

The Role of Semantic Roles in Disambiguating Verb Senses

Hoa Trang Dang

National Institute of Standards and Technology
Gaithersburg, MD 20899
hoa.dang@nist.gov

Martha Palmer

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
mpalmer@cis.upenn.edu

Abstract

We describe an automatic Word Sense Disambiguation (WSD) system that disambiguates verb senses using syntactic and semantic features that encode information about predicate arguments and semantic classes. Our system performs at the best published accuracy on the English verbs of Senseval-2. We also experiment with using the gold-standard predicate-argument labels from PropBank for disambiguating fine-grained WordNet senses and course-grained PropBank framesets, and show that disambiguation of verb senses can be further improved with better extraction of semantic roles.

1 Introduction

A word can have different meanings depending on the context in which it is used. Word Sense Disambiguation (WSD) is the task of determining the correct meaning (“sense”) of a word in context, and several efforts have been made to develop automatic WSD systems. Early work on WSD (Yarowsky, 1995) was successful for easily distinguishable homonyms like *bank*, which have multiple unrelated meanings. While homonyms are fairly tractable, highly polysemous verbs, which have related but subtly distinct senses, pose the greatest challenge for WSD systems (Palmer et al., 2001).

Verbs are syntactically complex, and their syntax is thought to be determined by their underlying semantics (Grimshaw, 1990; Levin, 1993). Levin verb

classes, for example, are based on the ability of a verb to occur in pairs of syntactic frames (diathesis alternations); different senses of a verb belong to different verb classes, which have different sets of syntactic frames that are supposed to reflect underlying semantic components that constrain allowable arguments. If this is true, then the correct sense of a verb should be revealed (at least partially) in its arguments.

In this paper we show that the performance of automatic WSD systems can be improved by using richer linguistic features that capture information about predicate arguments and their semantic classes. We describe our approach to automatic WSD of verbs using maximum entropy models to combine information from lexical collocations, syntax, and semantic class constraints on verb arguments. The system performs at the best published accuracy on the English verbs of the Senseval-2 (Palmer et al., 2001) exercise on evaluating automatic WSD systems. The Senseval-2 verb instances have been manually tagged with their WordNet sense and come primarily from the Penn Treebank WSJ. The WSJ corpus has also been manually annotated for predicate arguments as part of PropBank (Kingsbury and Palmer, 2002), and the intersection of PropBank and Senseval-2 forms a corpus containing gold-standard annotations of WordNet senses and PropBank semantic role labels. This provides a unique opportunity to investigate the role of predicate arguments in verb sense disambiguation. We show that our system’s accuracy improves significantly by adding features from PropBank, which explicitly encodes the predicate-argument informa-

tion that our original set of syntactic and semantic class features attempted to capture.

2 Basic automatic system

Our WSD system was built to combine information from many different sources, using as much linguistic knowledge as could be gathered automatically by NLP tools. In particular, our goal was to see the extent to which sense-tagging of verbs could be improved by adding features that capture information about predicate-arguments and selectional restrictions.

We used the Mallet toolkit (McCallum, 2002) for learning maximum entropy models with Gaussian priors for all our experiments. In order to extract the linguistic features necessary for the models, all sentences containing the target word were automatically part-of-speech-tagged using a maximum entropy tagger (Ratnaparkhi, 1998) and parsed using the Collins parser (Collins, 1997). In addition, an automatic named entity tagger (Bikel et al., 1997) was run on the sentences to map proper nouns to a small set of semantic classes.¹

2.1 Topical features

We categorized the possible model features into topical features and several types of local contextual features. Topical features for a verb in a sentence look for the presence of keywords occurring *anywhere* in the sentence and any surrounding sentences provided as context (usually one or two sentences). These features are supposed to show the domain in which the verb is being used, since some verb senses are used in only certain domains. The set of keywords is specific to each verb lemma to be disambiguated and is determined automatically from training data so as to minimize the entropy of the probability of the senses conditioned on the keyword. All alphabetic characters are converted to lower case. Words occurring less than twice in the training data or that are in a stoplist² of pronouns, prepositions, and conjunctions are ignored.

¹The inclusion or omission of a particular company or product implies neither endorsement nor criticism by NIST. Any opinions, findings, and conclusions expressed are the authors' own and do not necessarily reflect those of NIST.

