KU: Word Sense Disambiguation by Substitution

Deniz Yuret
Koç University
Istanbul, Turkey
dyuret@ku.edu.tr

Abstract

Data sparsity is one of the main factors that make word sense disambiguation (WSD) difficult. To overcome this problem we need to find effective ways to use resources other than sense labeled data. In this paper I describe a WSD system that uses a statistical language model based on a large unannotated corpus. The model is used to evaluate the likelihood of various substitutes for a word in a given context. These likelihoods are then used to determine the best sense for the word in novel contexts. The resulting system participated in three tasks in the SemEval 2007 workshop. The WSD of prepositions task proved to be challenging for the system, possibly illustrating some of its limitations: e.g. not all words have good substitutes. The system achieved promising results for the English lexical sample and English lexical substitution tasks.

1 Introduction

A typical word sense disambiguation system is trained on a corpus of manually sense tagged text. Machine learning algorithms are then employed to find the best sense for a word in a novel context by generalizing from the training examples. The training data is costly to generate and inter-annotator agreement is difficult to achieve. Thus there is very little training data available: the largest single corpus of sense tagged text, SemCor, has 41,497 sense tagged words. (Yuret, 2004) observed that approximately half of the test instances do not match any of the contextual features learned from the training data for an all words disambiguation task. (Yarowsky and Florian, 2002) found that each successive doubling of the training data only leads to a 3-4% error reduction within their experimental range.

Humans do not seem to be cursed with an exponential training data requirement to become proficient with the use of a word. Dictionaries typically contain a definition and one or two examples of usage for each sense. This seems to be sufficient for a human to use the word correctly in contexts that share no surface features with the dictionary examples. The 10^8 waking seconds it takes a person to become proficient in a language does not seem sufficient to master all the words and their different senses. We need models that do not require large amounts of annotated text to perform WSD.

What possible process can explain our proficiency without relying on a lot of labeled data? Let us look at a concrete example: The two most frequent senses of the word “board” according to WordNet 3.0 (Fellbaum, 1998) are the “committee” sense, and the “plank” sense. When we hear a sentence like “There was a board meeting”, it is immediately obvious that the first sense is intended. One hypothesis is that a common sense inference engine in your brain rules out the second sense. Maybe you visualize pieces of timber sitting around a meeting table and decide that it is absurd. Another hypothesis is that the plank sense does not even occur to you because you hear this sentence in the middle of a conversation about corporate matters. Therefore the plank sense is not psychologically “primed”. Finally, maybe you subconsciously perform a substitution and the sentence

"There was a plank meeting" just sounds bad to your linguistic "ear".

In this paper I will describe a system that judges potential substitutions in a given context using a statistical language model as a surrogate for the linguistic "ear". The likelihoods of the various substitutes are used to select the best sense for a target word.

The use of substitutes for WSD is not new. (Leacock et al., 1998) demonstrated the use of related monosemous words (monosemous relatives) to collect examples for a given sense from the Internet. (Mihalcea, 2002) used the monosemous relatives technique for bootstrapping the automatic acquisition of large sense tagged corpora. In both cases, the focus was on collecting more labeled examples to be subsequently used with supervised machine learning techniques. (Martinez et al., 2006) extended the method to make use of polysemous relatives. More importantly, their method places these relatives in the context of the target word to query a search engine and uses the search results to predict the best sense in an unsupervised manner.

There are three areas that distinguish my system from the previous work: (i) The probabilities for substitutes in context are determined using a statistical language model rather than search hits on heuristically constructed queries, (ii) The set of substitutes are derived from multiple sources and optimized using WSD performance as the objective function, and (iii) A probabilistic generative model is used to select the best sense rather than typical machine learning algorithms or heuristics. Each of these areas is explained further below.

Probabilities for substitutes: Statistical language modeling is the art of determining the probability of a sequence of words. According to the model used in this study, the sentence "There was a committee meeting" is 17,629 times more likely than the sentence "There was a plank meeting". Thus, a statistical language model can be used as a surrogate for your inner ear that decides what sounds good and what sounds bad. I used a language model based on the Web 1T 5-gram dataset (Brants and Franz, 2006) which gives the counts of 1 to 5-grams in a web corpus of 10^{12} words. The details of the Web1T model are given in the Appendix.

