Towards Robust High Performance Word Sense Disambiguation of English Verbs Using Rich Linguistic Features

Jinying Chen and Martha Palmer

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

Abstract. This paper shows that our WSD system using rich linguistic features achieved high accuracy in the classification of English SENSEVAL2 verbs for both fine-grained (64.6%) and coarse-grained (73.7%) senses. We describe three specific enhancements to our treatment of rich linguistic features and present their separate and combined contributions to our system's performance. Further experiments showed that our system had robust performance on test data without high quality rich features.

1 Introduction

Word sense disambiguation (WSD) has been regarded as essential or necessary in many high-level NLP applications that require a certain degree of semantic interpretation, such as machine translation, information retrieval (IR) and question answering, *etc.* However, previous investigations into the role of WSD in IR have shown that low accuracy in WSD negated any possible performance increase from ambiguity resolution [1,2]. This suggests that improving the performance of WSD systems is crucial for applications to attain benefits from WSD.

Much effort has been aimed at the creation of sense tagged corpora that can be used to develop supervised WSD systems with high accuracy.[1] However, highly polysemous words with subtle sense distinctions still pose major challenges for automatic systems, as evidenced in SENSEVAL2 [3]. This problem seems more serious for verbs, as indicated by the relatively poorer performance achieved by the best system in the SENSEVAL2 English lexical sample task for verbs: 56.6% accuracy, in contrast with the 64.2% accuracy for all parts-of-speech [4,5]. On the other hand, disambiguating verb senses accurately is very important for lexical selection in MT. It is also helpful for information retrieval, especially for fact retrieval systems that take full-sentence queries as their input. Therefore, this paper will focus on improving the accuracy of our supervised WSD system for verbs.

We are using a linguistically rich approach for verb sense disambiguation. Linguistically rich approaches [5-9] utilize syntactic and/or semantic features, e.g., syntactic relations, selectional preferences, and semantic information of NP arguments of verbs,

[1] http://www.senseval.org/

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 933 – 944, 2005.

etc. In verb sense disambiguation, Dang and Palmer's work [5] demonstrated that their system, which achieved 59.6% accuracy (62.5% in a recent report [10]) in disambiguating the SENSEVAL2 English verbs, benefited substantially from using rich linguistic features that capture information about a verb's lexical semantics.

On the other hand, the performance of a system using rich linguistic features relies heavily on the quality of preprocessing, such as part-of-speech tagging, parsing, feature extraction and generation, *etc.* How accurate and how robust can such a system be? In particular, we are interested in the following three questions: *How much advantage can we gain from the rich-feature approach by careful extraction and treatment of the rich features? How much will a relatively poor quality of preprocessing negatively affect the system's performance? Which strategies can we adopt to alleviate these negative effects?*

To address the first question, we enhance the feature extraction and generation of our original system, which was inspired by Dang’s system[10], in three ways. First, to increase the recall of the extraction of a verb's subject, we carefully handle relative clauses, nonfinite clauses, and verbs within prepositional phrases by using linguistic knowledge and heuristics. Second, to treat semantic features of NP arguments of verbs and prepositions in a more uniform way, we incorporate a rule-based pronoun resolver and also unify the semantic features generated by WordNet [11] and by a named entity tagger. Third, we treat sentential complements of verbs in a verb-specific way. Our evaluation on the SENSEVAL2 English verbs shows that our new system achieves 64.6% accuracy, which is significantly better than the best system on English verbs in SENSEVAL2 (57.6%) and also outperforms Dang's system (62.5%). Further experiments indicate that the three enhancements are all beneficial. They each boost the system's performance by 1.0~1.2 percent and the combined gain is 2.6 percent. A similar performance improvement is achieved for coarse-grained senses: 73.7% vs. Dang's 71.7%. The data analysis of the results suggests that further improvements may come from disambiguating WordNet synsets and from using statistical methods for subject extraction and pronoun resolution.

We address the last two robustness questions in two more experiments. To investigate how the parser's performance affects our system, we divide the test data into an easy set that is similar to the parser's training material and a hard set that is not. The evaluation shows that although our system's accuracy is lower on the hard set, it is still high (62.2%). In the second experiment, our system is trained with rich features and tested on data with linguistically impoverished features. The results show little penalty from missing rich features at the test phase. The observations from this experiment also suggest the following strategy for using WSD systems that utilize rich linguistic features. When good parsers are not available at the time of application, the use of topical features and any available, accurate rich features (e.g., features associated with the verb's direct object) will alleviate penalties.

