

Semantic Role Labeling Using Dependency Trees

Kadri Hacioglu

Center for Spoken Language Research

University of Colorado at Boulder

hacioglu@cslr.colorado.edu

Abstract

In this paper, a novel semantic role labeler based on dependency trees is developed. This is accomplished by formulating the semantic role labeling as a classification problem of dependency relations into one of several semantic roles. A dependency tree is created from a constituency parse of an input sentence. The dependency tree is then linearized into a sequence of dependency relations. A number of features are extracted for each dependency relation using a predefined linguistic context. Finally, the features are input to a set of one-versus-all support vector machine (SVM) classifiers to determine the corresponding semantic role label. We report results on CoNLL2004 shared task data using the representation and scoring scheme adopted for that task.

1 Introduction

In semantic role labeling (SRL) the goal is to group sequences of words together and classify them by using semantic labels. For semantic representation we select the predicate-argument structure that exists in most languages. In this structure a word is specified as a predicate and a number of word groups are considered as arguments accompanying the predicate. Those arguments are assigned different semantic categories depending on the roles that they play with respect to the predicate.

We illustrate the predicate-argument structure in Figure 1 for the sentence “We are prepared to pursue aggressively completion of this transaction he says” taken from the PropBank corpus. The chosen predicate is the word *pursue*, and its arguments with their associated word groups are illustrated. Note that the word *prepared* is another predicate of the sentence possibly with different argument labels attached to the same or different word groups. For example, the word *we* is A1 of *prepared*. This process of selecting a predicate in a sentence, grouping sequences of words and assigning the semantic roles they play with respect to the chosen predicate is often referred to as semantic role labeling. We believe that a highly accurate extraction of this structure is vital for high performance in many NLP tasks such as information extraction, question answering, summarization and machine translation.

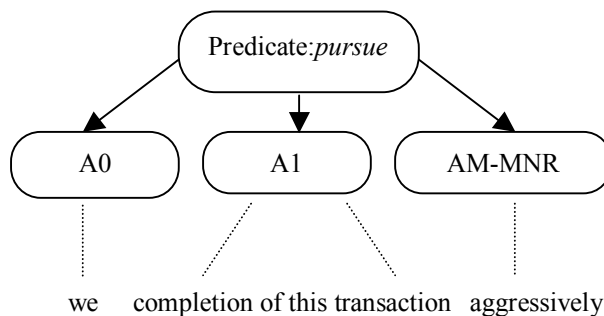


Figure 1: Predicate-argument structure of sample sentence. Argument labels are in PropBank-style.

Semantic role labeling based on predicate-argument structure was first explored in detail by (Gildea and Jurafsky, 2002). Since then several variants of the basic approach have been introduced using different features and different classifiers based on various machine learning techniques (Gildea and Palmer, 2002; Gildea and Hockenmaier, 2003; Surdeanu et. al., 2003; Chen and Rambow, 2003; Fleischman and Hovy, 2003; Hacioglu and Ward, 2003; Thompson et. al., 2003; Pradhan et. al., 2003b; Hacioglu, 2004). Large semantically annotated databases, like FrameNet (Baker et.al, 1998) and PropBank (Kingsbury and Palmer, 2002) have been used to train and test the classifiers. Most of those approaches can be divided into one of the following three broad classes with respect to the type of tokens classified; namely, constituent-by-constituent (C-by-C), phrase-by-phrase (P-by-P) and word-by-word (W-by-W) semantic role labelers.

In C-by-C semantic role labeling, the syntactic tree representation of a sentence is linearized into a sequence of its syntactic constituents (non-terminals). Then each constituent is classified into one of several semantic roles using a number of features derived from the sentence structure or a linguistic context defined for the constituent token. In the P-by-P and W-by-W methods (Hacioglu, 2004; Hacioglu and Ward, 2003) the problem is formulated as a chunking task and the features are derived for each base phrase and word, respectively. The tokens were classified into one of the semantic labels using an IOB (inside-outside-begin) representation and a bank of SVM classifiers; a one-versus-all classifier has been used for each class.

