Towards Robust Semantic Role Labeling

Sameer S. Pradhan*
BBN Technologies

Wayne Ward**
University of Colorado

James H. Martin†
University of Colorado

Most semantic role labeling (SRL) research has been focused on training and evaluating on the same corpus. This strategy, although appropriate for initiating research, can lead to over-training to the particular corpus. This article describes the operation of ASSERT, *a state-of-the art SRL system, and analyzes the robustness of the system when trained on one genre of data and used to label a different genre. As a starting point, results are first presented for training and testing the system on the PropBank corpus, which is annotated Wall Street Journal (WSJ) data. Experiments are then presented to evaluate the portability of the system to another source of data. These experiments are based on comparisons of performance using PropBanked WSJ data and PropBanked Brown Corpus data. The results indicate that whereas syntactic parses and argument identification transfer relatively well to a new corpus, argument classification does not. An analysis of the reasons for this is presented and these generally point to the nature of the more lexical/semantic features dominating the classification task where more general structural features are dominant in the argument identification task.*

1. Introduction

Automatic, accurate, and wide-coverage techniques that can annotate naturally occurring text with semantic structure can play a key role in NLP applications such as information extraction (Harabagiu, Bejan, and Morarescu 2005), question answering (Narayanan and Harabagiu 2004), and summarization. Semantic role labeling (SRL) is one method for producing such semantic structure. When presented with a sentence, a semantic role labeler should, for each predicate in the sentence, first identify and then label its semantic arguments. This process entails identifying groups of words in a sentence that represent these semantic arguments and assigning specific labels to them. In the bulk of recent work, this problem has been cast as a problem in supervised machine learning. Using these techniques with hand-corrected syntactic parses, it has

* Department of Speech and Language Processing, 10 Moulton Street, Room 2/245, Cambridge, MA 02138. E-mail: sameer@cemantix.org.
** The Center for Spoken Language Research, Campus Box 594, Boulder, CO 80309. E-mail: whw@colorado.edu.
† The Center for Spoken Language Research, Campus Box 594, Boulder, CO 80309. E-mail: martin@colorado.edu.

Submission received: 15 July 2006; revised submission received: 3 May 2007; accepted for publication: 19 June 2007.

been possible to achieve accuracies within the range of human inter-annotator agreement. More recent approaches have involved using improved features such as *n*-best parses (Koomen et al. 2005; Toutanova, Haghighi, and Manning 2005); exploiting argument interdependence (Jiang, Li, and Ng 2005); using information from fundamentally different, and complementary syntactic, views (Pradhan, Ward et al. 2005); combining hypotheses from different labeling systems using inference (Màrquez et al. 2005); as well as applying novel learning paradigms (Punyakanok et al. 2005; Toutanova, Haghighi, and Manning 2005; Moschitti 2006) that try to capture more sequence and contextual information. Some have also tried to jointly decode the syntactic and semantic structures (Yi and Palmer 2005; Musillo and Merlo 2006). This problem has also been the subject of two CoNLL shared tasks (Carreras and Màrquez 2004; Carreras and Màrquez 2005). Although all of these systems perform quite well on the standard test data, they show significant performance degradation when applied to test data drawn from a genre different from the data on which the system was trained. The focus of this article is to present results from an examination into the primary causes of the lack of portability across genres of data.

To set the stage for these experiments we first describe the operation of ASSERT, our state-of-the art SRL system. Results are presented for training and testing the system on the PropBank corpus, which is annotated *Wall Street Journal* (WSJ) data.

Experiments are then presented to assess the portability of the system to another genre of data. These experiments are based on comparisons of performance using PropBanked WSJ data and PropBanked Brown corpus data. The results indicate that whereas syntactic parses and identification of the argument bearing nodes transfer relatively well to a new corpus, role classification does not. Analysis of the reasons for this generally point to the nature of the more lexical/semantic features dominating the classification task, as opposed to the more structural features that are relied upon for identifying which constituents are associated with arguments.

2. Semantic Annotation and Corpora

In this article, we report on the task of reproducing the semantic labeling scheme used by the PropBank corpus (Palmer, Gildea, and Kingsbury 2005). PropBank is a 300k-word corpus in which predicate argument relations are marked for almost all occurrences of non-copula verbs in the WSJ part of the Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993). PropBank uses predicate independent labels that are sequential from ARG0 to ARG5, where ARG0 is the PROTO-AGENT (usually the subject of a transitive verb) and ARG1 is the PROTO-PATIENT (usually its direct object). In addition to these **core arguments**, additional **adjunctive arguments**, referred to as ARGMs, are also marked. Some examples are ARGM-LOC, for locatives, and ARGM-TMP, for temporals. Table 1 shows the argument labels associated with the predicate *operate* in PropBank.

Following is an example structure extracted from the PropBank corpus. The syntax tree representation along with the argument labels is shown in Figure 1.

[$_{\textbf{ARG0}}$ It] [$_{\textbf{predicate}}$ operates] [$_{\textbf{ARG1}}$ stores] [$_{\textbf{ARGM-LOC}}$ mostly in Iowa and Nebraska].

The PropBank annotation scheme assumes that a semantic argument of a predicate aligns with one or more nodes in the hand-corrected Treebank parses. Although most frequently the arguments are identified by one node in the tree, there can be cases where the arguments are discontinuous and more than one node is required to identify parts of the arguments.

Table 1
Argument labels associated with the predicate *operate* (sense: work) in the PropBank corpus.

Tag	Description
ARG0	Agent, operator
ARG1	Thing operated
ARG2	Explicit patient (thing operated on)
ARG3	Explicit argument
ARG4	Explicit instrument

Treebank trees can also have **trace** nodes which refer to another node in the tree, but do not have any words associated with them. These can also be marked as arguments. As traces are typically not reproduced by current automatic parsers, we decided not to consider them in our experiments—whether or not they represent arguments of a predicate. None of the previous work has attempted to recover such trace arguments. PropBank also contains arguments that are coreferential.

We treat discontinuous and coreferential arguments in accordance to the CoNLL shared task on semantic role labeling. The first part of a discontinuous argument is labeled as it is, and the second part of the argument is labeled with a prefix "C-" appended to it. All coreferential arguments are labeled with a prefix "R-" appended.

We follow the standard convention of using Section 02 to Section 21 as the training set, Section 00 as the development set, and Section 23 as the test set. The training set comprises about 90,000 predicates instantiating about 250,000 arguments and the test set comprises about 5,000 predicates instantiating about 12,000 arguments.