The rest of the paper is organized as follows. We introduce our system and the three major enhancements we made in Section 2. In Section 3, we show the evaluation results on SENSEVAL2 English verbs and show how much the three enhancements improve our system's performance. We then discuss the potential improvements of our system in the future. In Section 4, we investigate the robustness of our system and propose our strategy for alleviating the negative effects of poor preprocessing. We conclude our discussion in Section 5.

2 System Description

Our original WSD system was inspired by the successful MaxEnt WSD system of Dang [5,10]. We used the same machine learning model, Mallet, that implements a smoothing maximum entropy (ME) model with a Gaussian prior [12]. An attractive property of ME models is that there is no assumption of feature independence [13]. Empirical studies have shown that a ME model with a Gaussian prior generally outperforms ME models with other smoothing methods [14]. In addition to topical and collocation features, we also used similar rich syntactic and semantic features, although we implemented them in different ways. Furthermore, we enhanced the treatment of certain rich linguistic features, which we believed would boost the system's performance. Before discussing these enhancements, we first briefly describe the basic syntactic and semantic features used by our system:

Syntactic features:

1. Is the sentence passive, semi-passive[2] or active?
2. Does the target verb have a subject or object? If so, what is the head of its subject or/and object?
3. Does the target verb have a sentential complement?
4. Does the target verb have a PP adjunct? If so, what is the preposition and what is the head of the NP argument of the preposition?

Semantic features:

1. The Named Entity tags of proper nouns and certain types of common nouns
2. The WordNet synsets and hypernyms of head nouns of the NP arguments of verbs and prepositions

To better explore the advantage of using rich syntactic and semantic features, we enhanced our original system in three primary aspects: increasing the recall of the extraction of a verb's subject; unifying the treatment of semantic features of pronouns, common nouns and proper nouns; and providing a verb-specific treatment of sentential complements. These are each described in more detail below.

2.1 Increasing Subject Extraction Recall

To extract a subject, our original system simply checks the left NP siblings of the highest VP that contains the target verb and is within the innermost clause (see Figure 1). This method has high precision but low recall. Typical examples from SENSEVAL2 data that are not handled by this approach are shown in (1a-c).[3]

(1) a. **Relative clauses:** For Republicans$_{\text{sbj}}$ [$_{\text{SBAR}}$ who began$_{\text{verb}}$ this campaign with such high hopes], ...

b. **Nonfinite clauses:** I$_{\text{sbj}}$ didn't ever want [$_{\text{S}}$ to see$_{\text{verb}}$ that woman again].

c. **Verbs within PP's:** Karipo and her women$_{\text{sbj}}$ had succeeded [$_{\text{PP}}$ in driving$_{\text{verb}}$ a hundred invaders from the isle ...]

[2] Verbs that are past participles and are not preceded by *be* or *have* verbs are semi-passive.

[3] The target verb and its subject or subject candidates are underlined and the innermost clause or the PP containing the verb is bracketed.

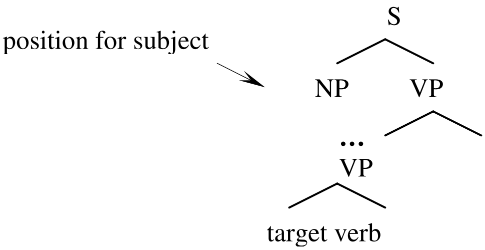
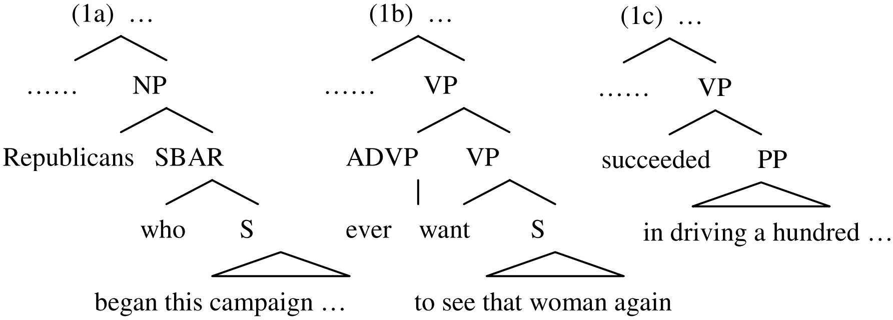


