A transformation-based approach to argument labeling

Derrick Higgins
Educational Testing Service
Mail Stop 12-R
Rosedale Road
Princeton, NJ 08541
dhiggins@ets.org

Abstract

This paper presents the results of applying transformation-based learning (TBL) to the problem of semantic role labeling. The great advantage of the TBL paradigm is that it provides a simple learning framework in which the parallel tasks of argument identification and argument labeling can mutually influence one another. Semantic role labeling nevertheless differs from other tasks in which TBL has been successfully applied, such as part-of-speech tagging and named-entity recognition, because of the large span of some arguments, the dependence of argument labels on global information, and the fact that core argument labels are largely arbitrary. Consequently, some care is needed in posing the task in a TBL framework.

1 Overview

In the closed challenge of the CoNLL shared task, the system is charged with both identifying argument boundaries, and correctly labeling the arguments with the correct semantic role, without using a parser to suggest candidate phrases. Transformation-based learning (Brill, 1995) is well-suited to simultaneously addressing this dual task of identifying and labeling semantic arguments of a predicate, because it allows intermediate hypotheses to influence the ultimate decisions made. More concretely, the category of an argument may decisively influence how the system places its boundaries, and conversely, the shape of an argument is an important factor in predicting its category.

We treat the task as a word-by-word tagging problem, using a variant of the `IOB2` labeling scheme.

2 Transformation-based learning

TBL is a general machine learning tool for assigning classes to a sequence of observations. TBL induces a set of transformational rules, which apply in sequence to change the class assigned to observations which meet the rules' conditions.

We use the software package `fnTBL` to design the model described here. This package, and the TBL framework itself, are described in detail by Ngai and Florian (2001).

3 Task Definition

Defining the task of semantic role labeling in TBL terms requires four basic steps. First, the problem has to be reduced to that of assigning an appropriate tag to each word in a sentence. Second, we must define the features associated with each word in the sentence, on which the transformational rules will operate. Third, we must decide on the exact forms the transformational rules will be allowed to take (the *rule templates*). Finally, we must determine a mapping from our word-by-word tag assignment to the labeled bracketing used to identify semantic arguments in the test data. Each of these steps is addressed below.

3.1 Tagging scheme

The simplest way of representing the chunks of text which correspond to semantic arguments is to use some variant of the IOB tagging scheme (Sang and Veenstra, 1999). This is the approach taken by Hacioglu et al. (2003), who apply the IOB2 tagging scheme in their word-by-word models, as shown in the second row of Figure 1.

However, two aspects of the problem at hand make this tag assignment difficult to use for TBL. First, semantic argument chunks can be very large in size. An argument which contains a relative clause, for example, can easily be longer than 20 words. Second, the label an argument is assigned is largely arbitrary, in the sense that core argument labels (**A0**, **A1**, etc.) generally cannot be assigned without some information external to the constituent, such as the class of the predicate, or the identity of other arguments which have already been assigned. So using the IOB2 format, it might take a complicated se-

quence of TBL rules to completely re-tag, say, an **A0** argument as **A1**. If this re-tagging is imperfectly achieved, we are left with the difficult decision of how to interpret the stranded **I-A0** elements, and the problem that they may incorrectly serve as an environment for other transformational rules.

For this reason, we adopt a modified version of the IOB2 scheme which is a compromise between addressing the tasks of *argument identification* and *argument labeling*. The left boundary (B) tags indicate the label of the argument, but the internal (I) tags are non-specific as to argument label, as in the last row of Figure 1. This allows a a single TBL rule to re-label an argument, while still allowing for interleaving of TBL rules which affect argument identification and labeling.

3.2 Feature Coding

With each word in a sentence, we associate the following features:

Word The word itself, normalized to lower-case.

Tag The word's part-of-speech tag, as predicted by the system of Giménez and Màrquez (2003).

Chunk The chunk label of the word, as predicted by the system of Carreras and Màrquez (2003).

Entity The named-entity label of the word, as predicted by the system of Chieu and Ng (2003).

L/R A feature indicating whether the word is to the left (L) or right (R) of the target verb.

Indent This feature indicates the clause level of the current word with respect to the target predicate. Using the clause boundaries predicted by the system of Carreras and Màrquez (2003), we compute a feature based on the linguistic notion of *c-command*.[1] If both the predicate and the current word are in the same basic clause, **Indent**=0. If the predicate c-commands the current word, and the current word is one clause level lower, **Indent**=1. If it is two clause levels lower, **Indent**=2, and so on. If the c-command relations are reversed, the indent levels are negative, and if neither c-commands the other, **Indent**='NA'. (Figure 2 illustrates how this feature is defined.) The absolute value of the **Indent** feature is not permitted to exceed 5.

is-PP A boolean feature indicating whether the word is included within a base prepositional phrase. This is true if its chunk tag is **B-PP** or **I-PP**, or if it is within an NP chunk directly following a PP chunk.

PP-head If **is-PP** is true, this is the head of the prepositional phrase; otherwise it is zero.

N-head The final nominal element of the next NP chunk at the same indent level as the current word, if it exists. For purposes of this feature, a possessive NP chunk is combined with the following NP chunk.

Verb The target predicate under consideration.

V-Tag The POS tag of the target predicate.

