

Semantic Role Chunking Combining Complementary Syntactic Views

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin and Daniel Jurafsky[†]

Center for Spoken Language Research, University of Colorado, Boulder, CO 80303

[†]Department of Linguistics, Stanford University, Stanford, CA 94305

{spradhan,hacioglu,whw,martin}@cslr.colorado.edu, jurafsky@stanford.edu

Abstract

This paper describes a semantic role labeling system that uses features derived from different syntactic views, and combines them within a phrase-based chunking paradigm. For an input sentence, syntactic constituent structure parses are generated by a Charniak parser and a Collins parser. Semantic role labels are assigned to the constituents of each parse using Support Vector Machine classifiers. The resulting semantic role labels are converted to an IOB representation. These IOB representations are used as additional features, along with flat syntactic chunks, by a chunking SVM classifier that produces the final SRL output. This strategy for combining features from three different syntactic views gives a significant improvement in performance over roles produced by using any one of the syntactic views individually.

1 Introduction

The task of Semantic Role Labeling (SRL) involves tagging groups of words in a sentence with the semantic roles that they play with respect to a particular predicate in that sentence. Our approach is to use supervised machine learning classifiers to produce the role labels based on features extracted from the input. This approach is neutral to the particular set of labels used, and will learn to tag input according

to the annotated data that it is trained on. The task reported on here is to produce PropBank (Kingsbury and Palmer, 2002) labels, given the features provided for the CoNLL-2005 closed task (Carreras and Màrquez, 2005).

We have previously reported on using SVM classifiers for semantic role labeling. In this work, we formulate the semantic labeling problem as a multi-class classification problem using Support Vector Machine (SVM) classifiers. Some of these systems use features based on syntactic constituents produced by a Charniak parser (Pradhan et al., 2003; Pradhan et al., 2004) and others use only a flat syntactic representation produced by a syntactic chunker (Hacioglu et al., 2003; Hacioglu and Ward, 2003; Hacioglu, 2004; Hacioglu et al., 2004). The latter approach lacks the information provided by the hierarchical syntactic structure, and the former imposes a limitation that the possible candidate roles should be one of the nodes already present in the syntax tree. We found that, while the chunk based systems are very efficient and robust, the systems that use features based on full syntactic parses are generally more accurate. Analysis of the source of errors for the parse constituent based systems showed that incorrect parses were a major source of error. The syntactic parser did not produce any constituent that corresponded to the correct segmentation for the semantic argument. In Pradhan et al. (2005), we reported on a first attempt to overcome this problem by combining semantic role labels produced from different syntactic parses. The hope is that the syntactic parsers will make different errors, and that combining their outputs will improve on

either system alone. This initial attempt used features from a Charniak parser, a Minipar parser and a chunk based parser. It did show some improvement from the combination, but the method for combining the information was heuristic and sub-optimal. In this paper, we report on what we believe is an improved framework for combining information from different syntactic views. Our goal is to preserve the robustness and flexibility of the segmentation of the phrase-based chunker, but to take advantage of features from full syntactic parses. We also want to combine features from different syntactic parses to gain additional robustness. To this end, we use features generated from a Charniak parser and a Collins parser, as supplied for the CoNLL-2005 closed task.

2 System Description

We again formulate the semantic labeling problem as a multi-class classification problem using Support Vector Machine (SVM) classifiers. TinySVM¹ along with YamCha² (Kudo and Matsumoto, 2000; Kudo and Matsumoto, 2001) are used to implement the system. Using what is known as the ONE VS ALL classification strategy, n binary classifiers are trained, where n is number of semantic classes including a NULL class.

The general framework is to train separate semantic role labeling systems for each of the parse tree views, and then to use the role arguments output by these systems as additional features in a semantic role classifier using a flat syntactic view. The constituent based classifiers walk a syntactic parse tree and classify each node as NULL (no role) or as one of the set of semantic roles. Chunk based systems classify each base phrase as being the B(eginning) of a semantic role, I(nside) a semantic role, or O(utside) any semantic role (ie. NULL). This is referred to as an IOB representation (Ramshaw and Marcus, 1995). The constituent level roles are mapped to the IOB representation used by the chunker. The IOB tags are then used as features for a separate base-phase semantic role labeler (chunker), in addition to the standard set of features used by the chunker. An n -fold cross-validation paradigm is used to train the constituent based role classifiers

and the chunk based classifier.

For the system reported here, two full syntactic parsers were used, a Charniak parser and a Collins parser. Features were extracted by first generating the Collins and Charniak syntax trees from the word-by-word decomposed trees in the CoNLL data. The chunking system for combining all features was trained using a 4-fold paradigm. In each fold, separate SVM classifiers were trained for the Collins and Charniak parses using 75% of the training data. That is, one system assigned role labels to the nodes in Charniak based trees and a separate system assigned roles to nodes in Collins based trees. The other 25% of the training data was then labeled by each of the systems. Iterating this process 4 times created the training set for the chunker. After the chunker was trained, the Charniak and Collins based semantic labelers were then retrained using all of the training data.