Fig. 1. position for verb's subject



To increase the recall, we refined the procedure of subject extraction by adding rules based on linguistic knowledge and bracketing labels that can handle relative clauses, nonfinite clauses, and verbs within prepositional phrases (PP's). For example, for cases like (1a), if a clause containing the target verb has a bracketing label SBAR and an NP parent, and is headed by a relative pronoun such as *that, which* or *who*, then check its left NP siblings for the verb's subject. For cases like (1b) and (1c), if the parent node of a nonfinite clause S or a PP is a VP, then continue searching positions outside the S or PP. For the last case, we also use a heuristic, i.e., a check as to whether the subject candidate is a person or an organization, to filter out non-person-and-organization candidate NPs whose parent nodes are not labeled as S or SBAR. Many cases like (2a-b) can be handled correctly using this heuristic.

(2) a. A number of accounts of the events accused the ministry$_{\text{sbj}}$ [$_{\text{PP}}$ of pulling$_{\text{verb}}$ the plug on the UAL deal ...].
b. Mr. Wolf$_{\text{sbj}}$ faces a monumental task [$_{\text{PP}}$ in pulling$_{\text{verb}}$ the company back together again].

The above rule-based approach does not handle difficult cases like (3a-b) very well.

(3) a. Freddy's instinct was [$_{\text{S}}$ to keep$_{\text{verb}}$ growing by stock mergers and small expenditure of cash ...]
b.The arrangement I had with him was [$_{\text{S}}$ to work$_{\text{verb}}$ four hours a day].

With this enhancement, our new system extracts about 35% more subjects than before.

2.2 Unifying Semantic Features

In this section we describe the changes to the use of semantic features. In order to provide a more uniform treatment for the semantic features of the NP arguments of verbs and prepositions, we first merge the semantic features associated with proper nouns and common nouns. We then extend our treatment to include pronouns by adding a pronoun resolution module.

2.2.1 Merging Semantic Features

Our system used an automatic named entity tagger, *IdentiFinder*^{TM} [15], to tag proper nouns with **Person, Organization** and **Location** and common nouns with **Date, Time, Percent** and **Money**. Additional semantic features are all WordNet synsets and hypernyms[4] of the head nouns of NP arguments, i.e., the system does not disambiguate different WordNet senses of a head noun.

To utilize semantic features more efficiently, we refine their treatment. Previously there was no overlap between semantic features generated by the named entity tagger and by WordNet. For example, a personal proper noun only has a **Person** tag that has no similarity to the WordNet synsets and hypernyms associated with similar common nouns such as *specialist* and *doctor*, *etc.* This is likely to be a problem for many WSD tasks that usually have small amounts of training data, such as SENSEVAL2. To overcome this problem, our new system associates a common noun (or a noun phrase) with each Named Entity tag (see 4) and adds the WordNet semantic features of these nouns (or noun phrases) to the original semantic feature set.

(4) Person – someone, Organization – organization, Location – location
Time – time unit, Date – time period, Percent – percent, Money – money

2.2.2 Adding Pronoun Resolution

Our original system has no special treatment for pronouns, although a rough count shows that about half of the training instances contain pronominal arguments. Lacking a high performance automatic pronoun resolution module, we adopt a hybrid approach. For personal pronouns, we simply treat them as personal proper nouns. For the rest of the pronouns including *they, them, it, themselves* and *itself*, which occur in about 13% of the training instances, we programmed a rather simple rule-based pronoun resolver. In brief, the resolver searches the parse tree for antecedent candidates similarly to Hobb's algorithm as exemplified in [16] and uses several syntactic and semantic constraints to filter out impossible candidates. The constraints include syntactic constraints for anaphora antecedents [16], number agreement, and whether the candidate is a person. The first candidate that survives the filtering is regarded as the antecedent of the pronoun and its semantic features are added to the original feature set.

2.3 Verb-Specific Sentential Complements

The different types of sentential complements can be very useful for distinguishing certain verb senses. (5a-b) shows two sentences containing the verb *call* in the SENSEVAL2 training data. *Call* has WordNet Sense 1 (name) in (5a) and Sense 3

[4] A unique number defined in WordNet represents each synset or hypernym.

(ascribe) in (5b). In both cases, *call* takes a small clause as its sentential complement, i.e., it has the subcategorization frame X *call* Y Z. The difference is that Z is a Named Entity when *call* is in Sense 1, and Z is usually a common NP or an adjective phrase (ADJP) when *call* is in Sense 3.

