Generalizing Semantic Role Annotations Across Syntactically Similar Verbs

Andrew S. Gordon
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA 90292 USA
gordon@ict.usc.edu

Reid Swanson
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA 90292 USA
swansonr@ict.usc.edu

Abstract

Large corpora of parsed sentences with semantic role labels (e.g. PropBank) provide training data for use in the creation of high-performance automatic semantic role labeling systems. Despite the size of these corpora, individual verbs (or rolesets) often have only a handful of instances in these corpora, and only a fraction of English verbs have even a single annotation. In this paper, we describe an approach for dealing with this sparse data problem, enabling accurate semantic role labeling for novel verbs (rolesets) with only a single training example. Our approach involves the identification of syntactically similar verbs found in PropBank, the alignment of arguments in their corresponding rolesets, and the use of their corresponding annotations in PropBank as surrogate training data.

1 Generalizing Semantic Role Annotations

A recent release of the PropBank (Palmer et al., 2005) corpus of semantic role annotations of Treebank parses contained 112,917 labeled instances of 4,250 rolesets corresponding to 3,257 verbs, as illustrated by this example for the verb *buy*.

$[_{\text{arg0}}$ Chuck$]$ $[_{\text{buy.01}}$ bought$]$ $[_{\text{arg1}}$ a car$]$ $[_{\text{arg2}}$ from Jerry$]$ $[_{\text{arg3}}$ for \$1000$]$.

Annotations similar to these have been used to create automated semantic role labeling systems (Pradhan et al., 2005; Moschitti et al., 2006) for use in natural language processing applications that require only shallow semantic parsing. As with all machine-learning approaches, the performance of these systems is heavily dependent on the availability of adequate amounts of training data. However, the number of annotated instances in PropBank varies greatly from verb to verb; there are 617 annotations for the *want* roleset, only 7 for *desire*, and 0 for any sense of the verb *yearn*. Do we need to keep annotating larger and larger corpora in order to generate accurate semantic labeling systems for verbs like *yearn*?

A better approach may be to generalize the data that exists already to handle novel verbs. It is reasonable to suppose that there must be a number of verbs within the PropBank corpus that behave nearly exactly like *yearn* in the way that they relate to their constituent arguments. Rather than annotating new sentences that contain the verb *yearn*, we could simply find these similar verbs and use their annotations as surrogate training data.

This paper describes an approach to generalizing semantic role annotations across different verbs, involving two distinct steps. The first step is to order all of the verbs with semantic role annotations according to their syntactic similarity to the target verb, followed by the second step of aligning argument labels between different rolesets. To evaluate this approach we developed a simple automated semantic role labeling algorithm based on the frequency of parse-tree paths, and then compared its performance when using real and surrogate training data from PropBank.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 192–199,
Prague, Czech Republic, June 2007.

2 Parse Tree Paths

A key concept in understanding our approach to both automated semantic role annotation and generalization is the notion of a *parse tree path*. Parse tree paths were used for semantic role labeling by Gildea and Jurafsky (2002) as descriptive features of the syntactic relationship between predicates and their arguments in the parse tree of a sentence. Predicates are typically assumed to be specific target words (verbs), and arguments are assumed to be spans of words in the sentence that are dominated by nodes in the parse tree. A parse tree path can be described as a sequence of transitions up from the target word then down to the node that dominates the argument span (e.g. Figure 1).

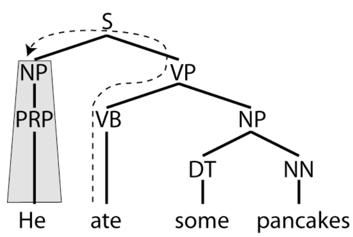

Figure 1: An example parse tree path from the predicate *ate* to the argument NP *He*, represented as ↑VB↑VP↑S↓NP

Parse tree paths are particularly interesting for automated semantic role labeling because they generalize well across syntactically similar sentences. For example, the parse tree path in Figure 1 would still correctly identify the "eater" argument in the given sentence if the personal pronoun "he" were swapped with a markedly different noun phrase, e.g. "the attendees of the annual holiday breakfast."