Given that I criticize existing WSD algorithms for using too much data, it might seem hypocritical to employ a data source with 10^{12} words. In my defense, from an engineering perspective, an unannotated 10^{12} word corpus exists, whereas large sense tagged corpora do not. From a scientific perspective, it is clear that no human ever comes close to experiencing 10^{12} words, but they do outperform simple n-gram language models based on that much data in predicting the likelihood of words in novel contexts (Shannon, 1951). So, even though we do not know how humans do it, we do know that they have the equivalent of a powerful statistical language model in their heads.

Selecting the best substitutes: Perhaps more important for the performance of the system is the decision of which substitutes to try. We never thought of using "monkey" as a potential substitute for "board". One possibility is to use the synonyms in WordNet which were selected such that they can be interchanged in at least some contexts. However 54% of WordNet synsets do not have any synonyms. Besides, synonymous words would not always help if they share similar ambiguities in meaning. Substitutes that are not synonyms, on the other hand, may be very useful such as "hot" vs. "cold" or "car" vs. "truck". In general we are looking for potential substitutes that have a high likelihood of appearing in contexts that are associated with a specific sense of the target word. The substitute selection method used in this work is described in Section 3.

Selecting the best sense: Once we have a language model and a set of substitutes to try, we need a decision procedure that picks the best sense of a word in a given context. An unsupervised system can be designed to keep track of the sense associated with each substitute based on the lexical resource used. However since I used multiple lexical resources, and had training data available, I chose a supervised approach. For each instance in the training set, the likelihood of each substitute is determined. Then instances of a single sense are grouped together to yield a probability distribution over the substitutes for that sense. When a test instance is encountered its substitute distribution is compared to that of each sense to select the most appropriate one. Section 2 describes the sense selection procedure in detail.

We could say each context is represented with the likelihood it assigns to various substitutes rather than its surface features. That way contexts that do not share any surface features can be related to each other.

Results: To summarize the results, in the Word Sense Disambiguation of Prepositions Task, the system achieved 54.7% accuracy[1]. This is 15.1% above the baseline of picking the most frequent sense but 14.6% below the best system. In the Coarse Grained English Lexical Sample WSD Task, the system achieved 85.1% accuracy, which is 6.4% above the baseline of picking the most frequent sense and 3.6% below the best system. Finally, in the English Lexical Substitution Task, the system achieved the top result for picking the best substitute for each word.

2 Sense Selection Procedure

Consider a target word w_0 with n senses $S = \{s_1, \ldots, s_n\}$. Let $C_j = \{c_{j1}, c_{j2}, \ldots\}$ be the set of contexts in the training data where w_0 has been tagged with sense s_j. The prior probability of a sense s_j will be defined as:

$$P(s_j) = \frac{|C_j|}{\sum_{k=1}^{n} |C_k|}$$

Suppose we decide to use m substitutes $W = \{w_1, \ldots, w_m\}$. The selection of the possible substitutes is discussed in Section 3. Let $P(w_i, c)$ denote the probability of the context c where the target word has been replaced with w_i. This probability is obtained from the Web1T language model. The conditional probability of a substitute w_i in a particular context c is defined as:

$$P(w_i|c) = \frac{P(w_i, c)}{\sum_{w \in W} P(w, c)}$$

The conditional probability of a substitute w_i for a particular sense s_j is defined as:

$$P(w_i|s_j) = \frac{1}{|C_j|} \sum_{c \in C_j} P(w_i|c)$$

Given a test context c_t, we would like to find out which sense s_j it is most likely to represent:

$$\text{argmax}_j \; P(s_j|c_t) \propto P(c_t|s_j)P(s_j)$$

To calculate the likelihood of the test context $P(c_t|s_j)$, we first find the conditional probability distribution of the substitutes $P(w_i|c_t)$, as described above. Treating these probabilities as fractional counts we can express the likelihood as:

$$P(c_t|s_j) \propto \prod_{w \in W} P(w|s_j)^{P(w|c_t)}$$

Thus we choose the sense that maximizes the posterior probability:

$$\text{argmax}_j P(s_j) \prod_{w \in W} P(w|s_j)^{P(w|c_t)}$$

3 Substitute Selection Procedure

Potential substitutes for a word were selected from WordNet 3.0 (Fellbaum, 1998), and the Roget Thesaurus (Thesaurus.com, 2007).