3. Task Description

In ASSERT, the task of semantic role labeling is implemented by assigning role labels to constituents of a syntactic parse. Parts of the overall process can be analyzed as three different tasks as introduced by Gildea and Jurafsky (2002):

1. *Argument Identification*—This is the process of identifying parsed constituents in the sentence that represent semantic arguments of

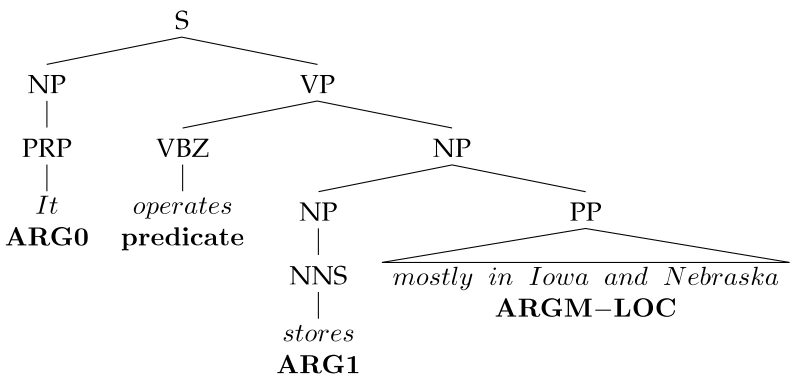

Figure 1
Syntax tree for a sentence illustrating the PropBank tags.

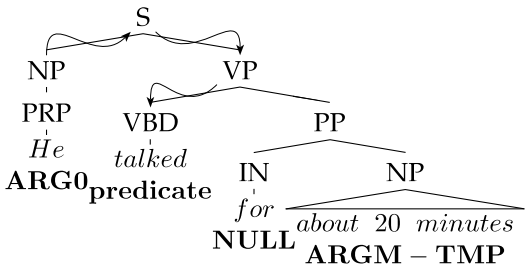

Figure 2
Syntax tree for a sentence illustrating the PropBank arguments.

a given predicate. Each node in a parse tree can be classified (with respect to a given predicate) as either one that represents a semantic argument (i.e., a NON-NULL node) or one that does not represent any semantic argument (i.e., a NULL node).

2. *Argument Classification*—Given constituents known to represent arguments of a predicate, this process assigns the appropriate argument labels to them.

3. *Argument Identification and Classification*—A combination of the two tasks.

For example, in the tree shown in Figure 2, the node IN that dominates *for* is a NULL node because it does not correspond to a semantic argument. The node NP that dominates *about 20 minutes* is a NON-NULL node, because it does correspond to a semantic argument—ARGM-TMP.

4. ASSERT (Automatic Statistical SEmantic Role Tagger)

4.1 System Architecture

ASSERT[1] produces a separate set of semantic role labels for each candidate predicate in a sentence. Because PropBank only annotates arguments for non-copula/non-auxiliary verbs, those are also the predicates considered by ASSERT. ASSERT performs constituent-based role assignment. The basic inputs are a sentence and a syntactic parse of the sentence. For each constituent in the parse tree, the system extracts a set of features and uses a classifier to assign a label to the constituent. The set of labels used are the PropBank argument labels plus NULL, which means no argument is assigned to that constituent for the predicate under consideration.

Support vector machines (SVMs) (Burges 1998; Vapnik 1998) have been shown to perform well on text classification tasks, where data is represented in a high dimensional space using sparse feature vectors (Joachims 1998; Kudo and Matsumoto 2000; Lodhi et al. 2002). We formulate the problem as a multi-class classification problem using an SVM classifier. We employ a ONE *vs* ALL (OVA) approach to train n classifiers for a multi-class problem. The classifiers are trained to discriminate between examples

1 www.cemantix.org/assert.

of each class, and those belonging to all other classes combined. During testing, the classifier scores on an example are combined to predict its class label.

ASSERT was developed using TinySVM[2] along with YamCha[3] (Kudo and Matsumoto 2000, 2001) as the SVM training and classification software. The system uses a polynomial kernel with degree 2; the cost per unit violation of the margin, $C = 1$; and, tolerance of the termination criterion, $e = 0.001$. SVMs output distances from the classification hyperplane, not probabilities. These distances may not be comparable across classifiers, especially if different features are used to train each binary classifier. These raw SVM scores are converted to probabilities by fitting to a sigmoid function as done by Platt (2000).

The architecture just described has the drawback that each argument classification is made independently, without considering other arguments assigned to the same predicate. This ignores a potentially important source of information: that a predicate is likely to instantiate a certain set of arguments. To represent this information, a backed-off trigram model is trained for the argument sequences. In this model, the predicate is considered as an argument and is part of the sequence. This model represents not only what arguments a predicate is likely to take, but also the probability of a given sequence of arguments. During the classification process the system generates an argument lattice using the n-best hypotheses for each node in the syntax tree. A Viterbi search through the lattice uses the probabilities assigned by the sigmoid as the observation probabilities, along with the argument sequence language model probabilities, to find the maximum likelihood path such that each node is either assigned a value belonging to the PropBank arguments, or NULL. The search is also constrained so that no two nodes that overlap are both assigned NON-NULL labels.

4.2 Features

The feature set used in ASSERT is a combination of features described in Gildea and Jurafsky (2002) as well as those introduced in Pradhan et al. (2004), Surdeanu et al. (2003), and the *syntactic-frame* feature proposed in (Xue and Palmer 2004). Following is the list of features used.

4.2.1 Predicate. This is the predicate whose arguments are being identified. The surface form as well as the lemma are added as features.

4.2.2 Path. The syntactic path through the parse tree from the parse constituent to the predicate being classified.

For example, in Figure 3, the path from ARG0 (*The lawyers*) to the predicate *went* is represented with the string NP↑S↓VP↓VBD. ↑ and ↓ represent upward and downward movement in the tree, respectively.

4.2.3 Phrase Type. Syntactic category (NP, PP, etc.) of the constituent.

4.2.4 Position. Whether the constituent is before or after the predicate.

2 www.chasen.org/~taku/software/TinySVM/.
3 www.chasen.org/~taku/software/YamCha/.

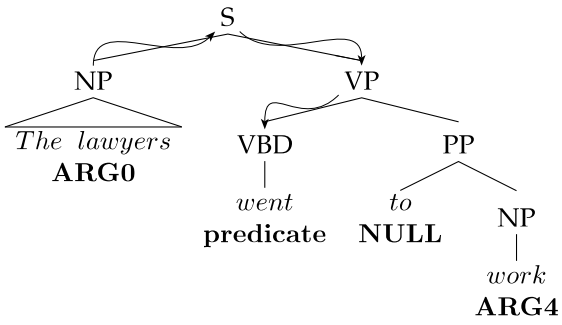

Figure 3
Illustration of path NP↑S↓VP↓VBD.