3 A Simple Semantic Role Labeler

To explore issues surrounding the generalization of semantic role annotations across verbs, we began by authoring a simple automated semantic role labeling algorithm that assigns labels according to the frequency of the parse tree paths seen in training data. To construct a labeler for a specific roleset, training data consisting of parsed sentences with role-labeled parse tree constituents are analyzed to identify all of the parse tree paths between predicates and arguments, which are then tabulated and sorted by frequency. For example, Table 1 lists the 10 most frequent pairs of arguments and parse tree paths for the *want.01* roleset in a recent release of PropBank.

Count	Argument	Parse tree path
189	ARG0	↑VBP↑VP↑S↓NP
159	ARG1	↑VBP↑VP↓S
125	ARG0	↑VBZ↑VP↑S↓NP
110	ARG1	↑VBZ↑VP↓S
102	ARG0	↑VB↑VP↑VP↑S↓NP
98	ARG1	↑VB↑VP↓S
96	ARG0	↑VBD↑VP↑S↓NP
79	ARGM	↑VB↑VP↑VP↓RB
76	ARG1	↑VBD↑VP↓S
43	ARG1	↑VBP↑VP↓NP

Table 1. Top 10 most frequent parse tree paths for arguments of the PropBank want.01 roleset, based on 617 annotations

To automatically assign role labels to an unlabeled parse tree, each entry in the table is considered in order of highest frequency. Beginning from the target word in the sentence (e.g. *wants*) a check is made to determine if the entry includes a possible parse tree path in the parse tree of the sentence. If so, then the constituent is assigned the role label of the entry, and all subsequent entries in the table that have the same argument label or lead to subconstituents of the labeled node are invalidated. Only subsequent entries that assign core arguments of the roleset (e.g. ARG0, ARG1) are invalidated, allowing for multiple assignments of non-core labels (e.g. ARGM) to a test sentence. In cases where the path leads to more than one node in a sentence, the leftmost path is selected. This process then continues down the list of valid table entries, assigning additional labels to unlabeled parse tree constituents, until the end of the table is reached.

This approach also offers a simple means of dealing with multiple-constituent arguments, which occasionally appear in PropBank data. In these cases, the data is listed as unique entries in the frequency table, where each of the parse tree paths to the multiple constituents are listed as a set. The labeling algorithm will assign the argument of the entry only if all parse tree paths in the set are present in the sentence.

The expected performance of this approach to semantic role labeling was evaluated using the PropBank data using a leave-one-out cross-validation experimental design. Precision and recall scores were calculated for each of the 3,086 rolesets with at least two annotations. Figure 2 graphs the average precision, recall, and F-score for rolesets according to the number of training examples of the roleset in the PropBank corpus. An additional curve in Figure 2 plots the percentage of these PropBank rolesets that have the given amount of training data or more. For example, F-scores above 0.7 are first reached with 62 training examples, but only 8% of PropBank rolesets have this much training data available.

Figure 2. Performance of our semantic role labeling approach on PropBank rolesets

4 Identifying Syntactically Similar Verbs

A key part of generalizing semantic role annotations is to calculate the syntactic similarity between verbs. The expectation here is that verbs that appear in syntactically similar contexts are going to behave similarly in the way that they relate to their arguments. In this section we describe a fully automated approach to calculating the syntactic similarity between verbs.

Our approach is strictly empirical; the similarity of verbs is determined by examining the syntactic contexts in which they appear in a large text corpus. Our approach is analogous to previous work in extracting collocations from large text corpora using syntactic information (Lin, 1998). In our work, we utilized the GigaWord corpus of English newswire text (Linguistic Data Consortium, 2003), consisting of nearly 12 gigabytes of textual data. To prepare this corpus for analysis, we extracted the body text from each of the 4.1 million entries in the corpus and applied a maximum-entropy algorithm to identify sentence boundaries (Reynar and Ratnaparkhi, 1997).

Next we executed a four-step analysis process for each of the 3,257 verbs in the PropBank corpus. In the first step, we identified each of the sentences in the prepared GigaWord corpus that contained any inflection of the given verb. To automatically identify all verb inflections, we utilized the English DELA electronic dictionary (Courtois, 2004), which contained all but 21 of the PropBank verbs (for which we provided the inflections ourselves), with old-English verb inflections removed. We extracted GigaWord sentences containing these inflections by using the GNU *grep* program and a template regular expression for each inflection list. The results of these searches were collected in 3,257 files (one for each verb). The largest of these files was for inflections of the verb *say* (15.9 million sentences), and the smallest was for the verb *namedrop* (4 sentences).