When selecting the WordNet substitutes, the program considered all synsets of the target word and neighboring synsets accessible following a single link. All words contained within these synsets and their glosses were considered as potential substitutes.

When selecting the Roget substitutes, the program considered all entries that included the target word. By default, the entries that included the target word as part of a multi word phrase and entries that had the wrong part of speech were excluded.

I observed that the particular set of substitutes used had a large impact on the disambiguation performance in cross validation. Therefore I spent a considerable amount of effort trying to optimize the substitute sets. The union of the WordNet and Roget substitutes were first sorted based on their discriminative power measured by the likelihood ratio of their best sense:

$$\text{LR}(w_i) = \max_j \frac{P(w_i|s_j)}{P(w_i|\overline{s}_j)}$$

The following optimization algorithms were then run to maximize the leave-one-out cross validation (loocv) accuracy on the lexical sample WSD training data.

[1] In all the tasks participated, the system submitted a unique answer for each instance. Therefore precision, recall, F-measure, and accuracy have the same value. I will use the term accuracy to represent them all.

1. Each substitute was temporarily deleted and the resulting gain in loocv was noted. The substitute that led to the highest gain was permanently deleted. The procedure was repeated until no further loocv gain was possible.

2. Each pair of substitutes were tried alone and the pair that gave the highest loocv score was chosen as the initial list. Other substitutes were then greedily added to this list until no further loocv gain was possible.

3. Golden section search was used to find the ideal cutoff point in the list of substitutes sorted by likelihood ratio. Substitutes below the cutoff point were deleted.

None of these algorithms consistently gave the best result. Thus, each algorithm was run for each target word and the substitute set that gave the best loocv result was used for the final testing. The loocv gain from using the optimized substitute sets instead of the initial union of WordNet and Roget substitutes was significant. For example the average gain was 9.4% and the maximum was 38% for the English Lexical Sample WSD task.

4 English Lexical Substitution

The *English Lexical Substitution Task* (McCarthy and Navigli, 2007), for both human annotators and systems is to replace a target word in a sentence with as close a word as possible. It is different from the standard WSD tasks in that there is no sense repository used, and even the identification of a discrete sense is not necessary.

The task used a lexical sample of 171 words with 10 instances each. For each instance the human annotators selected several substitutes. There were three subtasks: **best:** scoring the best substitute for a given item, **oot:** scoring the best ten substitutes for a given item, and **mw:** detection and identification of multi-words. The details of the subtasks and scoring can be found in (McCarthy and Navigli, 2007). My system participated in the first two subtasks.

Because there is no training set, the supervised optimization of the substitute set using the algorithms described in Section 3 is not applicable. Based on the trial data, I found that the Roget substitutes work better than the WordNet substitutes most of the time. The antonyms in each entry and the entries that did not have the target word as the head were filtered out to improve the accuracy. Antonyms happen to be good substitutes for WSD, but not so good for lexical substitution.

BEST	P	R	Mode P	Mode R
all	12.90	12.90	20.65	20.65
Further Analysis				
NMWT	13.39	13.39	21.20	21.20
NMWS	14.33	13.98	21.88	21.42
RAND	12.67	12.67	20.34	20.34
MAN	13.16	13.16	21.01	21.01

OOT	P	R	Mode P	Mode R
all	46.15	46.15	61.30	61.30
Further Analysis				
NMWT	48.43	48.43	63.42	63.42
NMWS	49.72	49.72	63.74	63.74
RAND	47.80	47.80	62.84	62.84
MAN	44.23	44.23	59.55	59.55

Table 1: BEST and OOT results: P is precision, R is recall, Mode indicates accuracy selecting the single preferred substitute when there is one, NMWT is the score without items identified as multi-words, NMWS is the score using only single word substitutes, RAND is the score for the items selected randomly, and MAN is the score for the items selected manually.

For the final output of the system, the substitutes w_i in a context c were simply sorted by $P(w_i, c)$ which is calculated based on the Web1T language model.

In the **best** subtask the system achieved 12.9% accuracy, which is the top score and 2.95% above the baseline. The system was able to find the mode (a single substitute preferred to the others by the annotators) in 20.65% of the cases when there was one, which is 5.37% above the baseline and 0.08% below the top score. The top part of Table 1 gives the breakdown of the **best** score, see (McCarthy and Navigli, 2007) for details.