²<http://www.d.umn.edu/~tperdese/Group01/WordNet/words.txt>

2.2 Local features

The local features for a verb w in a particular sentence tend to look only within the smallest clause containing w . They include *collocational* features requiring no linguistic preprocessing beyond part-of-speech tagging, *syntactic* features that capture relations between the verb and its complements, and *semantic* features that incorporate information about noun classes for subjects and objects:

Collocational features: Collocational features refer to ordered sequences of part-of-speech tags or word tokens immediately surrounding w . They include:

- unigrams: words $w_{-2}, w_{-1}, w_0, w_{+1}, w_{+2}$ and parts of speech $p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}$, where w_i and p_i are at position i relative to w
- bigrams: $w_{-2}w_{-1}, w_{-1}w_{+1}, w_{+1}w_{+2}; p_{-2}p_{-1}, p_{-1}p_{+1}, p_{+1}p_{+2}$
- trigrams: $w_{-3}w_{-2}w_{-1}, w_{-2}w_{-1}w_{+1}, w_{-1}w_{+1}w_{+2}, w_{+1}w_{+2}w_{+3}; p_{-3}p_{-2}p_{-1}, p_{-2}p_{-1}p_{+1}, p_{-1}p_{+1}p_{+2}, p_{+1}p_{+2}p_{+3}$

Syntactic features: The system uses heuristics to extract syntactic elements from the parse for the sentence containing w . Let commander VP be the lowest VP that dominates w and that is not immediately dominated by another VP, and let head VP be the lowest VP dominating w (See Figure 1). Then we define the *subject* of w to be the leftmost NP sibling of commander VP, and a *complement* of w to be a node that is a child of the head VP, excluding NPs whose head is a number or a noun from a list of common temporal nouns (“week”, “tomorrow”, “Monday”, etc.). The system extracts the following binary syntactic features:

- Is the sentence passive?
- Is there a subject, direct object (leftmost NP complement of w), indirect object (second leftmost NP complement of w), or clausal complement (S complement of w)?
- What is the word (if any) that is the particle or head of the subject, direct object, or indirect object?

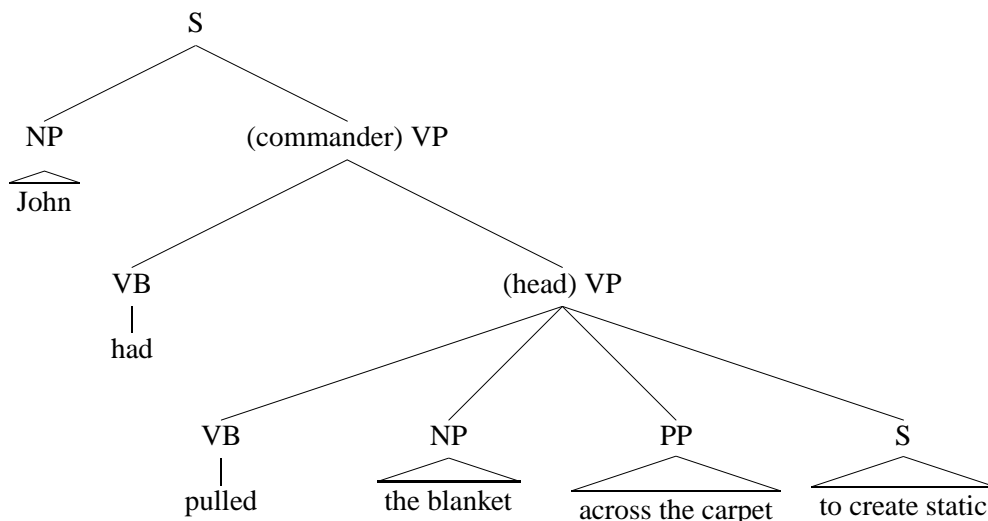


Figure 1: Example parse tree for w ="pulled", from which is extracted the syntactic features: $morph=normal$ $subj\ dobj\ sent-comp\ subj=john\ dobj=blanket\ prep=across\ across-obj=carpet$.

- If there is a PP complement, what is the preposition, and what is the object of the preposition?