The second step was to automatically generate syntactic parse trees for the GigaWord sentences found for each verb. It was our original intention to parse all of the found sentences, but we found that the slow speed of contemporary syntactic parsers made this impractical. Instead, we focused our efforts on the first 100 sentences found for each of the 3,257 verbs with 100 or fewer tokens: a total of 324,461 sentences (average of 99.6 per verb). For this task we utilized the August 2005 release of the Charniak parser with the default speed/accuracy settings (Charniak, 2000), which required roughly 360 hours of processor time on a 2.5 GHz PowerPC G5.

The third step was to characterize the syntactic context of the verbs based on where they appeared within the parse trees. For this purpose, we utilized parse tree paths as a means of converting tree structures into a flat, feature-vector representation. For each sentence, we identified all *possible* parse tree paths that begin from the verb inflection and terminate at a constituent that does not include the verb inflection. For example, the syntactic context of the verb in Figure 1 can be described by the following five parse tree paths:

1. ↑VB↑VP↑S↓NP
2. ↑VB↑VP↑S↓NP↓PRP
3. ↑VB↑VP↓NP
4. ↑VB↑VP↓NP↓DT
5. ↑VB↑VP↓NP↓NN

Possible parse tree paths were identified for every parsed sentence for a given verb, and the frequencies of each unique path were tabulated into a feature vector representation. Parse tree paths where the first node was not a Treebank part-of-speech tag for a verb were discarded, effectively filtering the non-verb homonyms of the set of inflections. The resulting feature vectors were normalized by dividing the values of each feature by the number of verb instances used to generate the parse tree paths; the value of each feature indicates the proportion of observed inflections in which the parse tree path is possible. As a representative example, 95 verb forms of *abandon* were found in the first 100 GigaWord sentences containing any inflection of this verb. For this verb, 4,472 possible parse tree paths were tabulated into 3,145 unique features, 2501 of which occurred only once.

The fourth step was to compute the distance between a given verb and each of the 3,257 feature vector representations describing the syntactic context of PropBank verbs. We computed and compared the performance of a wide variety of possible vector-based distance metrics, including Euclidean, Manhattan, and Chi-square (with un-normalized frequency counts), but found that the ubiquitous cosine measure was least sensitive to variations in sample size between verbs. To facilitate a comparative performance evaluation (section 6), pairwise cosine distance measures were calculated between each pair of PropBank verbs and sorted into individual files, producing 3,257 lists of 3,257 verbs ordered by similarity.

Table 2 lists the 25 most syntactically similar pairs of verbs among all PropBank verbs. There are a number of notable observations in this list. First is the extremely high similarity between *bind* and *bound*. This is partly due to the fact that they share an inflection (*bound* is the irregular past tense form of *bind*), so the first 100 instances of GigaWord sentences for each verb overlap significantly, resulting in overlapping feature vector representations. Although this problem appears to be restricted to this one pair of verbs, it could be avoided in the future by using the part-of-speech tag in the parse tree to help distinguish between verb lemmas.

A second observation of Table 2 is that several verbs appear multiple times in this list, yielding sets of verbs that all have high syntactic similarity. Three of these sets account for 19 of the verbs in this list:

1. plunge, tumble, dive, jump, fall, fell, dip
2. assail, chide, lambaste
3. buffet, embroil, lock, superimpose, whipsaw, pluck, whisk, mar, ensconce

The appearance of these sets suggests that our method of computing syntactic similarity could be used to identify distinct clusters of verbs that behave in very similar ways. In future work, it would be particularly interesting to compare empirically-derived verb clusters to verb classes derived from theoretical considerations (Levin, 1993), and to the automated verb classification techniques that use these classes (Joanis and Stevenson, 2003).

A third observation of Table 2 is that the verb pairs with the highest syntactic similarity are often synonyms, e.g. the cluster of *assail*, *chide*, and *lambaste*. As a striking example, the 14 most syntactically similar verbs to *believe* (in order) are *think*, *guess*, *hope*, *feel*, *wonder*, *theorize*, *fear*, *reckon*, *contend*, *suppose*, *understand*, *know*, *doubt*, and *suggest* – all mental action verbs. This observation further supports the distributional hypothesis of word similarity and corresponding technologies for identifying synonyms by similarity of lexical-syntactic context (Lin, 1998).

