of the human annotations for that pair. Correlations for each dataset are shown in Figure 4 left. In general, we find a statistically significant *negative* correlation between these values using χ^2 features, indicating that as the entropy of the pairwise cluster similarities increases (i.e., prototypes become more similar, and similarities become uniform), rater disagreement increases. This result is intuitive: if the occurrences of a particular word cannot be easily separated into coherent clusters (perhaps indicating high polysemy instead of homonymy), then human judgement will be naturally more difficult.

Rater variance depends more directly on the actual word similarity: word pairs at the extreme ranges of similarity have significantly lower variance as raters are more certain. By removing word pairs with similarity judgements in the middle two quartile ranges (4.4 to 7.5) we find significantly higher variance correlation (Figure 4 right). This result indicates that multi-prototype similarity variance accounts for a secondary effect separate from the primary effect that variance is naturally lower for ratings in extreme ranges.

Although the *entropy* of the prototypes correlates with the variance of the human ratings, we find that the individual senses captured by each prototype do not correspond to human intuition for a given word, e.g. the "hurricane" sense of *position* in Figure 1. This notion is evaluated empirically by computing the correlation between the predicted similarity us-

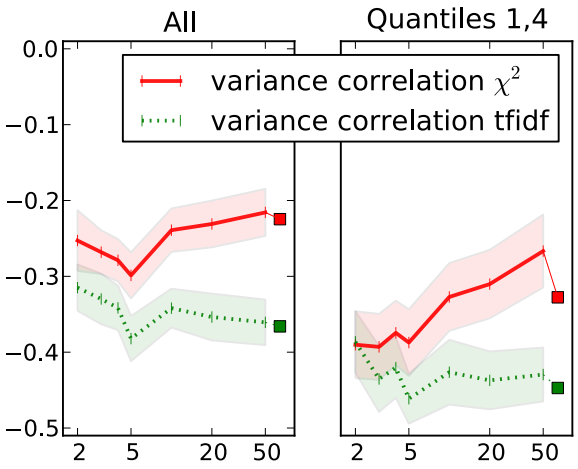

Figure 4: Plots of variance correlation; lower numbers indicate higher negative correlation, i.e. that prototype entropy predicts rater disagreement.

ing the contextual multi-prototype method and human similarity judgements for different usages of the *same* word. The Usage Similarity (USim) data set collected in Erk et al. (2009) provides such similarity scores from human raters. However, we find no evidence for correlation between USim scores and their corresponding prototype similarity scores ($\rho = 0.04$), indicating that prototype vectors may not correspond well to human senses.

5 Discussion and Future Work

Table 3 compares the inferred synonyms for several target words, generally demonstrating the ability of the multi-prototype model to improve the precision of inferred near-synonyms (e.g. in the case of *singer* or *need*) as well as its ability to include synonyms from less frequent senses (e.g., the *experiment* sense of *research* or the *verify* sense of *prove*). However, there are a number of ways it could be improved:

Feature representations: Multiple prototypes improve Spearman correlation on WordSim-353 compared to previous methods using the same underlying representation (Agirre et al., 2009). However we have not yet evaluated its performance when using more powerful feature representations such those based on Latent or Explicit Semantic Analysis (Deerwester et al., 1990; Gabrilovich and Markovitch, 2007). Due to its modularity, the multi-prototype approach can easily incorporate such advances in order to further improve its effectiveness.

Inferred Thesaurus	
bass	
single	guitar, drums, rhythm, piano, acoustic
multi	basses, contrabass, rhythm, guitar, drums
claim	
single	argue, say, believe, assert, contend
multi	assert, contend, allege, argue, insist
hold	
single	carry, take, receive, reach, maintain
multi	carry, maintain, receive, accept, reach
maintain	
single	ensure, establish, achieve, improve, promote
multi	preserve, ensure, establish, retain, restore
prove	
single	demonstrate, reveal, ensure, confirm, say
multi	demonstrate, verify, confirm, reveal, admit
research	
single	studies, work, study, training, development
multi	studies, experiments, study, investigations, training
singer	
single	musician, actress, actor, guitarist, composer
multi	vocalist, guitarist, musician, singer-songwriter, singers

Table 3: Examples of the top 5 inferred near-synonyms using the single- and multi-prototype approaches (with results merged). In general such clustering improves the precision and coverage of the inferred near-synonyms.