4.2.5 Voice. Whether the predicate is realized as an active or passive construction. A set of hand-written `tgrep` expressions operating on the syntax tree is used to identify passives.

4.2.6 SubCategorization. This is the phrase structure rule expanding the predicate's parent node in the parse tree. For example, in Figure 3, the subcategorization for the predicate "went" is VP→VBD-PP-NP.

4.2.7 Predicate Cluster. The distance function used for clustering is based on the intuition that verbs with similar semantics will tend to have similar direct objects. For example, verbs such as *eat*, *devour*, and *savor* will tend to all occur with direct objects describing food. The clustering algorithm uses a database of verb–direct-object relations extracted by Lin (1998). The verbs were clustered into 64 classes using the probabilistic co-occurrence model of Hofmann and Puzicha (1998). We then use the verb class of the current predicate as a feature.

4.2.8 Head Word. Syntactic head of the constituent.

4.2.9 Head Word POS. Part of speech of the head word.

4.2.10 Named Entities in Constituents. Binary features for seven named entities (PERSON, ORGANIZATION, LOCATION, PERCENT, MONEY, TIME, DATE) tagged by IdentiFinder (Bikel, Schwartz, and Weischedel 1999).

4.2.11 Path Generalizations.

1. **Partial Path—**Path from the constituent to the lowest common ancestor of the predicate and the constituent.
2. **Clause-based path variations—**Position of the clause node (S, SBAR) seems to be an important feature in argument identification (Hacioglu et al. 2004). Therefore we experimented with four clause-based path feature variations.
   (a) Replacing all the nodes in a path other than clause nodes with an asterisk. For example, the path NP↑S↑VP↑SBAR↑NP↑VP↓VBD becomes NP↑S↑*S↑*↑*↓VBD.

(b) Retaining only the clause nodes in the path, which for the given example would produce NP↑S↑S↓VBD.
(c) Adding a binary feature that indicates whether the constituent is in the same clause as the predicate.
(d) Collapsing the nodes between S nodes, which gives NP↑S↑NP↑VP↓VBD.

3. **Path n-grams—**This feature decomposes a path into a series of trigrams. For example, the path NP↑S↑VP↑SBAR↑NP↑VP↓VBD becomes: NP↑S↑VP, S↑VP↑SBAR, VP↑SBAR↑NP, SBAR↑NP↑VP, and so on. Shorter paths were padded with nulls.

4. **Single character phrase tags—**Each phrase category is clustered to a category defined by the first character of the phrase label.

4.2.12 Predicate Context. We added the predicate context to capture predicate sense variations. Two words before and two words after were added as features. The POS of the words were also added as features.

4.2.13 Punctuation. Punctuation plays an particularly important role for some adjunctive arguments, so punctuation on the left and right of the constituent are included as features. The absence of punctuation in either location was indicated with a NULL feature value.

4.2.14 Head Word of PP. Many adjunctive arguments, such as temporals and locatives, occur as prepositional phrases in a sentence, and it is often the case that the head words of those phrases, which are prepositions, are not very discriminative; for example, *in the city* and *in a few minutes* both share the same head word *in* and neither contain a named entity, but the former is ARGM-LOC, whereas the latter is ARGM-TMP. The head word of the first noun phrase inside the prepositional phrase is used for this feature. Preposition information is represented by appending it to the phrase type, for example, "PP-in" instead of "PP."

4.2.15 First and Last Word/POS in Constituent. The first and last words in a constituent along with their parts of speech.

4.2.16 Ordinal Constituent Position. In order to avoid false positives where constituents far away from the predicate are spuriously identified as arguments, we added this feature which is a concatenation of the constituent type and its ordinal position from the predicate.

4.2.17 Constituent Tree Distance. This is a more fine-grained way of specifying the already present position feature. This is the number of constituents that are encountered in the path from the predicate to the constituent under consideration.

4.2.18 Constituent Relative Features. These are nine features representing the phrase type, head word, and head word part of speech of the parent, and left and right siblings of the constituent.

4.2.19 Temporal Cue Words. There are several temporal cue words that are not captured by the named entity tagger and were added as binary features indicating their presence.

The BOW toolkit was used to identify words and bigrams that had highest average mutual information with the ARGM-TMP argument class.

4.2.20 Syntactic Frame. Sometimes there are multiple children under a constituent having the same phrase type, and one or both of them represent arguments of the predicate. In such situations, the path feature is not very good at discriminating between them, and the position feature is also not very useful. To overcome this limitation, Xue and Palmer (2004) proposed a feature which they call the **syntactic frame**. For example, if the subcategorization for the predicate is VP→VBD-NP-NP, then the syntactic frame feature for the first NP in the sequence would be, "vbd_NP_np," and for the second it would be "vbd_np_NP."

4.3 Performance

Table 2 illustrates the performance of the system using Treebank parses and using parses produced by a Charniak parser (Automatic). Precision (P), Recall (R), and F-scores are given for the identification and combined tasks, and Classification Accuracy (A) for the classification task. Classification performance using Charniak parses is only 1% absolute worse than when using Treebank parses. On the other hand, argument identification performance using Charniak parses is 10.9% absolute worse. About half of the ID errors are due to missing constituents in the Charniak parse. Techniques to address the issue of constituents missing from the syntactic parse tree are reported in Pradhan, Ward et al. (2005).

4.4 Feature Salience

In Pradhan, Hacioglu et al. (2005) we reported on a series of experiments to show the relative importance of features to the Identification task and the Classification task. The data show that different features are more salient for each of the two tasks. For the Identification task, the most salient features are the Path and Partial Path. The Predicate was not particularly salient. For Classification, the most salient features are Head Word, First Word, and Last Word of a constituent as well as the Predicate itself. For Classification, the Path and Phrase Type features were not very salient.

A reasonable conclusion is that structural features dominate the Identification task, whereas more specific lexical or semantic features are important for Classification. As

Table 2
Performance of ASSERT on WSJ test set (Section 23) using correct Treebank parses as well as Charniak parses.

Parse	Task	P (%)	R (%)	F	A (%)
Treebank	Id.	97.5	96.1	96.8	
	Class.	–	–	–	93.0
	Id. + Class.	91.8	90.5	91.2	
Automatic	Id.	87.8	84.1	85.9	
	Class.	–	–	–	92.0
	Id. + Class.	81.7	78.4	80.0	

we'll see later, this pattern has critical implications for the portability of these features across genres.