Verb pairs (instances)		*Cosine*
bind (83)	bound (95)	0.950
plunge (94)	tumble (87)	0.888
dive (36)	plunge (94)	0.867
dive (36)	tumble (87)	0.866
jump (79)	tumble (87)	0.865
fall (84)	fell (102)	0.859
intersperse (99)	perch (81)	0.859
assail (100)	chide (98)	0.859
dip (81)	fell (102)	0.858
buffet (72)	embroil (100)	0.856
embroil (100)	lock (73)	0.856
embroil (100)	superimpose (100)	0.856
fell (102)	jump (79)	0.855
fell (102)	tumble (87)	0.855
embroil (100)	whipsaw (63)	0.850
pluck (100)	whisk (99)	0.849
acquit (100)	hospitalize (99)	0.849
disincline (70)	obligate (94)	0.848
jump (79)	plunge (94)	0.848
dive (36)	jump (79)	0.847
assail (100)	lambaste (100)	0.847
festoon (98)	strew (100)	0.846
mar (78)	whipsaw (63)	0.846
pluck (100)	whipsaw (63)	0.846
ensconce (101)	whipsaw (63)	0.845

Table 2. Top 25 most syntactically similar pairs of the 3257 verbs in PropBank. Each verb is listed with the number of inflection instances used to calculate the cosine measurement.

5 Aligning Arguments Across Rolesets

The second key aspect of our approach to generalizing annotations is to make mappings between the argument roles of the novel target verb and the roles used for a given roleset in the PropBank corpus. For example, if we'd like to apply the training data for a roleset of the verb *desire* in PropBank to a novel roleset for the verb *yearn*, we need to know that the *desirer* corresponds to the *yearner*, the *desired* to the *yearned-for*, etc. In this section, we describe an approach to argument alignment that involves the application of the semantic role labeling approach described in section 3 to a single training example for the target verb.

To simplify the process of aligning argument labels across rolesets, we make a number of assumptions. First, we only consider cases where two rolesets have exactly the same number of arguments. The version of the PropBank corpus that we used in this research contained 4250 rolesets, each with 6 or fewer roles (typically two or three). Accordingly, when attempting to apply PropBank data to a novel roleset with a given argument count (e.g. two), we only consider the subset of PropBank data that labels rolesets with exactly the same count.

Second, our approach requires at least one fully-annotated training example for the target roleset. A fully-annotated sentence is one that contains a labeled constituent in its parse tree for each role in the roleset. As an illustration, the example sentence in section 1 (for the roleset *buy.01*) would not be considered a fully-annotated training example, as only four of the five arguments of the PropBank *buy.01* roleset are present in the sentence (it is missing a *benefactor*, as in "Chuck bought *his mother* a car from Jerry for $1000").

In both of these simplifying requirements, we ignore role labels that may be assigned to a sentence but that are not defined as part of the roleset, specifically the *ARGM* labels used in PropBank to label standard proposition modifiers (e.g. location, time, manner).

Our approach begins with a list of verbs ordered by their calculated syntactic similarity to the target verb, as described in section 4 of this paper. We subsequently apply two steps that transform this list into an ordered set of rolesets that can be aligned with the roles used in one or more fully-annotated training examples of the target verb. In describing these two steps, we use *instigate* as an example target verb. Instigate already appears in the PropBank corpus as a two-argument roleset, but it has only a single training example:

> [$_{\text{arg0}}$ The Mahatma, or "great souled one,"] [$_{\text{instigate.01}}$ instigated] [$_{\text{arg1}}$ several campaigns of passive resistance against the British government in India].

The syntactic similarity of *instigate* to all PropBank verbs was calculated in the manner described in the previous section. This resulting list of 3,180 entries begins with the following fourteen verbs: *orchestrate*, *misrepresent*, *summarize*, *wreak*, *rub*, *chase*, *refuse*, *embezzle*, *harass*, *spew*, *thrash*, *unearth*, *snub*, and *erect*.