Semantic features:

- What is the Named Entity tag (PERSON, ORGANIZATION, LOCATION, UNKNOWN) for each proper noun in the syntactic positions above?
- What are the possible WordNet synsets and hypernyms for each noun in the syntactic positions above? (Nouns are not explicitly disambiguated; all possible synsets and hypernyms for the noun are included.)

This set of local features relies on access to syntactic structure as well as semantic class information, and attempts to model richer linguistic information about predicate arguments. However, the heuristics for extracting the syntactic features are able to identify subjects and objects of only simple clauses. The heuristics also do not differentiate between arguments and adjuncts; for example, the feature $sent-comp$ is intended to identify clausal complements such as in (S (NP Mary) (VP (VB called) (S him a bastard))), but Figure 1 shows how a purpose clause can be mistakenly labeled as a clausal complement.

2.3 Evaluation

We tested the system on the 1806 test instances of the 29 verbs from the English lexical sample task for Senseval-2 (Palmer et al., 2001). Accuracy was defined to be the fraction of the instances for which the system got the correct sense. All significance testing between different accuracies was done using a one-tailed z-test, assuming a binomial distribution of the successes; differences in accuracy were considered to be significant if $p \leq 0.050$.

In Senseval-2, senses involving multi-word constructions could be identified directly from the sense tags themselves, and the head word and satellites of multi-word constructions were explicitly marked in the training and test data. We trained one model for each of the verbs and used a filter to consider only phrasal senses whenever there were satellites of multi-word constructions marked in the test data.

Feature	Accuracy
co	0.571
co+syn	0.598
co+syn+sem	0.625

Table 1: Accuracy of system on Senseval-2 verbs using topical features and different subsets of local features.

Table 1 shows the accuracy of the system using topical features and different subsets of local fea-

tures. Adding features from richer linguistic sources always improves accuracy. Adding lexical syntactic (“syn”) features improves accuracy significantly over using just collocational (“co”) features ($p = 0.050$). When semantic class (“sem”) features are added, the improvement is also significant.

Adding topical information to all the local features improves accuracy, but not significantly; when the topical features are removed the accuracy of our system falls only slightly, to 62.0%. Senses based on domain or topic occur rarely in the Senseval-2 corpus. Most of the information provided by topical features already seem to be captured by the local features for the frequent senses.

Features	Accuracy
co+syn	0.598
co+syn+ne	0.597
co+syn+wn	0.623
co+syn+ne+wn	0.625

Table 2: Accuracy of system on Senseval-2 verbs, using topical features and different subsets of semantic class features.

Semantic class information plays a significant role in sense distinctions. Table 2 shows the relative contribution of adding only named entity tags to the collocational and syntactic features (“co+syn+ne”), versus adding only the WordNet classes (“co+syn+wn”), versus adding both named entity and WordNet classes (“co+syn+ne+wn”). Adding all possible WordNet noun class features for arguments contributes a large number of parameters to the model, but this use of WordNet with no separate disambiguation of noun arguments proves to be very useful. In fact, the use of WordNet for common nouns proves to be even more beneficial than the use of a named entity tagger for proper nouns. Given enough data, the maximum entropy model is able to assign high weights to the correct hypernyms of the correct noun sense if they represent defining selectional restrictions.

Incorporating topical keywords as well as collocational, syntactic, and semantic local features, our system achieves 62.5% accuracy. This is in comparison to the 61.1% accuracy achieved by (Lee and Ng, 2002), which has been the best published result on this corpus.

3 PropBank semantic annotations

Our WSD system uses heuristics to attempt to detect predicate arguments from parsed sentences. However, recognition of predicate argument structures is not straightforward, because a natural language will have several different syntactic realizations of the same predicate argument relations.

PropBank is a corpus in which verbs are annotated with semantic tags, including coarse-grained sense distinctions and predicate-argument structures. PropBank adds a layer of semantic annotation to the Penn Wall Street Journal Treebank II. An important goal is to provide consistent predicate-argument structures across different syntactic realizations of the same verb. Polysemous verbs are also annotated with different *framesets*. Frameset tags are based on differences in subcategorization frames and correspond to a coarse notion of word senses.

A verb’s semantic arguments in PropBank are numbered beginning with 0. Arg0 is roughly equivalent to the thematic role of Agent, and Arg1 usually corresponds to Theme or Patient; however, argument labels are not necessarily consistent across different senses of the same verb, or across different verbs, as thematic roles are usually taken to be. In addition to the core, numbered arguments, verbs can take any of a set of general, adjunct-like arguments (ARGM), whose labels are derived from the Treebank functional tags (DIRection, LOCation, etc.).