(5) a. The slender, handsome fellow was $\underline{\text{called}_{\text{verb}}}$ [$_S$ Dandy Brandon].
b.The White House is purposely not $\underline{\text{calling}_{\text{verb}}}$ [$_S$ the meeting a summit] …

Another example is shown in (6). The verb *keep* has WordNet Sense 1 (maintain) in (6a) and Sense 2 (continue) in (6b). In Sense 1, *keep* often takes a small clause and has the subcategorization frame X *keep* Y ADJP. In contrast, *keep* takes a sentential complement the head verb of which is in the present tense when it is in Sense 2.

(6) a. He shook his head, $\underline{\text{kept}_{\text{verb}}}$ [$_S$ his face expressionless].
b. We $\underline{\text{keep}_{\text{verb}}}$ [$_S$ wondering what Mr. Gates wanted to say].

Our original system uses a single feature *hasSent* to represent whether the target verb has a sentential complement or not, which cannot capture the rich information that is crucial to distinguishing certain verb senses but is deeply embedded in the sentential complements, as described above. Therefore, we treat sentential complements in a more fine-grained, verb-specific way. We resort to WordNet and PropBank [17] for the information about verb subcategorization frames. Another advantage of this verb-specific treatment is that it can filter out illegal sentential complements generated by the parser.

3 System Evaluation

Since the more recent SENSEVAL3 data were collected over the internet and had a relatively low quality of annotation, we decided to evaluate our new system on the SENSEVAL2 English verbs. Ratnaparkhi's MaxEnt sentence boundary detector and POS tagger [18], Bikel's parsing engine [19], and a named entity tagger, *IdentiFinder^{TM}* [15], were used to preprocess the training and test data automatically.

3.1 Experimental Results

Table 1 shows the performance of our system (MX-RF) on the 29 verbs with fine-grained WordNet senses. Columns 2 and 3 show the number of senses and normalized sense perplexity[5] for each verb in the test data respectively. It also gives the performance of the best system on English verbs in SENSEVAL2, KUNLP [5], and Dang's system [10]. As we see, our system achieves an average accuracy of 64.6%, which is significantly better than KUNLP (57.6%) that only uses linguistically impoverished features (topical and collocation features). Our system also outperforms Dang's system (62.5%). Recall that the types of rich linguistic features used by our system were originally inspired by Dang’s system, although we implemented them in different ways. Therefore, we attribute the more success of our new system mainly to the three

[5] It is calculated as the entropy of the sense distribution of a verb in the test data divided by the largest possible entropy, i.e., $\log_2$ (the number of senses of the verb in the test data).

specific enhancements we made. To our best knowledge, the accuracy our system achieved is the best result for this task at present.

To investigate exactly how much we gain by enhancing the system in the three ways discussed in Section 2, we tested our system by removing our refinements (subject extraction, pronoun coreferences, and verb-specific sentential complements) separately and all together. The results (columns 8-11) show that each refinement boosts the system's performance by 1.0~1.2 percent and that together they achieve an improvement of 2.6 percent. This confirms the utility of these enhancements.

In addition to fine-grained verb senses, we also evaluated our system on coarse-grained senses (see Table 2). Previous work [20] suggested that not all NLP applications need fine-grained sense distinctions; in some cases coarser granularities will suffice. Furthermore, it has been demonstrated that annotation with coarser senses is much faster and more accurate [21]. The SENSEVAL2 verb senses have been grouped by using both syntactic and semantic criteria, with a resulting inter-annotator agreement (ITA) of 82% (column 4). As we expected, the accuracy of our system increases by about 9 percent on the coarse-grained senses to 73.7%, which again consistently outperforms Dang's system (71.7%).

3.2 Discussion

Compared verb-by-verb, the performance of our system is better than or comparable to Dang's on most verbs, except that it has notably lower accuracy on *develop, dress* and *serve*. It is not obvious why, since although our features are similar to Dang's, the implementations are different. Nevertheless, an investigation of the specific features our system generated for these three verbs gives us a few clues. The semantic categories of the direct objects of the three verbs are very diverse, so there are not enough instances of similar categories for the model to generalize. Therefore, the system performance benefits little from our enhancements. In fact, our system may be more susceptible to noisy data introduced by the pronoun resolver for these three verbs. Erroneous antecedents found by the resolver are indistinguishable from the actual direct objects that occur rarely in the training data, and therefore they get the same treatment from the machine learning algorithm.