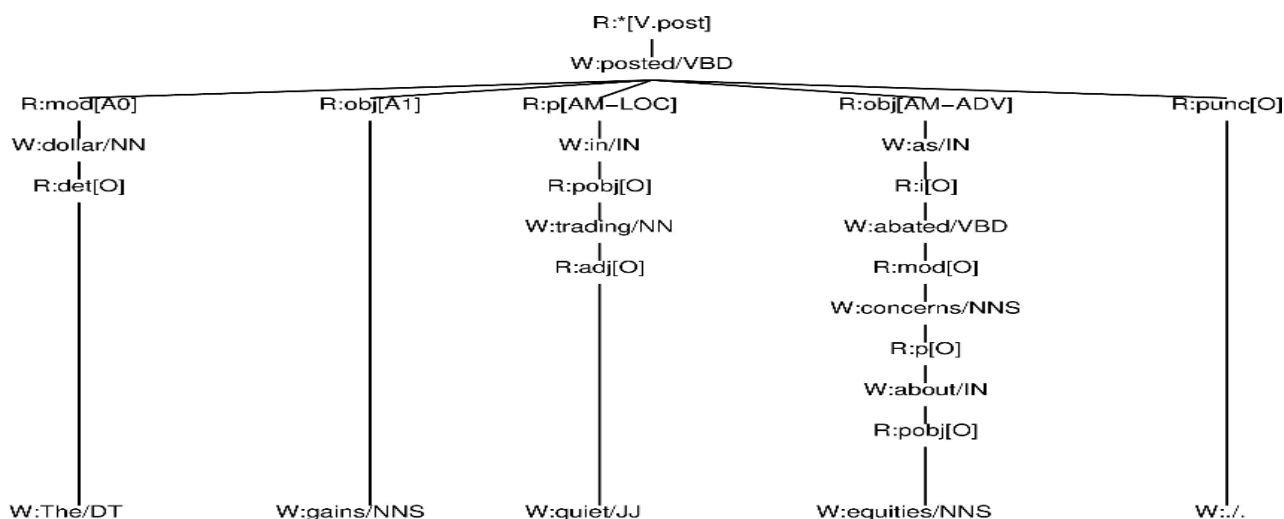


Figure 2. Example of a dependency tree augmented with semantic roles. Semantic labels correspond to the predicate *posted*. The same tree with different semantic labels also exists in the corpus for predicate *abated*.

In this paper, we introduce another approach that we refer to as the relation-by-relation (R-by-R) semantic role labeling. The method is based on dependency trees generated from constituency trees. Although the system currently does not use more information than C-by-C systems, the information is structured in a different manner and, consequently, the nature of some linguistic features is quite different. We point out that this information restructuring is very useful in localizing the semantic roles associated with the selected predicate, since the dependency trees directly encode the argument structure of lexical units populated at their nodes through dependency relations.

A related work is reported in (Gildea and Hockenmaier, 2003). However, they use Combinatory Categorical Grammar (CCG) to derive the dependency relations. In addition, our method differs in the selection of dependency relations for labeling, in the creation of features and in the implementation of the classifier.

Recently, there has been some interest in developing a deterministic machine-learning based approach for dependency parsing (Yamada and Matsumoto, 2003). In addition to relatively easier portability to other domains and languages the deterministic dependency parsing promises algorithms that are robust and efficient. Therefore, an SRL algorithm based on dependency structures is expected to benefit from those properties.

2 Dependency Bank (*DepBank*)

In this section, we describe the corpus that we automatically created using the syntactic annotations of the Penn TreeBank with the semantic annotations of the PropBank. Hereafter, we refer to this new corpus as *DepBank*.

Firstly, we convert constituency trees into dependency trees¹. The functional tags are removed from constituency trees before the conversion, since the current state-of-the-art syntactic parsers do not exploit those tags. Secondly, we trace the dependency trees to determine the word sequences covered by the dependency relation nodes. Finally, we augment those nodes with their semantic role labels that cover the same sequence of words. The relations that do not align with any semantic role are tagged using the label “O”. In Figure 2, we illustrate a sample dependency tree from the DepBank. It corresponds to the predicate *posted* of the following sentence (semantic roles are also indicated):

[A0 The dollar] [V posted] [A1 gains] [AM-LOC in quiet training] [AM-ADV as concerns about equities abated]

We note that the other predicate in the sentence is *abated* and the same tree with different semantic labels is also instantiated in the DepBank for it. The dependency relation nodes are indicated by “R:” in Figure 2. The lexical nodes are indicated by “W:”. The dependency relation types are paired with the corresponding semantic role labels. The only exception is the node that belongs to the predicate; the semantic label V is used with the lemma of the predicate. The lexical nodes include the word itself and its part-of-speech (POS) tag.