5. Robustness to Genre of Data

Most work on SRL systems has been focused on improving the labeling performance on a test set belonging to the same genre of text as the training set. Both the Treebank on which the syntactic parser is trained, and the PropBank on which the SRL systems are trained represent articles from the year 1989 of the *Wall Street Journal*. Improvements to the system may reflect tuning to the specific data set rather than real progress. For this technology to be widely accepted it is critical that it perform reasonably well on text with styles different from the training data. The availability of PropBank annotation for another corpus of a very different style than WSJ makes it possible to evaluate the portability of SRL techniques, and to understand some of the factors affecting performance.

5.1 The Brown Corpus

The Brown Corpus is a standard corpus of American English that consists of about one million words of English text printed in the calendar year 1961 (Kučera and Francis 1967). The corpus contains about 500 samples of 2,000+ words each. The motivation for creating this corpus was to create a heterogeneous sample of English text useful for comparative language studies. Table 3 lists the sections in the Brown corpus.

5.2 Semantic Annotation

Release 3 of the Penn Treebank contains hand-corrected syntactic trees from a subset of the Brown Corpus (sections F, G, K, L, M, N, P, and R). Sections belonging to the newswire genre were not included because a considerable amount of similar material was already available from the WSJ portion of the Treebank. Palmer, Gildea, and Kingsbury (2005) annotated a significant portion of the Treebanked Brown corpus

Table 3
List of sections in the Brown corpus.

A.	Press reportage
B.	Press editorial
C.	Press reviews (theater, books, music, and dance)
D.	Religion
E.	Skills and hobbies
F.	Popular lore
G.	Belles lettres, biography, memoirs, etc.
H.	Miscellaneous
J.	Learned
K.	General fiction
L.	Mystery and detective fiction
M.	Science fiction
N.	Adventure and Western fiction
P.	Romance and love story
R.	Humor

with PropBank roles. The PropBanking philosophy is the same as described earlier. In all, about 17,500 predicates are tagged with their semantic arguments. For these experiments we use the release of the Brown PropBank dated September 2005.

Table 4 shows the number of predicates that have been tagged for each section:

6. Robustness Experiments

In this section, we present a series of experiments comparing the performance of ASSERT on the WSJ corpus to performance on the Brown corpus. The intent is to understand how well the algorithms and features transfer to other sources and to understand the nature of any problems.

6.1 Cross-Genre Testing

The first experiment evaluates the performance of the system when it is trained on annotated data from one genre of text (WSJ) and is used to label a test set from a different genre (the Brown corpus). The ASSERT system described earlier, trained on WSJ Sections 02–21, was used to label arguments for the PropBanked portion of the Brown corpus. As before, the Charniak parser was used to generate the syntax parse trees.

Table 5 shows the F-score for Identification and combined Identification and Classification for each of the eight different text genres as well as the overall performance on Brown. As can be seen, there is a significant degradation across all the various sections of Brown. In addition, although there is a noticeable drop in performance for the Identification task, the bulk of the degradation comes in the combined task.

The following are among the likely factors contributing to this performance degradation:

1. Syntactic parsing errors—The semantic role labeler is completely dependent on the quality of the syntactic parses; missing, mislabeled, and misplaced constituents will all lead to errors. Because the syntactic parser used to generate the parse trees is heavily lexicalized, the genre difference will have an impact on the accuracy of the parses, and the features extracted from them.
2. The Brown corpus may in fact be fundamentally more difficult than the WSJ. There are many potential sources for this kind of difficulty. Among

Table 4
Number of predicates that have been tagged in the PropBanked portion of the Brown corpus.

Section	Total Propositions	Total Lemmas
F	926	321
G	777	302
K	8,231	1,476
L	5,546	1,118
M	167	107
N	863	269
P	788	252
R	224	140

Table 5
Performance on the entire PropBanked Brown corpus when ASSERT is trained on WSJ.

Train	Test	Id. F	Id. + Class F
WSJ	WSJ (Section 23)	85.9	80.0
WSJ	Brown (Popular lore)	77.2	64.9
WSJ	Brown (Biography, memoirs)	77.1	61.1
WSJ	Brown (General fiction)	78.9	64.9
WSJ	Brown (Detective fiction)	82.9	67.1
WSJ	Brown (Science fiction)	83.8	64.5
WSJ	Brown (Adventure)	82.5	65.5
WSJ	Brown (Romance and love story)	81.2	63.9
WSJ	Brown (Humor)	78.8	62.5
WSJ	Brown (All)	81.2	63.9

Table 6
Deleted/missing argument-bearing constituents in Charniak parses of the WSJ test set (Section 23) and the entire PropBanked Brown corpus.

	Total	Misses	%
WSJ (Section 23)	13,612	851	6.2
Brown (Popular lore)	2,280	219	9.6
Brown (Biography, memoirs)	2,180	209	9.6
Brown (General fiction)	21,611	1,770	8.2
Brown (Detective fiction)	14,740	1,105	7.5
Brown (Science fiction)	405	23	5.7
Brown (Adventure)	2,144	169	7.9
Brown (Romance and love story)	1,928	136	7.1
Brown (Humor)	592	61	10.3
Brown (All)	45,880	3,692	8.1

> the most obvious sources are a greater diversity in the range of use of predicates and headwords in the Brown domain. That is, the lexical features may be more varied in terms of predicate senses and raw number of predicates. More consistent usage of predicates and headwords in the WSJ may allow very specific features to be trained in WSJ that will not be as well trained or as salient in Brown.

The following discussion explores each of these possibilities in turn.

Table 6 shows the percentage of argument-bearing nodes deleted from the syntactic parse leading to an Identification error. The syntactic parser deletes 6.2% of the argument bearing nodes in the tree when it is trained and tested on WSJ. When tested on Brown, this number increases to 8.1%, a relative increase of 30%. This effect goes some way toward explaining the decrease in Identification performance, but does not explain the large degradation in combined task performance.

The effect of errors from the syntactic parse can be removed by using the correct syntactic trees from the Treebanks for both corpora. This permits an analysis of other

factors affecting the performance difference. For this experiment, we evaluated performance for all combinations of training and testing on WSJ and Brown. A test set for the Brown corpus was generated by selecting every tenth sentence in the corpus. The development set used by Bacchiani et al. (2006) was withheld for future parameter tuning. No parameter tuning was done for these experiments. The parameters used for the data reported in Table 2 were used for all subsequent tests reported in this article. This procedure results in a training set for Brown that contains approximately 14k predicates. In order to have training sets comparable in size for the two corpora, stratified sampling was used to create a WSJ training set of the same size as the Brown training set. Section 23 of WSJ is still used as the test set for that corpus.