The first step is to replace each of the verbs in the ordered list with corresponding rolesets from PropBank that have the same number of roles as the target verb. As an example, our target roleset for the verb *instigate* has two arguments, so each verb in the ordered list is replaced with the set of corresponding rolesets that also have two arguments, or removed if no two-argument rolesets exist for the verb in the PropBank corpus. The ordered list of verbs for *instigate* is transformed into an ordered list of 2,115 rolesets with two arguments, beginning with the following five entries: *orchestrate.01*, *chase.01*, *unearth.01*, *snub.01*, and *erect.01*.

The second step is to identify the alignments between the arguments of the target roleset and each of the rolesets in the ordered list. Beginning with the first roleset on the list (e.g. *orchestrate.01*), we build a semantic role labeler (as described in section 3) using its available training annotations from the PropPank corpus. We then apply this labeler to the single, fully-annotated example sentence for the target verb, treating it as if it were a test example of the same roleset. We then check to see if any of the core (numbered) role labels *overlap* with the annotations that are provided. In cases where an annotated constituent of the target test sentence is assigned a label from the source roleset, then the roleset mappings are noted along with the entry in the ordered list. If no mappings are found, the roleset is removed from the ordered list.

For example, the roleset for *orchestrate.01* contains two arguments (*ARG0* and *ARG1*) that correspond to the "conductor, manager" and the "things being coordinated or managed". This roleset is used for only three sentence annotations in the PropBank corpus. Using these annotations as training data, we build a semantic role labeler for this roleset and apply it to the annotated sentence for *instigate.01*, treating it as if it were a test sentence for the roleset *orchestrate.01*. The labeler assigns the *orchestrate.01* label *ARG1* to the same constituent labeled *ARG1* in the test sentence, but fails to assign a label to the other argument constituent in the test sentence. Therefore, a single mapping is recorded in the ordered list of rolesets, namely that *ARG1* of *orchestrate.01* can be mapped to *ARG1* of *instigate.01*.

After all of the rolesets are considered, we are left with a filtered list of rolesets with their argument mappings, ordered by their syntactic similarity to the target verb. For the roleset instigate.01, this list consists of 789 entries, beginning with the following 5 mappings.

1. *orchestrate.01*, 1:1
2. *chase.01*, 0:0, 1:1
3. *unearth.01*, 0:0, 1:1
4. *snub.01*, 1:1
5. *erect.01*, 0:0, 1:1

Given this list, arbitrary amounts of PropBank annotations can be used as surrogate training data for the *instigate.01* roleset, beginning at the top of the list. To utilize surrogate training data in our semantic role labeling approach (Section 3), we combine parse tree path information for a selected portion of surrogate training data into a single list sorted by frequency, and apply these files to test sentences as normal.

Although we use an existing PropBank roleset (*instigate.01*) as an example in this section, this approach will work for any novel roleset where one fully-annotated training example is available. For example, arbitrary amounts of surrogate PropBank data can be found for the novel verb *yearn* by 1) searching for sentences with the verb *yearn* in the GigaWord corpus, 2) calculating the syntactic similarity between *yearn* and all PropBank verbs as described in Section 4, 3) aligning the arguments in a single fully-annotated example of *yearn* with ProbBank rolesets with the same number of arguments using the method described in this section, and 4) selecting arbitrary amounts of PropBank annotations to use as surrogate training data, starting from the top of the resulting list.

6 Evaluation

We conducted a large-scale evaluation to determine the performance of our semantic role labeling algorithm when using variable amounts of surrogate training data, and compared these results to the performance that could be obtained using various amounts of real training data (as described in section 3). Our hypothesis was that learning-curves for surrogate-trained labelers would be somewhat less steep, but that the availability of large-amounts of surrogate training data would more than make up for the gap.

To test this hypothesis, we conducted an evaluation using the PropBank corpus as our testing data as well as our source for surrogate training data. As described in section 5, our approach requires the availability of at least one fully-annotated sentence for a given roleset. Only 28.5% of the PropBank annotations assign labels for each of the numbered arguments in their given roleset, and only 2,858 of the 4,250 rolesets used in PropBank annotations (66.5%) have at least one fully-annotated sentence. Of these, 2,807 rolesets were for verbs that appeared at least once in our analysis of the GigaWord corpus (Section 4). Accordingly, we evaluated our approach using the annotations for this set of 2,807 rolesets as test data. For each of these rolesets, various amounts of surrogate training data were gathered from all 4,250 rolesets represented in PropBank, leaving out the data for whichever roleset was being tested.