PropBank provides manual annotation of predicate-argument information for a large number of verb instances in the Senseval-2 data set. The intersection of PropBank and Senseval-2 forms a corpus containing gold-standard annotations of fine-grained WordNet senses, coarse-grained PropBank framesets, and PropBank role labels. The combination of such gold-standard semantic annotations provides a unique opportunity to investigate the role of predicate-argument features in word sense disambiguation, for both coarse-grained framesets and fine-grained WordNet senses.

3.1 PropBank features

We conducted experiments on the effect of using features from PropBank for sense-tagging verbs. Both PropBank role labels and PropBank framesets were used. In the case of role labels, only the

gold-standard labels found in PropBank were used, because the best automatic semantic role labelers only perform at about 84% precision and 75% recall (Pradhan et al., 2004).

From the PropBank annotation for each sentence, we extracted the following features:

1. Labels of the semantic roles: rel, ARG0, ARG1, ARG2-WITH, ARG2, ..., ARGM-LOC, ARGM-TMP, ARGM-NEG, ...
2. Syntactic labels of the constituent instantiating each semantic role: ARG0=NP, ARGM-TMP=PP, ARG2-WITH=PP, ...
3. Head word of each constituent in (2): rel=called, sats=up, ARG0=company, ARGM-TMP=day, ...
4. Semantic classes (named entity tag, WordNet hypernyms) of the nouns in (3): ARG0syn=ORGANIZATION, ARG0syn=16185, ARGM-TMPsyn=13018, ...

When a numbered role appears in a prepositional phrase (e.g., ARG2-WITH), we take the “head word” to be the object of the preposition. If a constituent instantiating some semantic role is a trace, we take the head of its referent instead.

- [ARG0 Mr. Bush] has [rel called] [ARG1-for for an agreement by next September at the latest] .

For example, the PropBank features that we extract for the sentence above are:

```
arg0 arg0=bush arg0syn=person arg0syn=1740 ...
rel rel=called
arg1-for arg1 arg1=agreement arg1syn=12865 ...
```

3.2 Role labels for frameset tagging

We collected all instances of the Senseval-2 verbs from the PropBank corpus. Only 20 of these verbs had more than one frameset in the PropBank corpus, resulting in 4887 instances of polysemous verbs. The instances for each word were partitioned randomly into 10 equal parts, and the system was tested on each part after being trained on the remaining nine. For these 20 verbs with more than one PropBank frameset tag, choosing the most frequent frameset gives a baseline accuracy of 76.0%.

The sentences were automatically pos-tagged with the Ratnaparki tagger and parsed with the Collins parser. We extracted local contextual features as for WordNet sense-tagging and used the local features to train our WSD system on the coarse-grained sense-tagging task of automatically assigning PropBank frameset tags. We tested the effect of using only collocational features (“co”) for frameset tagging, as well as using only PropBank role features (“pb”) or only our original syntactic/semantic features (“synsem”) for this task, and found that the combination of collocational features with PropBank features worked best. The system has the worst performance on the word *strike*, which has a high number of framesets and a low number of training instances. Table 3 shows the performance of the system on different subsets of local features.

Feature	Accuracy
baseline	0.760
co	0.853
synsem	0.859
co+synsem	0.883
pb	0.901
co+pb	0.908
co+synsem+pb	0.907

Table 3: Accuracy of system on frameset-tagging task for verbs with more than one frameset, using different types of local features (no topical features); all features except pb were extracted from automatically pos-tagged and parsed sentences.

We obtained an overall accuracy of 88.3% using our original local contextual features. However, the system’s performance improved significantly when we used only PropBank role features, achieving an accuracy of 90.1%. Furthermore, adding collocational features and heuristically extracted syntactic/semantic features to the PropBank features do not provide additional information and affects the accuracy of frameset-tagging only negligibly. It is not surprising that for the coarse-grained sense-tagging task of assigning the correct PropBank frameset tag to a verb, using the PropBank role labels is better than syntactic/semantic features heuristically extracted from parses because these heuristics are meant to capture the predicate-argument informa-

tion that is encoded more directly in the PropBank role labels.