Table 7 shows the results of this experiment. Rows 2 and 4 show the conditions when the system is trained on the 14k predicate WSJ training. Testing on Brown vs. WSJ results in a modest reduction in F-score from 95.3 to 93.0 for argument identification. Although there is some reduction in Identification performance in the absence of errors in the syntactic parse tree, the effect is not large. However, argument classification shows a large drop in accuracy from 86.1% to 72.9%. These data reiterate the point that *syntactic parse errors are not the major factor* accounting for the reduction in performance for Brown.

The next point to note is the effect of varying the amount of training data for WSJ for testing results on WSJ and Brown. The first row of Table 7 shows the performance when ASSERT is trained on the full WSJ training set of Sections 2–21 (90k predicates). The second row shows performance when it is trained on the reduced set of 14k predicates. Whereas the F1 score for Identification dropped by 1.5 percentage points (from 96.8% to 95.3%) the Classification rate dropped by 6.9% percent absolute. Classification seemingly requires considerable more data before its performance begins to asymptote.

Table 7
Performance when ASSERT is trained using correct Treebank parses, and is used to classify test set from either the same genre or another. For each data set, the number of examples used for training are shown in parentheses.

SRL Train	SRL Test	Task	P (%)	R (%)	F	A (%)
WSJ (90k)	WSJ (5k)	Id.	97.5	96.1	96.8	
		Class.				93.0
		Id. + Class.	91.8	90.5	91.2	
WSJ (14k)	WSJ (5k)	Id.	96.3	94.4	95.3	
		Class.				86.1
		Id. + Class.	84.4	79.8	82.0	
BROWN (14k)	BROWN (1.6k)	Id.	95.7	94.9	95.2	
		Class.				80.1
		Id. + Class.	79.9	77.0	78.4	
WSJ (14k)	BROWN (1.6k)	Id.	94.6	91.5	93.0	
		Class.				72.9
		Id. + Class.	72.1	67.2	69.6	
BROWN (14k)	WSJ (5k)	Id.	94.9	93.8	94.3	
		Class.				78.3
		Id. + Class.	76.6	73.3	74.9	

Finally, row 3 shows the performance for training and testing on Brown. The performance of argument Identification is essentially the same as when training and testing on WSJ. However, argument Classification is 6 percentage points worse (80.1% vs. 86.1%) when training and testing on Brown than when training and testing on WSJ. This pattern is consistent with our third hypothesis given previously: Brown may be an intrinsically harder corpus for this task.

Some possible causes for this difficulty are:

1. More unique predicates or head words than are seen in the WSJ set, so there is less training data for each;
2. More predicate sense ambiguity in Brown;
3. Less consistent relations between predicates and head words;
4. A greater preponderance of difficult semantic roles in Brown;
5. Relatively fewer examples of predictive features such as named entities.

The remainder of this section explores each of these possibilities in turn.

In order to test the importance of predicate sense in this process, we added oracle predicate sense information as a feature in ASSERT. Because only about 60% of the PropBanked Brown corpus was tagged with predicate sense information, these results are not directly comparable to the one reported in the earlier tables. In this case, both the Brown training and test sets are subsets of the earlier ones, with about 10k predicates in training and 1k in testing. For comparison, we used the same size WSJ training data. Table 8 shows the performance when trained on WSJ and Brown, and tested on Brown, with and without predicate sense information, and for both Treebank parses and Charniak parses. We find that there is a small increase in the combined identification and classification performance when trained on Brown and tested on Brown.

One reason for this could simply be the raw number of instances that are seen in the training data. Because we know that Predicate and Head Word are two particularly salient features for classification, the percentages of a combination of these features in the Brown test set that are seen in both the training sets should be informative. This information is shown in Table 9. In order to get a cross-corpus statistic, we also present the same numbers on the WSJ test set.

Table 8
Performance on Brown test, using Brown and WSJ training sets, with and without oracle predicate sense information when using Treebank parses.

		Id.			Id. + Class.		
Train	Predicate Sense	P %	R %	F	P %	R %	F
Brown (10k)	×	95.6	95.4	95.5	78.6	76.2	77.4
	√	95.7	95.7	95.7	81.1	77.1	79.0
WSJ (10k)	×	93.4	91.7	92.5	71.1	65.8	68.4
	√	93.3	91.8	92.5	71.3	66.1	68.6

Table 9
Features seen in training for various test sets.

	Test →	WSJ		Brown	
Corpora	Features ↓	T seen (%)	t seen (%)	T seen (%)	t seen (%)
WSJ	Predicate Lemma (P)	76	94	65	80
	Predicate Sense (S)	79	93	64	78
	Head Word (HW)	61	87	49	76
	P+HW	19	31	13	17
Brown	Predicate Lemma (P)	64	85	86	94
	Predicate Sense (S)	29	35	91	96
	Head Word (HW)	37	63	68	87
	P+HW	10	17	27	33

T = types; t = tokens.

It can be seen that for both the WSJ and Brown corpus test sets, the number of predicate lemmas as well as the particular senses seen in the respective test sets is quite high. However, a cross comparison shows that there is about a 15% drop in coverage from WSJ/WSJ to WSJ/Brown. It is also interesting to note that for WSJ, the drop in coverage for predicate lemmas is almost the same as that for individual predicate senses. This further confirms the hypothesis that WSJ has a more homogeneous collection of predicates.

When we compare the drop in coverage for Brown/Brown vs. WSJ/Brown, we find about the same drop in coverage for predicate lemmas, but a much more significant drop for the senses. This variation in senses in Brown is probably the reason that adding sense information helps more for the Brown test set. In the WSJ case, the addition of word sense as a feature does not add much information, and so the numbers are not much different than for the baseline. Similarly, we can see that percentage of head words seen across the two genres also drop significantly, and they are much lower to begin with. Finding the coverage for the predicate lemma and head word combination is still worse, and this is not even considering the sense. Therefore, data sparseness is another potential reason that the importance of the predicate sense feature does not reflect in the performance numbers.

As noted earlier, another possible source of difficulty for Brown may be the distribution of PropBank arguments in this corpus. Table 10 shows the classification performance for each argument, for each of the four configurations (train on Brown or WSJ and test on WSJ or Brown). Among the two most frequent arguments—ARG0 and ARG1—ARG1 seems to be affected the most. When the training and test sets are from the same genre, the performance on ARG0 is slightly worse on the Brown test set. ARG1 on the other hand is about 5% worse on both precision and recall, when trained and tested on Brown. For core-arguments ARG2–5 which are highly predicate sense dependent, there is a much larger performance drop.