The experimental results and the above data analysis suggest that our system can be improved further by increasing the accuracy of subject extraction and pronoun resolution. We expect a state-of-the-art pronoun resolution module and a statistical subject finder to do better jobs in the future. Our current system does not distinguish senses of nouns when using WordNet synsets and hypernyms as semantic features, which introduces many irrelevant features (associated with the irrelevant senses). The machine learning algorithm sometimes cannot generalize well using these features. A potential solution for this problem is to distinguish the senses of the target verb and its NP arguments simultaneously. Furthermore, we need to have a better generalization, or clustering, of WordNet synsets and hypernyms, especially when the subject or object of a verb has semantic versatility. More performance improvements will bring us closer to our goal of an overall level of accuracy of 80%, especially with respect to coarse-grained senses, that should finally be more beneficial to NLP applications.

Table 1. Evaluation of MX-RF on the SENSEVAL2 English verbs, with fine-grained senses

Verb	#of Sen	Sen Per-plex.	ITA	KUNLP	Dang 2004	MX-RF	MX-RF w/o sbj extract.	MX-RF w/o pron.	MX-RF w/o verb spec sent-comp	MX-RF w/o all three
begin	7	0.63	81.2	81.4	89.3	91.2	90.0	90.4	89.3	88.6
call	17	0.86	69.3	48.5	54.5	56.8	56.8	55.3	53.8	52.3
carry	19	0.87	60.7	45.5	39.4	44.7	45.5	40.2	43.2	42.4
collaborate	2	0.47	75.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0
develop	14	0.82	67.8	42.0	58.0	49.3	49.3	50.7	49.3	49.3
draw	21	0.95	76.7	34.1	31.7	41.5	39.0	34.1	41.5	36.6
dress	12	0.79	86.5	71.2	72.9	64.4	64.4	69.5	67.8	64.4
drift	9	0.89	50.0	53.1	40.6	67.2	51.6	60.9	64.1	48.4
drive	13	0.84	58.8	54.8	59.5	60.7	60.7	58.3	60.7	58.3
face	6	0.38	78.6	82.8	83.9	81.2	82.3	83.3	81.2	83.3
ferret	0	0.00	1.00	100.0	100.0	100.0	100.0	100.0	100.0	100.0
find	17	0.94	44.3	27.9	36.8	41.2	36.8	36.8	36.8	33.8
keep	20	0.79	79.1	44.8	61.2	64.2	61.9	65.7	61.2	57.5
leave	10	0.86	67.2	50.0	60.6	57.6	57.6	54.5	53.0	50.0
live	9	0.70	79.7	59.7	70.1	69.4	69.4	67.9	69.4	67.9
match	7	0.79	56.5	52.4	50.0	59.5	61.9	57.1	59.5	57.1
play	20	0.85	N/A	37.9	53.0	62.1	59.1	62.1	62.1	62.1
pull	25	0.89	68.1	45.0	50.0	58.3	56.6	58.3	53.3	56.7
replace	4	0.85	65.9	55.6	60.0	61.1	60.0	55.5	61.1	57.8
see	13	0.84	70.9	39.1	39.1	44.2	39.9	42.8	41.3	35.5
serve	11	0.85	90.8	68.6	74.5	68.6	66.7	64.7	68.6	66.7
strike	20	0.89	76.2	40.7	38.9	51.9	50.0	53.7	51.9	55.6
train	8	0.87	28.8	58.7	63.5	60.3	60.3	63.5	60.3	63.5
treat	5	0.88	96.9	56.8	50.0	50.0	50.0	52.3	50.0	56.8
turn	26	0.93	74.2	37.3	49.3	48.5	44.0	47.8	47.0	46.3
use	6	0.65	74.3	65.8	71.1	69.7	72.4	68.4	69.7	68.4
wander	5	0.47	65.0	82.0	80.0	82.0	82.0	82.0	82.0	82.0
wash	7	0.94	87.5	83.3	66.7	75.0	75.0	75.0	75.0	75.0
work	18	0.84	N/A	45.0	45.0	53.3	51.7	50.0	53.3	43.3
average	12	0.77	71.3	57.6	62.5	64.6	63.4	63.6	63.4	62.0

Table 2. Evaluation of MX-RF on coarse-grained senses of the SENSEVAL2 English verbs

	# of grp	ITA grp	Acc. of Dang 2004	Acc. of MX-RF
Ave. on 29 verbs	5.9	82.0	71.7	73.7

4 System Robustness

A frequent criticism of systems using rich linguistic features is that they do not port well to domains for which accurate preprocessors are not available. In this section we discuss two experiments designed to address the following two questions: How much will a relatively poor quality of preprocessing negatively affect the system's performance? Which strategies can we adopt to alleviate these negative effects?