Nonparametric clustering: The success of the combined approach indicates that the optimal number of clusters may vary per word. A more principled approach to selecting the number of prototypes per word is to employ a clustering model with infinite capacity, e.g. the Dirichlet Process Mixture Model (Rasmussen, 2000). Such a model would allow naturally more polysemous words to adopt more flexible representations.

Cluster similarity metrics: Besides AvgSim and MaxSim, there are many similarity metrics over mixture models, e.g. KL-divergence, which may correlate better with human similarity judgements.

Comparing to traditional senses: Compared to WordNet, our best-performing clusterings are significantly more fine-grained. Furthermore, they often do not correspond to agreed upon semantic distinctions (e.g., the "hurricane" sense of *position* in Fig. 1). We posit that the finer-grained senses actually capture useful aspects of word meaning, leading to better correlation with WordSim-353. However, it would be good to compare prototypes learned from supervised sense inventories to prototypes produced by automatic clustering.

Joint model: The current method independently clusters the contexts of each word, so the senses discovered for w cannot influence the senses discovered for $w' \neq w$. Sharing statistical strength across similar words could yield better results for rarer words.

6 Conclusions

We presented a resource-light model for vector-space word meaning that represents words as collections of prototype vectors, naturally accounting for lexical ambiguity. The multi-prototype approach uses word sense discovery to partition a word's contexts and construct "sense specific" prototypes for each cluster. Doing so significantly increases the accuracy of lexical-similarity computation as demonstrated by improved correlation with human similarity judgements and generation of better near synonyms according to human evaluators. Furthermore, we show that, although performance is sensitive to the number of prototypes, combining prototypes across a large range of clusterings performs nearly as well as the ex-post best clustering. Finally, variance in the prototype similarities is found to correlate with inter-annotator disagreement, suggesting psychological plausibility.

Acknowledgements

We would like to thank Katrin Erk for helpful discussions and making the USim data set available. This work was supported by an NSF Graduate Research Fellowship and a Google Research Award. Experiments were run on the Mastodon Cluster, provided by NSF Grant EIA-0303609.

References

Eneko Agirre and Phillip Edmond. 2006. *Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proc. of NAACL-HLT-09*, pages 19–27.

F. Gregory Ashby and Leola A. Alfonso-Reese. 1995. Categorization as probability density estimation. *J. Math. Psychol.*, 39(2):216–233.

L. Douglas Baker and Andrew K. McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103.

Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 59–66.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh. College of Science.

Scott C. Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Inderjit S. Dhillon and Dharmendra S. Modha. 2001. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175.

Katrin Erk, Diana McCarthy, Nicholas Gaylord Investigations on Word Senses, and Word Usages. 2009. Investigations on word senses and word usages. In *Proc. of ACL-09*.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Association for Computer Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proc. of WWW-01*, pages 406–414, New York, NY, USA. ACM.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 1606–1611.

David Graff. 2003. *English Gigaword*. Linguistic Data Consortium, Philadephia.

Tom L. Griffiths, Kevin. R. Canini, Adam N. Sanborn, and Daniel. J. Navarro. 2007. Unifying rational models of categorization via the hierarchical Dirichlet process. In *Proc. of CogSci-07*.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Lillian Lee. 1999. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32.

Dekang Lin and Patrick Pantel. 2002. Concept discovery from text. In *Proc. of COLING-02*, pages 1–7.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the Interational Joint Conference on Artificial Intelligence*, pages 1492–1493. Morgan Kaufmann.

Bradley C. Love, Douglas L. Medin, and Todd M. Gureckis. 2004. SUSTAIN: A network model of category learning. *Psych. Review*, 111(2):309–332.

Will Lowe. 2001. Towards a theory of semantic space. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pages 576–581.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proc. of SIGKDD-02*, pages 613–619, New York, NY, USA. ACM.

Fernando C. N. Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 183–190, Columbus, Ohio.

Daniel Ramage, Anna N. Rafferty, and Christopher D. Manning. 2009. Random walks for text semantic similarity. In *Proc. of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 23–31.

Carl E. Rasmussen. 2000. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems*, pages 554–560. MIT Press.

Yves Rosseel. 2002. Mixture models of categorization. *J. Math. Psychol.*, 46(2):178–210.

Mark Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proc. of SIGIR-94*, pages 142–151.

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

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP-08*.

Amos Tversky and Itamar Gati. 1982. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154.

Bram Vandekerckhove, Dominiek Sandra, and Walter Daelemans. 2009. A robust and extensible exemplar-based model of thematic fit. In *Proc. of EACL 2009*, pages 826–834. Association for Computational Linguistics.