Finally, another possible reason for the drop in performance is the distribution of named entities in the corpus. Table 11 shows the frequency of occurrence of name entities in 10k WSJ and Brown training sets. It can be seen that number of organizations talked about in Brown is much smaller than in WSJ, and there are more person names. Also, monetary amounts which frequently fill the ARG3 and ARG4 slots are also much more infrequent in Brown, and so is the incidence of percentages. This would definitely have some impact on the usability of these features in the learned models.

7. Effect of Improved Syntactic Parses

Practical natural language processing systems will always use errorful automatic parses, and so it would be interesting to find out how much syntactic parser errors hinder performance on the task of semantic role labeling. Fortunately, recent improvements to the Charniak parser provided an opportunity to test this hypothesis. We use the latest version of the Charniak parser that does *n*-best re-ranking (Charniak and Johnson 2005) and the model that is self-trained using the North American News corpus (NANC). This version adapts much better to the Brown corpus (McClosky, Charniak, and Johnson

Table 10
Classification accuracy for each argument type in the WSJ (W) and Brown (B) test sets.

			W×W		B×B		B×W		W×B	
Argument	Number in WSJ Test	Number in Brown Test	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
ARG0	3,149	1,122	91.1	96.8	90.4	92.8	83.4	92.2	87.4	93.3
ARG1	4,264	1,375	90.2	92.0	85.0	88.5	78.7	79.7	83.4	89.0
ARG2	796	312	73.3	66.6	65.9	60.6	49.7	56.4	59.5	48.1
ARG3	128	25	74.3	40.6	71.4	20.0	30.8	16.0	28.6	4.7
ARG4	72	20	89.1	68.1	57.1	60.0	16.7	5.0	61.1	15.3
C-ARG0	2	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C-ARG1	165	34	91.5	64.8	80.0	35.3	64.7	32.4	82.1	19.4
R-ARG0	189	45	83.1	93.7	82.7	95.6	62.5	88.9	76.8	77.2
R-ARG1	122	44	77.8	63.1	91.7	75.0	64.5	45.5	54.5	59.8
ARGM-ADV	435	290	78.0	66.0	67.6	64.8	74.7	44.8	49.9	71.0
ARGM-CAU	65	15	82.5	72.3	80.0	53.3	62.5	66.7	86.0	56.9
ARGM-DIR	72	114	57.1	50.0	71.0	62.3	46.6	36.0	39.7	43.1
ARGM-DIS	270	65	87.6	86.7	81.0	72.3	54.1	70.8	89.6	64.1
ARGM-EXT	31	10	83.3	48.4	0.0	0.0	0.0	0.0	33.3	3.2
ARGM-LOC	317	147	73.8	80.8	60.8	70.7	52.6	48.3	60.6	65.6
ARGM-MNR	305	144	56.1	59.0	64.5	63.2	42.6	55.6	51.4	48.9
ARGM-MOD	454	129	99.6	100.0	100.0	100.0	100.0	99.2	99.6	100.0
ARGM-NEG	201	85	100.0	99.5	97.7	98.8	100.0	85.9	94.8	99.5
ARGM-PNC	99	43	60.4	58.6	66.7	55.8	54.8	39.5	52.8	57.6
ARGM-PRD	5	8	0.0	0.0	33.3	12.5	0.0	0.0	0.0	0.0
ARGM-TMP	978	280	85.4	90.4	84.8	85.4	71.3	83.6	82.2	76.0

W×B = ASSERT trained on B and used to classify W test set.

Table 11
Distribution of the named entities in a 10k data from WSJ and Brown corpora.

Name Entity	WSJ	Brown
PERSON	1,274	2,037
ORGANIZATION	2,373	455
LOCATION	1,206	555
MONEY	831	32
DATE	710	136
PERCENT	457	5
TIME	9	21

Table 12
Performance for different versions of the Charniak parser used in the experiments.

Train	Test	F
WSJ	WSJ	91.0
WSJ	Brown	85.2
Brown	Brown	88.4
WSJ+NANC	Brown	87.9

2006a, 2006b). We also use another model that is trained on the Brown corpus itself. The performance of these parsers is shown in Table 12.

We describe the results of the following five experiments:

1. ASSERT is trained on features extracted from automatically generated parses of the PropBanked WSJ sentences. The syntactic parser (Charniak parser) is itself trained on the WSJ training sections of the Treebank. This is used to classify Section 23 of WSJ.
2. ASSERT is trained on features extracted from automatically generated parses of the PropBanked WSJ sentences. The syntactic parser (Charniak parser) is itself trained on the WSJ training sections of the Treebank. This is used to classify the Brown test set.
3. ASSERT is trained on features extracted from automatically generated parses of the PropBanked Brown corpus sentences. The syntactic parser is trained using the WSJ portion of the Treebank. This is used to classify the Brown test set.
4. ASSERT is trained on features extracted from automatically generated parses of the PropBanked Brown corpus sentences. The syntactic parser is trained using the Brown training portion of the Treebank. This is used to classify the Brown test set.
5. ASSERT is trained on features extracted from automatically generated parses of the PropBanked Brown corpus sentences. The syntactic parser is the version that is self-trained using 2,500,000 sentences from NANC, and where the starting version is trained only on WSJ data (McClosky, Charniak, and Johnson 2006b). This is used to classify the Brown test set.

The same training and test sets used for the systems in Table 7 are used in this experiment. Table 13 shows the results. For simplicity of discussion we have labeled the five conditions as A, B, C, D, and E. Comparing conditions B and C shows that when the features used to train ASSERT are extracted using a syntactic parser that is trained on WSJ it performs at almost the same level on the task of identification, regardless of whether it is trained on the PropBanked Brown corpus or the PropBanked WSJ corpus. This, however, is about 5–6 F-score points lower than when all the three (the syntactic parser training set, ASSERT training set, and ASSERT test set) are from the same genre—WSJ or Brown, as seen in A and D. For the combined task, the gap between the performance for conditions B and C is about 10 F-score points apart (59.1 vs. 69.8). Looking at the argument classification accuracies, we see that using ASSERT trained on WSJ to test Brown sentences results in a 12-point drop in F-score. Using ASSERT trained on Brown

Table 13
Performance on WSJ and Brown test sets when ASSERT is trained on features extracted from automatically generated syntactic parses.