For each of the target 2,807 rolesets, we generated a list of semantic role mappings ordered by syntactic similarity, using the methods described in sections 4 and 5. In aligning arguments, only a single training example from the target roleset was used, namely the first annotation within the PropBank corpus where all of the rolesets arguments were assigned. Our approach failed to identify any argument mappings for 41 of the target rolesets, leaving them without any surrogate training data to utilize. Of the remaining 2,766 rolesets, the number of mapped rolesets for a given target ranged from 1,041 to 1 (mean = 608, stdev = 297).

For each of the 2,766 target rolesets with alignable roles, we gathered increasingly larger amounts of surrogate training data by descending the ordered list of mappings translating the PropBank data for each entry according to its argument mappings. Then each of these incrementally larger sets of training data was then used to build a semantic role labeler as described in section 3. The performance of each of the resulting labelers was then evaluated by applying it to all of the test data available for target roleset in PropBank, using the same scoring methods described in section 3. The performance scores for each labeler were recorded along with the total number of surrogate training examples used to build the labeler.

Figure 3 presents the performance result of our semantic role labeling approach using various amounts of surrogate training data. Along with precision, recall, and F-score data, Figure 3 also graphs the percentage of PropBank rolesets for which a given amount of training data had been identified using our approach, of the 2,858 rolesets with at least one fully-annotated training example. For instance, with 120 surrogate annotations our system achieves an F-score above 0.5, and we identified this much surrogate training data for 96% of PropBank rolesets with at least one fully-annotated sentence. This represents 64% of all PropBank rolesets that are used for annotation. Beyond 120 surrogate training examples, F-scores remain around 0.6 before slowly declining after around 700 examples.

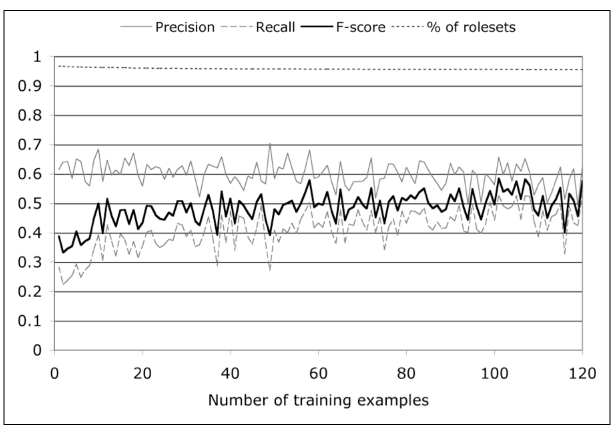

Figure 3. Performance of our semantic role labeling approach on PropBank rolesets using various amounts of surrogate training data

Several interesting comparisons can be made between the results presented in Figure 3 and those in Figure 2, where actual PropBank training data is used instead of surrogate training data. First, the precision obtained with surrogate training data is roughly 10% lower than with real data. Second, the recall performance of surrogate data performs similar to real data at first, but is consistently 10% lower than with real data after the first 50 training examples. Accordingly, F-scores for surrogate training data are 10% lower overall.

Even though the performance obtained using surrogate training data is less than with actual data, there is abundant amounts of it available for most PropBank rolesets. Comparing the "% of rolesets" plots in Figures 2 and 3, the real value of surrogate training data is apparent. Figure 2 suggests that over 20 real training examples are needed to achieve F-scores that are consistently above 0.5, but that less than 20% of PropBank rolesets have this much data available. In contrast, 64% of all PropBank rolesets can achieve this F-score performance with the use of surrogate training data. This percentage increases to 96% if every PropBank roleset is given at least one fully annotated sentence, where all of its numbered arguments are assigned to constituents.

In addition to supplementing the real training data available for existing PropBank rolesets, these results predict the labeling performance that can be obtained by applying this technique to a novel roleset with one fully-annotated training example, e.g. for the verb *yearn*. Using the first 120 surrogate training examples and our simple semantic role labeling approach, we would expect F-scores that are above 0.5, and that using the first 700 would yield F-scores around 0.6.