Two pieces of the system have problems scaling to large training sets – the final chunk based classifier and the NULL VS NON-NULL classifier for the parse tree syntactic views. Two techniques were used to reduce the amount of training data – active sampling and NULL filtering. The active sampling process was performed as follows. We first train a system using 10k seed examples from the training set. We then labeled an additional block of data using this system. Any sentences containing an error were added to the seed training set. The system was retrained and the procedure repeated until there were no misclassified sentences remaining in the training data. The set of examples produced by this procedure was used to train the final NULL VS NON-NULL classifier. The same procedure was carried out for the chunking system. After both these were trained, we tagged the training data using them and removed all most likely NULLs from the data.

Table 1 lists the features used in the constituent based systems. They are a combination of features introduced by Gildea and Jurafsky (2002), ones proposed in Pradhan et al. (2004), Surdeanu et al. (2003) and the *syntactic-frame* feature proposed in (Xue and Palmer, 2004). These features are extracted from the parse tree being labeled. In addition to the features extracted from the parse tree being labeled, five features were extracted from the other parse tree (phrase, head word, head word POS, path

¹<http://chasen.org/~taku/software/TinySVM/>

²<http://chasen.org/~taku/software/yamcha/>

PREDICATE LEMMA
PATH: Path from the constituent to the predicate in the parse tree.
POSITION: Whether the constituent is before or after the predicate.
PREDICATE SUB-CATEGORIZATION
HEAD WORD: Head word of the constituent.
HEAD WORD POS: POS of the head word
NAMED ENTITIES IN CONSTITUENTS: Person, Organization, Location and Miscellaneous.
PARTIAL PATH: Path from the constituent to the lowest common ancestor of the predicate and the constituent.
HEAD WORD OF PP: Head of PP replaced by head word of NP inside it, and PP replaced by PP-preposition
FIRST AND LAST WORD/POS IN CONSTITUENT
ORDINAL CONSTITUENT POSITION
CONSTITUENT TREE DISTANCE
CONSTITUENT RELATIVE FEATURES: 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.
SYNTACTIC FRAME
CONTENT WORD FEATURES: Content word, its POS and named entities in the content word
CLAUSE-BASED PATH VARIATIONS: I. Replacing all the nodes in a path other than clause nodes with an “*”. For example, the path NP↑S↑VP↑SBAR↑NP↓VP↓VBD becomes NP↑S↑*S↑*S↑*↓VBD II. Retaining only the clause nodes in the path, which for the above example would produce NP↑S↑S↓VBD. III. Adding a binary feature that indicates whether the constituent is in the same clause as the predicate, IV. collapsing the nodes between S nodes which gives NP↑S↑NP↓VP↓VBD.
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, etc. We used the first ten trigrams as ten features. Shorter paths were padded with nulls.
SINGLE CHARACTER PHRASE TAGS: Each phrase category is clustered to a category defined by the first character of the phrase label.
PREDICATE CONTEXT: Two words and two word POS around the predicate and including the predicate were added as ten new features.
PUNCTUATION: Punctuation before and after the constituent were added as two new features.
FEATURE CONTEXT: Features for argument bearing constituents were added as features to the constituent being classified.

Table 1: Features used by the constituent-based system

and predicate sub-categorization). So for example, when assigning labels to constituents in a Charniak parse, all of the features in Table 1 were extracted from the Charniak tree, and in addition phrase, head word, head word POS, path and sub-categorization were extracted from the Collins tree. We have previously determined that using different sets of features for each argument (role) achieves better results than using the same set of features for all argument classes. A simple feature selection was implemented by adding features one by one to an initial set of features and selecting those that contribute significantly to the performance. As described in Pradhan et al. (2004), we post-process lattices of n-best decision using a trigram language model of argument sequences.

Table 2 lists the features used by the chunker. These are the same set of features that were used

in the CoNLL-2004 semantic role labeling task by Hacıoglu, et al. (2004) with the addition of the two semantic argument (IOB) features. For each token (base phrase) to be tagged, a set of features is created from a fixed size context that surrounds each token. In addition to the features in Table 2, it also uses previous semantic tags that have already been assigned to the tokens contained in the linguistic context. A 5-token sliding window is used for the context.

SVMs were trained for begin (B) and inside (I) classes of all arguments and an outside (O) class.