The low numbers here are partly a consequence of the scoring formula used. Specifically, the score for a single item is bounded by the frequency of the best substitute in the gold standard file. Therefore, the

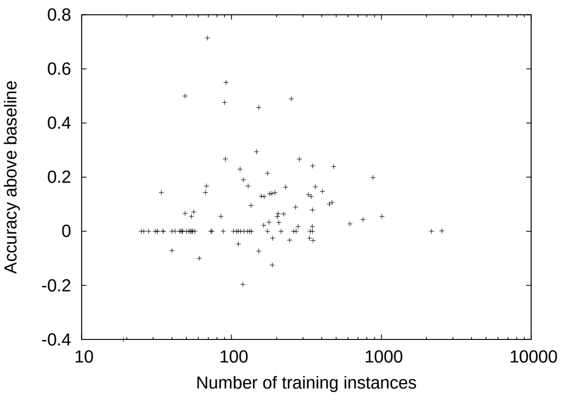

Figure 1: Training set size vs. accuracy above baseline for the English lexical sample task.

highest achievable score was not 100%, but 45.76%. A more intuitive way to look at the result may be the following: Human annotators assigned 4.04 distinct substitutes for each instance on average, and my system was able to guess one of these as the best in 33.73% of the cases.

In the **oot** subtask the system achieved 46.15% accuracy, which is 16.45% above the baseline and 22.88% below the top result. The system was able to find the mode as one of its 10 guesses in 61.30% of the cases when there was a mode, which is 20.73% above the best baseline and 4.96% below the top score. Unlike the **best** scores, 100% accuracy is possible for **oot**. Each item had 1 to 9 distinct substitutes in the gold standard, so an ideal system could potentially cover them all with 10 guesses. The second part of Table 1 gives the breakdown of the **oot** score.

In conclusion, selecting substitutes based on a standard repository like Roget and ranking them using the ngram language model gives a good baseline for this task. To improve the performance along these lines we need better language models, and better substitute selection procedures. Even the best language model will only tell us which words are most likely to replace our target word, not which ones preserve the meaning. Relying on repositories like Roget for the purpose of substitute selection seems ad-hoc and better methods are needed.

5 English Lexical Sample WSD

The *Coarse-Grained English Lexical Sample WSD Task* (Palmer et al., 2007), provided training and test data for sense disambiguation of 65 verbs and 35 nouns. On average there were 223 training and 49 testing instances for each word tagged with an OntoNote sense tag (Hovy et al., 2006). OntoNote sense tags are groupings of WordNet senses that are more coarse-grained than traditional WN entries, and which have achieved on average 90% inter-annotator agreement. The number of senses for a word ranged from 1 to 13 with an average of 3.6.

I used substitute sets optimized for each word as described in Section 3. Then a single best sense for each test instance was selected based on the model given in Section 2. The system achieved 85.05% accuracy, which is 6.39% above the baseline of picking the most frequent sense and 3.65% below the top score.

These numbers seem higher than previous Senseval lexical sample tasks. The best system in Senseval-3 (Mihalcea et al., 2004; Grozea, 2004) achieved 72.9% fine grained, 79.3% coarse grained accuracy. Many factors may have played a role but the most important one is probably the sense inventory. The nouns and verbs in Senseval-3 had 6.1 fine grained and 4.5 coarse grained senses on average.

The leave-one-out cross-validation result of my system on the training set was 83.21% with the unfiltered union of Roget and WordNet substitutes, and 90.69% with the optimized subset. Clearly there is some over-fitting in the substitute optimization process which needs to be improved.

Table 2 details the performance on individual words. The accuracy is 88.67% on the nouns and 81.02% on the verbs. One can clearly see the relation of the performance with the number of senses (decreasing) and the frequency of the first sense (increasing). Interestingly no clear relation exists between the training set size and the accuracy above the baseline. Figure 1 plots the relationship between training set size vs. the accuracy gain above the most frequent sense baseline. This could indicate that the system peaks at a low training set size and generalizes well because of the language model. However, it should be noted that each point in the plot represents a different word, not experiments with the same word at different training set sizes. Thus the difficulty of each word may be the overriding factor in determining performance. A more detailed study similar to (Yarowsky and Florian, 2002) is needed to explore the relationship in more detail.