3 Semantic Role Labeling of Relations

In the proposed approach, we first linearize the dependency tree in a bottom-up left-to-right manner into a sequence of dependency relations. During this

¹ *engconst2dep*, from the University of Maryland, is used. Special thanks to R. Hwa, A. Lopez and M. Diab.

process we filter out the dependency relations that are less likely to be an argument. The selection mechanism is based on simple heuristics derived from dependency trees. Then we extract a set of features for each dependency relation. Finally, we input the features to a bank of SVM classifiers. A one-versus-all SVM classifier is used for each semantic role.

3.1 Dependency Relation Selection

In dependency tree representations, we observe that the semantic roles are highly localized with respect to the chosen predicate. We exploit this observation to devise a method for deciding whether a dependency relation is likely to be a semantic role or not. We define a tree-structured *family* of a predicate as a measure of locality. It is a set of dependency relation nodes that consists of the predicate's parent, children, grandchildren, siblings, siblings' children and siblings' grandchildren with respect to its dependency tree. Any relation that does not belong to this set is skipped while we linearize the dependency tree in a bottom-up left-to-right manner. Further selection is performed on the family members that are located at the leaves of the tree. For example, a leaf member with *det* dependency relation is not considered for semantic labeling. Our selection mechanism reduces the data for semantic role labeling by approximately 3-4 fold with nearly 1% miss of semantic labels, since a quite large number of nodes in the dependency trees are not associated with any semantic role.

3.2 Features

For each candidate dependency relation we extract a set of features. In the following, we explain these features and give examples for their values referring to the dependency tree shown in Figure 1 (feature values for the relation node R:mod with the semantic label [A0] is given in parentheses). The features that are specific to the dependency relation (i.e. token-level features) are

Type: This feature indicates the type of the dependency relation (**mod**)

Family membership: This feature indicates how the dependency relation is related to the predicate in the *family* (**child**)

Position: This feature indicates the position of the headword of the dependency relation with respect to the predicate position in the sentence (**before**)

Headword: the modified (head) word in the relation (**posted**).

Dependent word: the modifying word in the relation (**dollar**)

POS tag of headword: (VBD)

POS tag of dependent word: (NN)

Path: the chain of relations from relation node to predicate. (**mod**→*)

and the features that are specific to the predicate (i.e. sentence-level features):

POS pattern of predicate's children: This feature indicates the left-to-right chain of the POS tags of the immediate words that depend on the predicate. (NN-NNS-IN-IN)

Relation pattern of predicate's children: This feature indicates the left-to-right chain of the relation labels of the predicate's children (**mod-obj-p-obj**)

POS pattern of predicate's siblings: This feature indicates the left-to-right chain of the POS tags of the headwords of the siblings of predicate. (-)

Relation pattern of predicate's siblings: This feature indicates the left-to-right chain of the relation labels of the predicate's siblings. (-).

3.3 Classifier

We selected support vector machines (Vapnik, 1995) to implement the semantic role classifiers. The motivation for this selection was the ability of SVMs to handle an extremely large number of interacting or overlapping features with quite strong generalization properties. Support vector machines for SRL were first used in (Hacioglu and Ward, 2003) as word-by-word (W-by-W) classifiers. The system was then applied to the constituent-by-constituent (C-by-C) classification in (Hacioglu et. al., 2003) and phrase-by-phrase (P-by-P) classification in (Hacioglu, 2004). Several extensions of the basic system with state-of-the-art performance were reported in (Pradhan et.al, 2003; Pradhan et. al. 2004; Hacioglu et. al. 2004). All SVM classifiers for semantic argument labeling were realized using the TinySVM with a polynomial kernel of degree 2 and the general purpose SVM based chunker YamCha².