Setup	Parser Train	SRL Train	SRL Test	Task	P (%)	R (%)	F	A (%)
A.	WSJ (40k – sec:00–21)	WSJ (14k)	WSJ (5k)	Id.	87.3	84.8	86.0	
				Class.				84.1
				Id. + Class.	77.5	69.7	73.4	
B.	WSJ (40k – sec:00–21)	WSJ (14k)	Brown (1.6k)	Id.	81.7	78.3	79.9	
				Class.				72.1
				Id. + Class.	63.7	55.1	59.1	
C.	WSJ (40k – sec:00–21)	Brown (14k)	Brown (1.6k)	Id.	81.7	78.3	80.0	
				Class.				79.2
				Id. + Class.	78.2	63.2	69.8	
D.	Brown (20k)	Brown (14k)	Brown (1.6k)	Id.	87.6	82.3	84.8	
				Class.				78.9
				Id. + Class.	77.4	62.1	68.9	
E.	WSJ+NANC (2,500k)	Brown (14k)	Brown (1.6k)	Id.	87.7	82.5	85.0	
				Class.				79.9
				Id. + Class.	77.2	64.4	70.0	
H.	WSJ+NANC (2,500k)	Brown (14k)	WSJ (5k)	Id.	88.2	78.2	82.8	
				Class.				76.9
				Id. + Class.	75.4	51.6	61.2	

using the WSJ-trained syntactic parser reduces accuracy by about 5 F-score points. When ASSERT is trained on Brown using a syntactic parser also trained on Brown, we get a quite similar classification performance, which is again about 5 points lower than what we get using all WSJ data. Finally, looking at conditions C and D we find that the difference in performance on the combined task of identification and classification using the Brown corpus for training ASSERT is very close (69.8 vs. 68.9) even though the syntactic parser used in C has a performance that is about 3.2 points worse than that used in D. This indicates that better parse structure is less important than lexical semantic coverage for obtaining better performance on the Brown corpus.

8. Adapting to a New Genre

One possible way to ameliorate the effects of domain specificity is to incrementally add small amounts of data from a new domain to the already available out-of-domain training data. In the following experiments we explore this possibility by slowly adding data from the Brown corpus to a fixed amount of WSJ data.

One section of the Brown corpus—section K—has about 8,200 predicates annotated. Therefore, we will take six different scenarios—two in which we will use correct Treebank parses, and the four others in which we will use automatically generated parses using the variations used before. All training sets start with the same number of examples as that of the Brown training set. The part of this section used as a test set for the CoNLL 2005 shared task was used as the test set for these experiments. This test set contains 804 predicates in 426 sentences of Brown section K.

Table 14 shows the results. In all six settings, the performance on the task of identification and classification improves gradually until about 5,625 examples of section K, which is about 75% of the total added, above which it adds very little. Even when the syntactic parser is trained on WSJ and the SRL is trained on WSJ, adding 7,500 instances of this new genre achieves almost the same performance as when all three are from the same genre (67.2 vs. 69.9). For the task of argument identification, the incremental addition of data from the new genre shows only minimal improvement. The system that uses a self-trained syntactic parser performs slightly better than other

Table 14
Effect of incrementally adding data from a new genre.

		Id.			Id. + Class		
Parser Train	SRL Train	P (%)	R (%)	F	P (%)	R (%)	F
WSJ (Treebank parses)	WSJ (14k) (Treebank parses)						
	+0 examples from K	96.2	91.9	94.0	74.1	66.5	70.1
	+1,875 examples from K	96.1	92.9	94.5	77.6	71.3	74.3
	+3,750 examples from K	96.3	94.2	95.1	79.1	74.1	76.5
	+5,625 examples from K	96.4	94.8	95.6	80.4	76.1	78.1
	+7,500 examples from K	96.4	95.2	95.8	80.2	76.1	78.1
Brown (Treebank parses)	Brown (14k) (Treebank parses)						
	+0 examples from K	96.1	94.2	95.1	77.1	73.0	75.0
	+1,875 examples from K	96.1	95.4	95.7	78.8	75.1	76.9
	+3,750 examples from K	96.3	94.6	95.3	80.4	76.9	78.6
	+5,625 examples from K	96.2	94.8	95.5	80.4	77.2	78.7
	+7,500 examples from K	96.3	95.1	95.7	81.2	78.1	79.6
WSJ (40k)	WSJ (14k)						
	+0 examples from K	83.1	78.8	80.9	65.2	55.7	60.1
	+1,875 examples from K	83.4	79.3	81.3	68.9	57.5	62.7
	+3,750 examples from K	83.9	79.1	81.4	71.8	59.3	64.9
	+5,625 examples from K	84.5	79.5	81.9	74.3	61.3	67.2
	+7,500 examples from K	84.8	79.4	82.0	74.8	61.0	67.2
WSJ (40k)	Brown (14k)						
	+0 examples from K	85.7	77.2	81.2	74.4	57.0	64.5
	+1,875 examples from K	85.7	77.6	81.4	75.1	58.7	65.9
	+3,750 examples from K	85.6	78.1	81.7	76.1	59.6	66.9
	+5,625 examples from K	85.7	78.5	81.9	76.9	60.5	67.7
	+7,500 examples from K	85.9	78.1	81.7	76.8	59.8	67.2
Brown (20k)	Brown (14k)						
	+0 examples from K	87.6	80.6	83.9	76.0	59.2	66.5
	+1,875 examples from K	87.4	81.2	84.1	76.1	60.0	67.1
	+3,750 examples from K	87.5	81.6	84.4	77.7	62.4	69.2
	+5,625 examples from K	87.5	82.0	84.6	78.2	63.5	70.1
	+7,500 examples from K	87.3	82.1	84.6	78.2	63.2	69.9
WSJ+NANC (2,500k)	Brown (14k)						
	+0 examples from K	89.1	81.7	85.2	74.4	60.1	66.5
	+1,875 examples from K	88.6	82.2	85.2	76.2	62.3	68.5
	+3,750 examples from K	88.3	82.6	85.3	76.8	63.6	69.6
	+5,625 examples from K	88.3	82.4	85.2	77.7	63.8	70.0
	+7,500 examples from K	88.9	82.9	85.8	78.2	64.9	70.9

versions that use automatically generated syntactic parses. The improvement on the identification performance is almost exclusively due to recall. The precision numbers are almost unaffected, except when the labeler is trained on WSJ PropBank data.

9. Conclusions

In this article, we have presented results from a state-of-the-art Semantic Role Labeling system trained on PropBank WSJ data and then used to label test sets from both the WSJ corpus and the Brown corpus. The system's performance on the Brown test set exhibited a large drop compared to the WSJ test set. An analysis of these results revealed that the subtask of Identification, determining which constituents of a syntax tree are arguments of a predicate, is responsible for only a relatively small part of the drop in performance. The Classification task, assigning labels to constituents known to be arguments, is where the major performance loss occurs.