6 WSD of Prepositions

The *Word Sense Disambiguation of Prepositions Task* (Litkowski and Hargraves, 2007), provided training and test data for sense disambiguation of 34 prepositions. On average there were 486 training and 234 test instances for each preposition. The number of senses for a word ranged from 1 to 20 with an average of 7.4.

The system described in Sections 2 and 3 were applied to this task as well. WordNet does not have information about prepositions, so most of the candidate substitutes were obtained from Roget and The Preposition Project (Litkowski, 2005). After optimizing the substitute sets the system achieved 54.7% accuracy which is 15.1% above the most frequent sense baseline and 14.6% below the top result. Unfortunately there were only three teams that participated in this task. The detailed breakdown of the results can be seen in the second part of Table 2.

The loocv result on the training data with the initial unfiltered set of substitutes was 51.70%. Optimizations described in Section 3 increased this to 59.71%. This increase is comparable to the one in the lexical substitution task. The final result of 54.7% shows signs of overfitting in the substitute selection process.

The average gain above the baseline for prepositions (39.6% to 54.7%) is significantly higher than the English lexical sample task (78.7% to 85.1%). However the preposition numbers are generally lower compared to the nouns and verbs because they are more ambiguous: the number of senses is higher and the first sense frequency is lower.

Good quality substitutes are difficult to find for prepositions. Unlike common nouns and verbs, common prepositions play unique roles in language and are difficult to replace. Open class words have synonyms, hypernyms, antonyms etc. that provide good substitutes: it is easy to come up with “I ate halibut” when you see “I ate fish”. It is not as easy to replace “of” in the phrase “the president of the company”. Even when there is a good substitute, e.g. “over” vs. “under”, the two prepositions usually share the exact same ambiguities: they can both express a physical direction or a quantity comparison. Therefore the substitution based model presented in this work may not be a good match for preposition disambiguation.

7 Contributions and Future Work

A WSD method employing a statistical language model was introduced. The language model is used to evaluate the likelihood of possible substitutes for the target word in a given context. Each context is represented with its preferences for possible substitutes, thus contexts with no surface features in common can nevertheless be related to each other.

The set of substitutes used for a word had a large effect on the performance of the resulting system. A substitute selection procedure that uses the language model itself rather than external lexical resources may work better.

I hypothesize that the model would be advantageous on tasks like “all words” WSD, where data sparseness is paramount, because it is able to link contexts with no surface features in common. It can be used in an unsupervised manner where the substitutes and their associated senses can be obtained from a lexical resource. Work along these lines was not completed due to time limitations.

Finally, there are two failure modes for the algorithm: either there are no good substitutes that differentiate the various senses (as I suspect is the case for some prepositions), or the language model does not yield accurate preferences among the substitutes that correspond to our intuition. In the first case we have to fall back on other methods, as the substitutes obviously are of limited value. The correspondence between the language model and our intuition requires further study.

Appendix: Web1T Language Model

The Web 1T 5-gram dataset (Brants and Franz, 2006) that was used to build a language model for this work consists of the counts of word sequences up to length 5 in a 10^{12} word corpus derived from the Web. The data consists of mostly English words that have been tokenized and sentence tagged. Tokens that appear less than 200 times and ngrams that appear less than 40 times have been filtered out.

I used a smoothing method loosely based on the *one-count* method given in (Chen and Goodman, 1996). Because ngrams with low counts are not included in the data I used ngrams with missing counts instead of ngrams with one counts. The missing count is defined as:

$$m(w_{i-n+1}^{i-1}) = c(w_{i-n+1}^{i-1}) - \sum_{w_i} c(w_{i-n+1}^{i})$$

where w_{i-n+1}^{i} indicates the n-word sequence ending with w_i, and $c(w_{i-n+1}^{i})$ is the count of this sequence. The corresponding smoothing formula is:

$$P(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) + (1+\alpha_n)m(w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1})}{c(w_{i-n+1}^{i-1}) + \alpha_n m(w_{i-n+1}^{i-1})}$$

The parameters $\alpha_n > 0$ for $n = 2 \ldots 5$ was optimized on the Brown corpus to yield a cross entropy of 8.06 bits per token. The optimized parameters are given below:

$$\alpha_2 = 6.71,\ \alpha_3 = 5.94,\ \alpha_4 = 6.55,\ \alpha_5 = 5.71$$

References

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium, Philadelphia. LDC2006T13.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*.