WORDS
PREDICATE LEMMAS
PART OF SPEECH TAGS
BP POSITIONS: The position of a token in a BP using the IOB2 representation (e.g. B-NP, I-NP, O, etc.)
CLAUSE TAGS: The tags that mark token positions in a sentence with respect to clauses.
NAMED ENTITIES: The IOB tags of named entities.
TOKEN POSITION: The position of the phrase with respect to the predicate. It has three values as “before”, “after” and “-” (for the predicate)
PATH: It defines a flat path between the token and the predicate
HIERARCHICAL PATH: Since we have the syntax tree for the sentences, we also use the hierarchical path from the phrase being classified to the base phrase containing the predicate.
CLAUSE BRACKET PATTERNS
CLAUSE POSITION: A binary feature that identifies whether the token is inside or outside the clause containing the predicate
HEADWORD SUFFIXES: suffixes of headwords of length 2, 3 and 4.
DISTANCE: Distance of the token from the predicate as a number of base phrases, and the distance as the number of VP chunks.
LENGTH: the number of words in a token.
PREDICATE POS TAG: the part of speech category of the predicate
PREDICATE FREQUENCY: Frequent or rare using a threshold of 3.
PREDICATE BP CONTEXT: The chain of BPs centered at the predicate within a window of size -2/+2.
PREDICATE POS CONTEXT: POS tags of words immediately preceding and following the predicate.
PREDICATE ARGUMENT FRAMES: Left and right core argument patterns around the predicate.
DYNAMIC CLASS CONTEXT: Hypotheses generated for two preceding phrases.
NUMBER OF PREDICATES: This is the number of predicates in the sentence.
CHARNIAK-BASED SEMANTIC IOB TAG: This is the IOB tag generated using the tagger trained on Charniak trees
COLLINS-BASED SEMANTIC IOB TAG: This is the IOB tag generated using the tagger trained on Collins’ trees

Table 2: Features used by phrase-based chunker.

3 Experimental Results

Table 3 shows the results obtained on the WSJ development set (Section 24), the WSJ test set (Section 23) and the Brown test set (Section ck/01-03)

4 Acknowledgments

This research was partially supported by the ARDA AQUAINT program via contract OCG4423B and by the NSF via grants IS-9978025 and ITR/HCI

	Precision	Recall	$F_{\beta=1}$
Development	80.90%	75.38%	78.04
Test WSJ	81.97%	73.27%	77.37
Test Brown	73.73%	61.51%	67.07
Test WSJ+Brown	80.93%	71.69%	76.03

Test WSJ	Precision	Recall	$F_{\beta=1}$
Overall	81.97%	73.27%	77.37
A0	91.39%	82.23%	86.57
A1	79.80%	76.23%	77.97
A2	68.61%	62.61%	65.47
A3	73.95%	50.87%	60.27
A4	78.65%	68.63%	73.30
A5	75.00%	60.00%	66.67
AM-ADV	61.64%	46.05%	52.71
AM-CAU	76.19%	43.84%	55.65
AM-DIR	53.33%	37.65%	44.14
AM-DIS	80.56%	63.44%	70.98
AM-EXT	100.00%	46.88%	63.83
AM-LOC	64.48%	51.52%	57.27
AM-MNR	62.90%	45.35%	52.70
AM-MOD	98.64%	92.38%	95.41
AM-NEG	98.21%	95.65%	96.92
AM-PNC	56.67%	44.35%	49.76
AM-PRD	0.00%	0.00%	0.00
AM-REC	0.00%	0.00%	0.00
AM-TMP	83.37%	71.94%	77.23
R-A0	94.29%	88.39%	91.24
R-A1	85.93%	74.36%	79.73
R-A2	100.00%	37.50%	54.55
R-A3	0.00%	0.00%	0.00
R-A4	0.00%	0.00%	0.00
R-AM-ADV	0.00%	0.00%	0.00
R-AM-CAU	0.00%	0.00%	0.00
R-AM-EXT	0.00%	0.00%	0.00
R-AM-LOC	90.00%	42.86%	58.06
R-AM-MNR	66.67%	33.33%	44.44
R-AM-TMP	75.00%	40.38%	52.50
\bar{V}	98.86%	98.86%	98.86

Table 3: Overall results (top) and detailed results on the WSJ test (bottom).

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.

Special thanks to Matthew Woitaszek, Theron Voran and the other administrative team of the Hemisphere and Occam Beowulf clusters. Without these the training would never be possible.

References

- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*.
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.
- Kadri Hacioglu and Wayne Ward. 2003. Target word detection and semantic role chunking using support vector machines. In *Proceedings of the Human Language Technology Conference*, Edmonton, Canada.
- Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James Martin, and Dan Jurafsky. 2003. Shallow semantic parsing using support vector machines. Technical Report TR-CSLR-2003-1, Center for Spoken Language Research, Boulder, Colorado.
- Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James Martin, and Daniel Jurafsky. 2004. Semantic role labeling by tagging syntactic chunks. In *Proceedings of the 8th Conference on CoNLL-2004, Shared Task – Semantic Role Labeling*.
- Kadri Hacioglu. 2004. A lightweight semantic chunking model based on tagging. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association of Computational Linguistics (HLT/NAACL)*, Boston, MA.
- Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Canary Islands, Spain.
- Taku Kudo and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. In *Proceedings of the 4th Conference on CoNLL-2000 and LLL-2000*, pages 142–144.
- Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*.
- Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James Martin, and Dan Jurafsky. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM 2003)*, Melbourne, Florida.
- Sameer Pradhan, 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)*, Boston, MA.
- Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the Association for Computational Linguistics 43rd annual meeting (ACL-2005)*, Ann Arbor, MI.
- L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.
- Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.
- Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.