7 Discussion

The overall performance of our semantic role labeling approach is not competitive with leading contemporary systems, which typically employ support vector machine learning algorithms with syntactic features (Pradhan et al., 2005) or syntactic tree kernels (Moschitti et al., 2006). However, our work highlights a number of characteristics of the semantic role labeling task that will be helpful in improving performance in future systems. Parse tree paths features can be used to achieve high precision in semantic role labeling, but much of this precision may be specific to individual verbs. By generalizing parse tree path features only across syntactically similar verbs, we have shown that the drop in precision can be limited to roughly 10%.

The approach that we describe in this paper is not dependent on the use of PropBank rolesets; any large corpus of semantic role annotations could be generalized in this manner. In particular, our approach would be applicable to corpora with frame-specific role labels, e.g. FrameNet (Baker et al., 1998). Likewise, our approach to generalizing parse tree path feature across syntactically similar verbs may improve the performance of automated semantic role labeling systems based on FrameNet data. Our work suggests that feature generalization based on verb-similarity may compliment approaches to generalization based on role-similarity (Gildea and Jurafsky, 2002; Baldewein et al., 2004).

There are a number of improvements that could be made to the approach described in this paper. Enhancements to the simple semantic role labeling algorithm would improve the alignment of arguments across rolesets, which would help align rolesets with greater syntactic similarity, as well as improve the performance obtained using the surrogate training data in assigning semantic roles.

This research raises many questions about the relationship between syntactic context and verb semantics. An important area for future research will be to explore the correlation between our distance metric for syntactic similarity and various quantitative measures of semantic similarity (Pedersen, et al., 2004). Particularly interesting would be to explore whether different senses of a given verb exhibited markedly different profiles of syntactic context. A strong syntactic/semantic correlation would suggest that further gains in the use of surrogate annotation data could be gained if syntactic similarity was computed between rolesets rather than their verbs. However, this would first require accurate word-sense disambiguation both for the test sentences as well as for the parsed corpora used to calculate parse tree path frequencies. Alternatively, parse tree path profiles associated with rolesets may be useful for word sense disambiguation, where the probability of a sense is computed as the likelihood that an ambiguous verb's parse tree paths are sampled from the distributions associated with each verb sense. These topics will be the focus of our future work in this area.

Acknowledgments

The project or effort depicted was or is sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and that the content or information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Baker, C., Fillmore, C., and Lowe, J. 1998. The Berkeley FrameNet Project, In Proceedings of COLING-ACL, Montreal.

Baldewein, U., Erk, K., Pado, S., and Prescher, D. 2004. Semantic role labeling with similarity-based generalization using EM-based clustering. Proceedings of Senseval-3, Barcelona.

Charniak, E. 2000. A maximum-entropy-inspired parser, Proceedings NAACL-ANLP, Seattle.

Courtois, B. 2004. Dictionnaires électroniques DELAF anglais et français. In C. Leclère, E. Laporte, M. Piot and M. Silberztein (eds.) Syntax, Lexis and Lexicon-Grammar: Papers in Honour of Maurice Gross. Amsterdam: John Benjamins.

Gildea, D. and Jurafsky, D. 2002. Automatic Labeling of Semantic Roles. Computational Linguistics 28:3, 245-288.

Joanis, E. and Stevenson, S. 2003. A general feature space for automatic verb classification. Proceedings EACL, Budapest.

Levin, B. 1993. English Verb Classes and Alterna-tions: A Preliminary Investigation. Chicago, IL: University of Chicago Press.

Lin, D. 1998. Automatic Retrieval and Clustering of Similar Words. COLING-ACL, Montreal.

Linguistic Data Consortium. 2003. English Gigaword. Catalog number LDC2003T05. Available from LDC at http://www.ldc.upenn.edu.

Moschitti, A., Pighin, D. and Basili, R. 2006. Semantic Role Labeling via Tree Kernel joint inference. Proceedings of CoNLL, New York.

Palmer, M., Gildea, D., and Kingsbury, P. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics 31(1):71-106.

Pedersen, T., Patwardhan, S. and Michelizzi, J. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. Proceedings NAACL-04, Boston, MA.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., and Jurafsky, D. 2005. Semantic role labeling using different syntactic views. Proceedings ACL-2005, Ann Arbor, MI.

Reynar, J. and Ratnaparkhi, A. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. Proceedings of ANLP, Washington, D.C.