4.1 Experiment I

Since the parser is the most critical component of our preprocessing and is more likely to have lower performance when it is used in an unfamiliar data set, we investigate how the performance of the parser on different test data sets affects our system. We divided the SENSEVAL2 test data into two sets: an easy set and a hard set. The test data from the Wall Street Journal (wsj) sections of Penn Treebank (PTB) [22] are put into the easy set because they are similar to the parser's training data: 02-21 wsj sections. The hard set contains test data from the Brown sections of PTB and BNC data. It is expected that the parser and therefore the system will perform better on the easy set. We trained our system on the whole SENSEVAL2 training data set and evaluated its performance on the easy and hard test sets separately. The results are shown in Table 3.

Table 3. Performance on different test data sets

Test data set	Hard	Easy	Whole Set
Num. of test inst.	**895**	**911**	**1806**
Average Acc.	**62.2**	**66.9**	**64.6**

As we expected, the system's performance on the hard test set is 4.7 percent lower than on the easy set. On the other hand, even on the hard set, its accuracy (62.2%) is still high and is comparable to Dang's system. It is worth noting that the experiment is preliminary because the easy set and the hard set are most likely to be different not only on whether they are familiar to the parser but also on the subtlety and distributions of their senses. Nevertheless, it is evidence of our system's robustness.

4.2 Experiment I I

There will be situations where systems trained with rich linguistic features extracted from high quality parses will be run on applications where such rich features will not be available. It is most likely that systems in such situations will go back to a position similar to where rich features are not available in both the training and test phases. However, could things get even worse? A machine learning model often tends to favor informative features (e.g., rich linguistic features in our case) and fit the distribution of these features well in its training phase. Therefore, it is expected that the model will be penalized more heavily when these informative features are used in its training phase but are not accessible in its test phase. In this subsection, we discuss a second experiment to test the robustness of our system in such situations and explore possible strategies for alleviating penalties.

We trained our system with rich features of the SENSEVAL2 training data and tested its performance on the SENSEVAL2 test data with three different feature sets: a rich set containing topical, collocation, syntactic and semantic features (top+col+syn+sem), a poor set containing topical and collocation features (top+col) and a medium set containing topical and collocation features plus features for direct objects (top+col+obj). The reason we include the medium set is that a parser can usually find the direct object of verbs. Furthermore, we trained and tested our system on SENSEVAL2 data with linguistically impoverished features (top+col) and used this result as a control. As shown in Table 4, the system's accuracy drops to the same level as the control (58.0% vs. 58.1%) when it is trained with rich features but tested with poor features. When the features associated with the verb's direct object are added, the system's performance improves (59.1%).

The experimental results here suggest that our system has not been penalized very much when rich linguistic features are only available in its training phase. Intuitively, the topical features[6] our system uses alleviate the penalty. As expected, when the topical features of the test data were excluded, the performance of our system dropped to 54.8%. But this will be a common problem for all systems using topical features, not only for systems using rich linguistic features.[7] These results suggest a strategy for using our system and other similar systems in a more robust way. When a state-of-art parser is not available for the application data, topical features can be used to alleviate the penalty. Rich features that can be obtained more easily and reliably, e.g., features associated with the direct object of verbs, can also be used whenever they are available.

Table 4. Performance of our system trained and tested on data sets with different features

Training set **Test set**	**top+col+syn+sem**	**top+col**
top+col+syn+sem	**64.6**	
top+col	**58.0**	**58.1**
top+col+obj	**59.1**	

5 Conclusion

We have shown that our system using rich linguistic features was more successful, compared with the previous best systems, in classifying the fine-grained and coarse-grained SENSEVAL2 verb senses. The three enhancements to the system's treatment

[6] Our system uses all the contextual nouns, verbs, adjectives and adverbs that are not in a stop word list as topical features.