4 Experiments

Experiments were carried out using a part of the February 2004 release of the PropBank. Sections 15 through 18 were used for training, Section 20 was used for developing and Section 21 was used for testing. This is exactly the same data used for CoNLL2004 shared task on SRL. Therefore, the results can be directly compared to the performance of the systems that used or that will use the same data. The system performance is evaluated by using precision, recall and F metrics. In the experiments,

² <http://cl.aist-nara.ac.jp/~taku-ku/software/>

the gold standard constituency parses were used. Therefore, the results provide an upper bound on the performance with automatic parses. Table 1 presents the results on the *DepBank* development set. The results on the CoNLL2004 development set are also illustrated. After we project the predicted semantic role labels in the *DepBank* dev set onto the CoNLL2004 dev set (directly created from the PropBank) we observe a sharp drop in the recall performance. The drop is due to the loss of approximately 8% of semantic roles in the *DepBank* dev set during the conversion process; not all phrase nodes in constituency trees find an equivalent relation node in dependency trees. However, this mismatch is significantly less than the 23% mismatch reported in (Gildea and Hockenmaier, 2003) between the CCGBank and an earlier version of the PropBank.

Dev Set	Precision	Recall	F ₁
<i>DepBank</i>	85.6%	83.6%	84.6
CoNLL	84.9%	75.2%	79.8

Table 1: Results on *DepBank* and CoNLL04 sets.

5 Conclusions

We have automatically created a new corpus of dependency trees augmented with semantic role labels. Using this corpus, we have developed and experimented with a novel SRL system that classifies dependency relations. This is quite different from previous research on semantic role labeling. We have presented encouraging intermediate results. Currently, we are investigating the reasons of mismatch between PropBank and *DepBank* semantic annotations. We also plan to add new features, experiment with automatic parses, and compare and combine the system with our state-of-the-art C-by-C system.

Acknowledgements

This research was supported in part by the ARDA AQUAINT Program via contract OCG4423B and by the NSF grant ISS-9978025.

References

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe 1998. The Berkley FrameNet Project. In *Proc. of CoLING-ACL'98*.
- John Chen and Owen Rambow. 2003. Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments. In *Proc. of EMNLP-2003*
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28:3, pages 245-288.
- Daniel Gildea and Martha Palmer. 2002. The Necessity of Syntactic Parsing for Predicate Argument Recognition. In *Proc. of ACL'02*.
- Daniel Gildea and Julia Hockenmaier. 2003. Identifying Semantic Roles Using Combinatory Categorical Grammar. In *Proc. of EMNL'03, Japan*.
- Micheal Fleischman and Eduard Hovy. 2003. A Maximum Entropy Approach to FrameNet Tagging. In *Proc. of HLT/NAACL-03*.
- Kadri Hacioglu and Wayne Ward. 2003. Target word detection and semantic role chunking using support vector machines. In *Proc. of HLT/NAACL-03*.
- Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin and Daniel Jurafsky. 2003. Shallow Semantic Parsing using Support Vector Machines. *CSLR Technical Report*, CSLR-TR-2003-1.
- Kadri Hacioglu. 2004. A Semantic Chunking Model Based on Tagging. In *Proc. of HLT/NAACL-04*.
- Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin and Daniel Jurafsky. 2004. Semantic Role Labeling by Tagging Syntactic Chunks. *CONLL-2004 Shared Task*.
- Paul Kingsbury, Martha Palmer, 2002. From TreeBank to PropBank. In *Proc. of LREC-2002*.
- Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, Dan Jurafsky. 2003. Semantic Role Parsing: Adding Semantic Structure to Unstructured Text. In *Proc. of ICDM 2003*.
- Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, Dan Jurafsky. 2004. Support Vector Learning for Semantic Argument Classification. To appear in *Journal of Machine Learning*.
- Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using Predicate-Argument Structure for Information Extraction. In *Proc. of ACL03*.
- Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. 2003. A Generative Model for Semantic Role Labeling. In *Proc. of ECML-03*.
- Vladimir Vapnik 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.
- Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proc. of IWPT'03*.