Christiane Fellbaum, editor. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press.

Cristian Grozea. 2004. Finding optimal parameter settings for high performance word sense disambiguation. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Eduard H. Hovy, M. Marcus, M. Palmer, S. Pradhan, L. Ramshaw, and R. Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology / North American Association of Computational Linguistics conference (HLT-NAACL 2006)*, New York, NY. Short paper.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–166, March.

Ken Litkowski and Orin Hargraves. 2007. SemEval-2007 Task 06: Word sense disambiguation of prepositions. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*.

K. C. Litkowski. 2005. The preposition project. In *Proceedings of the Second ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, Colchester, England, April. University of Essex.

David Martinez, Eneko Agirre, and Xinglong Wang. 2006. Word relatives in context for word sense disambiguation. In *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW 2006)*, pages 42–50.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 Task 10: English lexical substitution task. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Rada Mihalcea. 2002. Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Languages Resources and Evaluations LREC 2002*, Las Palmas, Spain, May.

Martha Palmer, Sameer Pradhan, and Edward Loper. 2007. SemEval-2007 Task 17: English lexical sample, English SRL and English all-words tasks. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*.

Claude Elwood Shannon. 1951. Prediction and entropy of printed English. *The Bell System Technical Journal*, 30:50–64.

Thesaurus.com. 2007. *Roget's New Millennium*TM*Thesaurus, First Edition (v 1.3.1)*. Lexico Publishing Group, LLC. http://thesaurus.reference.com.

David Yarowsky and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310.

Deniz Yuret. 2004. Some experiments with a Naive Bayes WSD system. In *ACL 2004 Senseval-3 Workshop*, Barcelona, Spain, July.

English Lexical Sample WSD

lexelt	trn/tst	s	mfs	acc	lexelt	trn/tst	s	mfs	acc
affect.v	45/19	1	1.000	1.000	allow.v	108/35	2	0.971	0.971
announce.v	88/20	2	1.000	1.000	approve.v	53/12	2	0.917	0.917
area.n	326/37	3	0.703	0.838	ask.v	348/58	6	0.517	0.759
attempt.v	40/10	1	1.000	1.000	authority.n	90/21	4	0.238	0.714
avoid.v	55/16	1	1.000	1.000	base.n	92/20	5	0.100	0.650
begin.v	114/48	4	0.562	0.792	believe.v	202/55	2	0.782	0.836
bill.n	404/102	3	0.755	0.902	build.v	119/46	3	0.739	0.543
buy.v	164/46	5	0.761	0.783	capital.n	278/57	4	0.965	0.982
care.v	69/7	3	0.286	1.000	carrier.n	111/21	7	0.714	0.667
cause.v	73/47	1	1.000	1.000	chance.n	91/15	4	0.400	0.667
claim.v	54/15	3	0.800	0.800	come.v	186/43	10	0.233	0.372
complain.v	32/14	2	0.857	0.857	complete.v	42/16	2	0.938	0.938
condition.n	132/34	2	0.765	0.765	contribute.v	35/18	2	0.500	0.500
defense.n	120/21	7	0.286	0.476	describe.v	57/19	3	1.000	1.000
development.n	180/29	3	0.621	0.759	disclose.v	55/14	1	0.929	0.929
do.v	207/61	4	0.902	0.934	drug.n	205/46	2	0.870	0.935
effect.n	178/30	3	0.767	0.800	end.v	135/21	4	0.524	0.619
enjoy.v	56/14	2	0.571	0.643	estimate.v	74/16	1	1.000	1.000
examine.v	26/3	3	1.000	1.000	exchange.n	363/61	5	0.738	0.902
exist.v	52/22	2	1.000	1.000	explain.v	85/18	2	0.889	0.944
express.v	47/10	1	1.000	1.000	feel.v	347/51	3	0.686	0.765
find.v	174/28	5	0.821	0.821	fix.v	32/2	5	0.500	0.500
future.n	350/146	3	0.863	0.829	go.v	244/61	12	0.459	0.426
grant.v	19/5	2	0.800	0.400	hold.v	129/24	8	0.375	0.542
hope.v	103/33	1	1.000	1.000	hour.n	187/48	4	0.896	0.771
improve.v	31/16	1	1.000	1.000	job.n	188/39	3	0.821	0.795
join.v	68/18	4	0.389	0.556	keep.v	260/80	7	0.562	0.562
kill.v	111/16	4	0.875	0.875	lead.v	165/39	6	0.385	0.513
maintain.v	61/10	2	0.900	0.800	management.n	284/45	2	0.711	0.978
move.n	270/47	4	0.979	0.979	need.v	195/56	2	0.714	0.857
negotiate.v	25/9	1	1.000	1.000	network.n	152/55	3	0.909	0.836
occur.v	47/22	2	0.864	0.864	order.n	346/57	7	0.912	0.930
part.n	481/71	4	0.662	0.901	people.n	754/115	4	0.904	0.948
plant.n	347/64	2	0.984	0.984	point.n	469/150	9	0.813	0.920
policy.n	331/39	2	0.974	0.949	position.n	268/45	7	0.467	0.556
power.n	251/47	3	0.277	0.766	prepare.v	54/18	2	0.778	0.833
president.n	879/177	3	0.729	0.927	produce.v	115/44	2	0.750	0.750
promise.v	50/8	2	0.750	0.750	propose.v	34/14	2	0.857	1.000
prove.v	49/22	3	0.318	0.818	purchase.v	35/15	1	1.000	1.000
raise.v	147/34	7	0.147	0.441	rate.n	1009/145	2	0.862	0.917
recall.v	49/15	3	0.867	0.933	receive.v	136/48	2	0.958	0.958
regard.v	40/14	3	0.714	0.643	remember.v	121/13	2	1.000	1.000
remove.v	47/17	1	1.000	1.000	replace.v	46/15	2	1.000	1.000
report.v	128/35	3	0.914	0.914	rush.v	28/7	2	1.000	1.000
say.v	2161/541	5	0.987	0.987	see.v	158/54	6	0.444	0.574
set.v	174/42	9	0.286	0.500	share.n	2536/525	2	0.971	0.973
source.n	152/35	5	0.371	0.829	space.n	67/14	5	0.786	0.929
start.v	214/38	6	0.447	0.447	state.n	617/72	3	0.792	0.819
system.n	450/70	5	0.486	0.586	turn.v	340/62	13	0.387	0.516
value.n	335/59	3	0.983	0.983	work.v	230/43	7	0.558	0.721
AVG	222.8/48.5	3.6	0.787	0.851					