Even when the original local features were extracted from the gold-standard pos-tagged and parsed sentences of the Penn Treebank, the system performed significantly worse than when PropBank role features were used. This suggests that more effort should be applied to improving the heuristics for extracting syntactic features.

We also experimented with adding topical features and ARGM features from PropBank. In all cases, these additional features reduced overall accuracy, but the difference was never significant ($p \geq 0.100$). Topical features do not help because frameset tags are based on differences in subcategorization frames and not on the domain or topic. ARGM features do not help because they are supposedly used uniformly across verbs and framesets.

3.3 Role labels for WordNet sense-tagging

We experimented with using PropBank role labels for fine-grained WordNet sense-tagging. While ARGM features are not useful for coarse-grained frameset-tagging, some sense distinctions in WordNet are based on adverbial modifiers, such as “live well” or “serves someone well.” Therefore, we included PropBank ARGM features in our models for WordNet sense-tagging to capture a wider range of linguistic behavior. We looked at the 2571 instances of 29 Senseval-2 verbs that were in both Senseval-2 and the PropBank corpus.

Features	Accuracy
co	0.628
synsem	0.638
co+synsem	0.666
pb	0.656
co+pb	0.681
co+synsem+pb	0.694

Table 4: Accuracy of system on WordNet sense-tagging for instances in both Senseval-2 and PropBank, using different types of local features (no topical features).

Table 4 shows the accuracy of the system on WordNet sense-tagging using different subsets of features; all features except pb were extracted from automatically pos-tagged and parsed sentences. By

adding PropBank role features to our original local feature set, accuracy rose from 0.666 to 0.694 on this subset of the Senseval-2 verbs ($p = 0.020$); the extraction of syntactic features from the parsed sentences is again not successfully capturing all the predicate-argument information that is explicit in PropBank.

The verb “match” illustrates why accuracy improves using additional PropBank features. As shown in Figure 2, the matched objects may occur in different grammatical relations with respect to the verb (subject, direct object, object of a preposition), but they each have an ARG1 semantic role label in PropBank.³ Furthermore, only one of the matched objects needs to be specified, as in Example 3 where the second matched object (presumably the company’s prices) is unstated. Our heuristics do not handle these alternations, and cannot detect that the syntactic subject in Example 1 has a different semantic role than the subject of Example 3.

Roleset **match.01** “match”:

Arg0: person performing match

Arg1: matching objects

Ex1: [*ARG1* the wallpaper] [*rel* matched] [*ARG1* the paint]

Ex2: [*ARG0* The architect] [*rel* matched] [*ARG1* the paint] [*Arg1-WITH* with the wallpaper]

Ex3: [*ARG0* The company] [*rel* matched] [*ARG1* Kodak’s higher prices]

Figure 2: PropBank roleset for “match”

Our basic WSD system (using local features extracted from automatic parses) confused WordNet Sense 1 with Sense 4:

1. match, fit, correspond, check, jibe, gibe, tally, agree – (be compatible, similar or consistent; coincide in their characteristics; “The two stories don’t agree in many details”; “The handwriting checks with the signature on the check”; “The suspect’s fingerprints don’t match those on the gun”)
4. equal, touch, rival, match – (be equal to in

³PropBank annotation for “match” allows multiple ARG1 labels, one for each of the matching objects. Other verbs that have more than a single ARG1 in PropBank include: “attach, bolt, coincide, connect, differ, fit, link, lock, pin, tack, tie.”

quality or ability; “Nothing can rival cotton for durability”; “Your performance doesn’t even touch that of your colleagues”; “Her persistence and ambition only matches that of her parents”)

The senses are differentiated in that the matching objects (ARG1) in Sense 4 have some quantifiable characteristic that can be measured on some scale, whereas those in Sense 1 are more general. Gold-standard PropBank annotation of ARG1 allows the system to generalize over the semantic classes of the arguments and distinguish these two senses more accurately.

3.4 Frameset tags for WordNet sense-tagging

PropBank frameset tags (either gold-standard or automatically tagged) were incorporated as features in our WSD system to see if knowing the coarse-grained sense tags would be useful in assigning fine-grained WordNet sense tags. A frameset tag for the instance was appended to each feature; this effectively partitions the feature set according to the coarse-grained sense provided by the frameset. To automatically tag an instance of a verb with its frameset, the set of all instances of the verb in PropBank was partitioned into 10 subsets, and an instance in one subset was tagged by training a maximum entropy model on the instances in the other nine subsets. Various local features were considered, and the same feature types were used to train the frameset tagger and the WordNet sense tagger that used the automatically-assigned frameset.