[7] In fact, the performance of our system trained with (top+col) features and tested with only collocation features also dropped to 55.8%, in contrast to the control accuracy 58.1%.

of rich linguistic features were beneficial. Further improvements may come from disambiguating WordNet synsets and improving the accuracy of subject extraction and pronoun resolution. Furthermore, our system was robust when it was applied to test data that had a relatively poor quality of rich features. Based on the experimental results, we proposed a strategy for using systems with rich features in a more robust way. Our goal is to continue to improve the performance of our current WSD system, with respect to both fine-grained and coarse-grained senses, so that it becomes increasingly beneficial to NLP applications.

References

1. Mark Sanderson: Word sense disambiguation and information retrieval. In Proceedings of the 17th Int. ACM SIGIR, Dublin, IE (1994).
2. Christopher Stokoe, Michael P. Oakes, John Tait: Word sense disambiguation and information retrieval revisited. In Proceedings of the 26th annual int. ACM SIGIR conference on research and development in information retrieval. Toronto, Canada (2003).
3. Philip Edmonds and Scott Cotton: SENSEVAL-2: Overview. In Proceedings of SENSEVAL-2: 2nd Int. Workshop on Evaluating WSD Systems. ACL-SIGLEX, Toulouse, France (2001).
4. David Yarowsky, Silviu Cucerzan, Radu Florian, Charles Schafer and Richard Wicentowski: The Johns hopkins SENSEVAL2 system description. In Proceedings of SENSEVAL-2: 2nd Int.Workshop on Evaluating WSD Systems. Toulouse France (2001).
5. Hoa T. Dang and Martha Palmer: Combining contextual features for word sense disambiguation. In Proceedings of the SIGLEX/SENSEVAL Workshop on WSD: Recent Successes and Future Directions, in conjunction with ACL-02, Philadelphia (2002).
6. Martínez David, Agirre Enek. and Màrquez Liuis: Syntactic Features for High Precision Word Sense Disambiguation. In Proceedings of the 19th International COLING. Taipei (2002).
7. Dekang Lin: Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity In Proceedings of ACL-97, Madrid, Spain (1997).
8. Yoong Keok Lee and Hwee Tou Ng: 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 (EMNLP) (2002) pages 41–48.
9. Rada Mihalcea and Ehsanul Faruque: Sense Learner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text. In Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004).
10. Hoa T. Dang: Investigations into the role of lexical semantics in word sense disambiguation. PhD Thesis. University of Pennsylvania (2004).
11. Christiane Fellbaum: WordNet - an Electronic Lexical Database. The MIT Press, Cambridge, Massachusetts, London, UK (1998).
12. Andrew K. McCallum: MALLET: A Machine Learning for Language Toolkit. http://www.cs. umass.edu/~mccallum/mallet (2002).
13. Adam L. Berger, Stephen A. Della Piertra, and Vincent J. Della Pietra: A maximum entropy approach to natural language processing. Compuational Linguistics, (1996) 22(1): 39-71.
14. Stanley. F. Chen and Ronald Rosenfeld: A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, CMU (1999).

15. Daniel M. Bikel, Richard Schwartz and Ralph M. Weischedel: An algorithm that learns what's in a name. Machine Learning, (1999) 34(1-3). Special Issue on Natural Language Learning.
16. Shalom Lappin and Herbert Leass: An algorithm for pronominal anaphora resolution. Computational Linguistics, (1994) 20(4): 535-561.
17. Paul Kingsbury, Martha Palmer, and Mitch Marcus: Adding semantic annotation to the Penn Tree-Bank. In Proceedings of HLT 2002, San Diego, CA (2002).
18. Adwait Ratnaparkhi: Maximum entropy models for natural language ambiguity resolution. Ph.D. thesis, University of Pennsylvania (1998).
19. 19 Daniel M. Bikel: Design of a multi-lingual, parallel-processing statistical parsing engine.In Proceedings of HLT 2002. San Diego, CA (2002).
20. Paul Buitelaar: Reducing lexical semantic complexity with systematic polysemous classes and underspecification. In Poceedings of the ANLP Workshop on Syntactic and Semantic Complexity in NLP Systems. Seattle, WA (2000).
21. Martha Palmer, Olga B. Malaya and Hoa T. Dang: Different sense granularities for different appli-cations. In Proceedings of HLT/NAACL-04. Boston (2004).
22. Mitchell Marcus, Grace Kim, Mary A. Marcinkiewicz, Robert MacIntyre, Mark Ferguson, Karen Katz and Britta Schasberger: The Penn Treebank: annotating predicate argument structure. In Proceedings of the ARPA'94 HLT Workshop (1994).