Preposition WSD

lexelt	trn/tst	s	mfs	acc	lexelt	trn/tst	s	mfs	acc
about.p	710/364	6	0.885	0.934	above.p	48/23	5	0.609	0.522
across.p	319/151	2	0.960	0.960	after.p	103/53	6	0.434	0.585
against.p	195/92	6	0.435	0.793	along.p	364/173	3	0.954	0.954
among.p	100/50	3	0.300	0.680	around.p	334/155	6	0.452	0.535
as.p	173/84	1	1.000	1.000	at.p	715/367	12	0.425	0.662
before.p	47/20	3	0.450	0.850	behind.p	138/68	4	0.662	0.676
beneath.p	57/28	3	0.571	0.679	beside.p	62/29	1	1.000	1.000
between.p	211/102	7	0.422	0.765	by.p	509/248	10	0.371	0.556
down.p	332/153	3	0.438	0.647	during.p	81/39	2	0.385	0.564
for.p	950/478	13	0.238	0.395	from.p	1204/578	16	0.279	0.415
in.p	1391/688	13	0.362	0.436	inside.p	67/38	4	0.526	0.579
into.p	604/297	8	0.451	0.539	like.p	266/125	7	0.768	0.808
of.p	3000/1478	17	0.205	0.374	off.p	161/76	4	0.763	0.776
on.p	872/441	20	0.206	0.469	onto.p	117/58	3	0.879	0.879
over.p	200/98	12	0.327	0.510	round.p	181/82	7	0.378	0.512
through.p	440/208	15	0.495	0.538	to.p	1182/572	10	0.322	0.579
towards.p	214/102	4	0.873	0.873	with.p	1187/578	15	0.249	0.455
AVG	486.3/238.1	7.4	0.397	0.547					

Table 2: English Lexical Sample and Preposition WSD Results: lexelt is the lexical item, trn/tst is the number of training and testing instances, s is the number of senses in the training set, mfs is the most frequent sense baseline, and acc is the final accuracy.