Several possible factors were examined to determine their effect on this performance difference:

- The syntactic parser was trained on WSJ. It was shown that errors in the syntactic parse are not a large factor in the overall performance difference. The syntactic parser does not show a large degradation in performance when run on Brown. Even more telling, there is still a large drop in performance when training and testing using Treebank parses.

When the system was trained and tested on Brown, the performance was still significantly worse than training and testing on WSJ, even when the amount of training data is controlled for. Training and testing on Brown showed performance intermediate between training and testing on WSJ and training on WSJ and testing on Brown. This leads to our final hypothesis.

- The Brown corpus is in some sense fundamentally more difficult for this problem. The most obvious reason for this is that it represents a more heterogeneous source than the WSJ. Among the likely manifestations of this is that predicates tend to have a single dominating sense in WSJ and are more polysemous in Brown. Data was presented using gold-standard word sense information for the predicates for training and testing Brown. Adding predicate sense information has a large effect for some predicates, but over the whole Brown test set has only a small effect. Fewer predicates and headwords could allow very specific modeling of high frequency predicates, and predicate–headword relations do have a large effect on overall performance.

The initial experiment is a case of training on homogeneous data and testing on different data. The more homogeneous training data allows the system to rely heavily on specific features and relations in the data. It is usually the case that training on a more heterogeneous data set does not give quite as high performance on test data from the same corpus as more homogeneous data, but the heterogeneous data ports better to other corpora. This is seen when training on Brown compared to WSJ. The observation that the Identification task ports well while the classification task does not is consistent with this explanation. For the Identification task, structural features such as path and

partial path tend to be the most salient while the Classification task relies more heavily on lexical/semantic features such as specific predicate-head word combinations.

The question now is what to do about this. Two possibilities are:

- **Less homogeneous corpora—**Rather than using many examples drawn from one source, fewer examples could be drawn from many sources. This would reduce the likelihood of learning idiosyncratic senses and argument structures for predicates.
- **Less specific features—**Features, and the values they take on, should be designed to reduce the likelihood of learning idiosyncratic aspects of the training domain. Examples of this might include the use of more general named entity classes, and the use of abstractions over specific headwords and predicates rather than the words themselves.

Both of these manipulations would, in all likelihood, reduce performance on both the training data and on test sets of the same genre as the training data. But they would be more likely to lead to better generalization across genres. Training on very homogeneous training sets and testing on similar test sets gives a misleading impression of the performance of a system.

Acknowledgments

We are extremely grateful to Martha Palmer for providing us with the PropBanked Brown corpus, and to David McClosky for providing us with hypotheses on the Brown test set as well as a cross-validated version of the Brown training data for the various models reported in his work reported at HLT 2006.

This research was partially supported by the ARDA AQUAINT program via contract OCG4423B and by the NSF via grants IS-9978025 and ITR/HCI 0086132. Computer time was provided by NSF ARI Grant CDA-9601817, NSF MRI Grant CNS-0420873, NASA AIST grant NAG2-1646, DOE SciDAC grant DE-FG02-04ER63870, NSF sponsorship of the National Center for Atmospheric Research, and a grant from the IBM Shared University Research (SUR) program.

References

Bacchiani, Michiel, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34:211–231.

Burges, Christopher J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Carreras, Xavier and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, pages 89–97, Boston, MA.

Carreras, Xavier and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 152–164, Ann Arbor, MI.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine *n*-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Hacioglu, Kadri, Sameer Pradhan, Wayne Ward, James Martin, and Daniel Jurafsky. 2004. Semantic role labeling by tagging syntactic chunks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, Boston, MA.

Harabagiu, Sanda, Cosmin Adrian Bejan, and Paul Morarescu. 2005. Shallow semantics for relation extraction. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1061–1067, Edinburgh, Scotland.

Hofmann, Thomas and Jan Puzicha. 1998. Statistical models for co-occurrence data. Memo, Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA.

Jiang, Zheng Ping, Jia Li, and Hwee Tou Ng. 2005. Semantic argument classification exploiting argument interdependence. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1067–1072, Edinburgh, Scotland.

Joachims, Thorsten. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 137–142, Chemnitz, Germany.

Koomen, Peter, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized inference with multiple semantic role labeling systems. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 181–184, Ann Arbor, MI.

Kučera, Henry and W. Nelson Francis. 1967. *Computational Analysis of Present-day American English*. Brown University Press, Providence, RI.

Kudo, Taku and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. In *Proceedings of the Fourth Conference on Computational Natural Language Learning (CoNLL)*, pages 142–144, Lisbon, Portugal.

Kudo, Taku and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Pittsburgh, PA.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the Seventeenth International Conference on Computational Linguistics and Thirty Sixth Annual Meeting of the Association of Computational Linguistics (COLING/ACL)*, pages 768–774, Montreal, Canada.

Lodhi, Huma, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Màrquez, Lluís, Mihai Surdeanu, Pere Comas, and Jordi Turmo. 2005. A robust combination strategy for semantic role labeling. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 644–651, Vancouver, British Columbia.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 152–159, New York, NY.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the Twenty First International Conference on Computational Linguistics and Forty Fourth Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, pages 337–344, Sydney, Australia.

Moschitti, Alessandro. 2006. Syntactic kernels for natural language learning: The semantic role labeling case. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 97–100, New York, NY.

Musillo, Gabriele and Paola Merlo. 2006. Accurate parsing of the proposition bank. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 101–104, New York, NY.

Narayanan, Srini and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 693–701, Geneva, Switzerland.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Platt, John. 2000. Probabilities for support vector machines. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, pages 61–74.

Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James Martin, and Dan Jurafsky. 2005. Support vector

learning for semantic argument classification. *Machine Learning Journal*, 60(1):11–39.

Pradhan, Sameer, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/ North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 233–240, Boston, MA.

Pradhan, Sameer, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the Forty-Third Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 581–588, Ann Arbor, MI.

Punyakanok, Vasin, Dan Roth, Wen tau Yih, and Dav Zimak. 2005. Learning and inference over constrained output. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1117–1123, Edinburgh, Scotland.

Surdeanu, Mihai, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the Forty-First Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8–15, Sapporo, Japan.

Toutanova, Kristina, Aria Haghighi, and Christopher Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of the Forty-Third Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 589–596, Ann Arbor, MI.

Vapnik, Vladimir. 1998. *Statistical Learning Theory*. Wiley, New York.

Xue, Nianwen and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 88–94, Barcelona, Spain.

Yi, Szu-ting and Martha Palmer. 2005. The integration of syntactic parsing and semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 237–240, Ann Arbor, MI.