For the 20 Senseval-2 verbs that had more than one frameset in PropBank, we extracted all instances that were in both Senseval-2 and PropBank, yielding 1468 instances. We examined the effect of incorporating the gold-standard PropBank frameset tags into our maximum entropy models for these 20 verbs by partitioning the instances according to their frameset tag. Table 5 shows a breakdown of the accuracy by feature type. Adding the gold-standard frameset tag (“*fset”) to our original local features (“orig”) did not increase the accuracy significantly. However, the increase in accuracy (from 59.7% to 62.8%) was significant when these frameset tags were incorporated into the model that used both our original features and all the PropBank features.

Feature	Accuracy
orig	0.564
orig*fset	0.587
orig+pb	0.597
(orig+pb)*fset	0.628

Table 5: Accuracy of system on WordNet sense-tagging of 20 Senseval-2 verbs with more than one frameset, with and without gold-standard frameset tag.

However, partitioning the instances using the automatically generated frameset tags has no significant effect on the system’s performance; the information provided by the automatically assigned coarse-grained sense tag is already encoded in the features used for fine-grained sense-tagging.

4 Related Work

Our approach of using rich linguistic features combined in a single maximum entropy framework contrasts with that of (Florian et al., 2002). Their feature space was much like ours, but did not include semantic class features for noun complements. With this more impoverished feature set, they experimented with combining diverse classifiers to achieve an improvement of 2.1% over all parts of speech (noun, verb, adjective) in the Senseval-2 lexical sample task; however, this improvement was over an initial accuracy of 56.6% on verbs, indicating that their performance is still below ours for verbs.

(Lee and Ng, 2002) explored the relative contribution of different knowledge sources and learning algorithms to WSD; they used Support Vector Machines (SVM) and included local collocations and syntactic relations, and also found that adding syntactic features improved accuracy. Our features are similar to theirs, but we added semantic class features for the verb arguments. We found that the difference in machine learning algorithms did not play a large role in performance; when we used our features in SVM we obtained almost no difference in performance over using maximum entropy models with Gaussian priors.

(Gomez, 2001) described an algorithm using WordNet to simultaneously determine verb senses and attachments of prepositional phrases, and iden-

tify thematic roles and adjuncts; our work is different in that it is trained on *manually* annotated corpora to show the relevance of semantic roles for verb sense disambiguation.

5 Conclusion

We have shown that disambiguation of verb senses can be improved by leveraging information about predicate arguments and their semantic classes. Our system performs at the best published accuracy on the English verbs of Senseval-2 even though our heuristics for extracting syntactic features fail to identify all and only the arguments of a verb. We show that associating WordNet semantic classes with nouns is beneficial even without explicit disambiguation of the noun senses because, given enough data, maximum entropy models are able to assign high weights to the correct hypernyms of the correct noun sense if they represent defining selectional restrictions. Knowledge of gold-standard predicate-argument information from PropBank improves WSD on both coarse-grained senses (PropBank framesets) and fine-grained WordNet senses. Furthermore, partitioning instances according to their gold-standard frameset tags, which are based on differences in subcategorization frames, also improves the system's accuracy on fine-grained WordNet sense-tagging. Our experiments suggest that sense disambiguation for verbs can be improved through more accurate extraction of features representing information such as that contained in the framesets and predicate argument structures annotated in PropBank.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This paper describes research that was conducted while the first author was at the University of Pennsylvania.

References

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: A high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, July.

Radu Florian, Silviu Cucerzan, Charles Schafer, and David Yarowsky. 2002. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341.

Fernando Gomez. 2001. An algorithm for aspects of semantic interpretation using an enhanced wordnet. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Jane Grimshaw. 1990. *Argument Structure*. MIT Press, Cambridge, MA.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of Third International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands, Spain, May.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. The University of Chicago Press.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France, July.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics*, May.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

D. Yarowsky. 1995. *Three Machine Learning Algorithms for Lexical Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania Department of Computer and Information Sciences.