V-Passive A boolean feature indicating whether the target verb is in the passive voice. This is determined using a simple regular expression over the sentence.

Path As in (Pradhan et al., 2003), this feature is an ordered list of the chunk types intervening between the target verb and the current word, with consecutive NP chunks treated as one.

3.3 Rule Templates

In order to define the space of rules searched by the TBL algorithm, we must specify a set of rule templates, which determine the form transformational rules may take. The rule templates used in our system are 130 in number, and fall into a small number of classes, as described below.

These rules all take the form $f^1 \ldots f^n \rightarrow \text{label}_w$, where f^1 through f^n are features of the current word w or words in its environment, and usually include the current (semantic argument) label assigned to w. The categorization of rule templates below, then, basically amounts to a list of the different feature sets which are used to predict the argument label of each word.

The initial assignment of tags which is given to the TBL algorithm is a very simple chunk-based assignment. Every word is given the tag **O** (outside all semantic arguments), except if it is within an NP chunk at **Indent** level zero. In that case, the word is assigned the tag **I** if its chunk label is **I-NP**, **B-A0** if its chunk label is **B-NP** and it is to the left of the verb, and **B-A1** if its chunk label is **B-NP** and it is to the right of the verb.

3.3.1 Basic rules (10 total)

The simplest class of rules simply change the current word's argument label based on its own local features, including the current label, and the features **L/R**, **Indent**, and **Chunk**.

3.3.2 Basic rules using local context (29)

An expanded set of rules using all features of the current word, as well as the argument labels of the current and previous words. For example, the following rule will change the label **O** to **I** within an NP chunk, if the initial

[1] A node α (reflexively) c-commands a node β iff there is a node γ such that γ directly dominates α, and γ dominates β. Note that only clauses (S nodes) are considered in our application described above.

Argument boundaries	[A1 The deal]	[V collapsed]	[AM-TMP on Friday]	.
IOB2	[B-A1 The] [I-A1 deal]	[B-V collapsed]	[B-AM-TMP on] [I-AM-TMP Friday]	[O .]
Modified scheme	[B-A1 The] [I deal]	[B-V collapsed]	[B-AM-TMP on] [I Friday]	[O .]

Figure 1: Tag assignments for word-by-word semantic role assignment

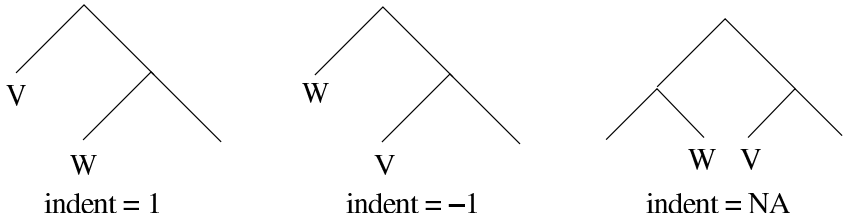

Figure 2: Sample values of **Indent** feature for different clause embeddings of a word W and target verb V

portion of the chunk has already been marked as within a semantic argument:

$$\begin{array}{l} \text{label}_{w_0} = \textbf{O} \\ \text{indent}_{w_0} = \textbf{0} \\ \text{chunk}_{w_0} = \textbf{I-NP} \\ \text{L/R}_{w_0} = \textbf{R} \\ \text{label}_{w_{-1}} = \textbf{I} \end{array} \rightarrow \text{label}_{w_0} = \textbf{I}.$$

3.3.3 Lexically conditioned rules (14)

These rules change the argument label of the current word based on the **Word** feature of the current or surrounding words, in combination with argument labels and chunk labels from the surrounding context. For example, this rule marks the adverb *back* as a directional modifier when it follows the target verb:

$$\begin{array}{l} \text{label}_{w_0} = \textbf{O} \\ \text{chunk}_{w_0} = \textbf{B-ADVP} \\ \text{word}_{w_0} = \textbf{back} \\ \text{label}_{w_{-1}} = \textbf{B-V} \\ \text{chunk}_{w_{-1}} = \textbf{B-VP} \end{array} \rightarrow \text{label}_{w_0} = \textbf{B-AM-DIR}.$$

3.3.4 Entity (24)

These rules further add the named-entity tag of the current, preceding, or following word to the basic and local-context rules above.

3.3.5 Verb tag (15)

These rules add the POS tag of the predicate to the basic and simpler local-context rules above.

3.3.6 Verb-Noun dependency (9)

These rules allow the argument label of the current word to be changed, based on its **Verb** and **N-head** features,as well as other local features.

3.3.7 Word-Noun dependency (3)

These rules allow the argument label of the current word to be changed, based on its **Word**, **N-head**, **Indent**, **L/R**, and **Chunk** features, as well as the argument labels of adjacent words.

3.3.8 Long-distance rules (6)

Because many of the dependencies involved in the semantic role labeling task hold over the domain of the entire sentence, we include a number of long-distance rules. These rules allow the argument label to be changed depending on the word's current label, the features **L/R**, **Indent**, **Verb**, and the argument label of a word within 50 or 100 words of the current word. These rules are intended to support generalizations like "if the current word is labeled **A0**, but there is already an **A0** further to the left, change it to **I**".

3.3.9 "Smoothing" rules (15)

Finally, there are a number of "smoothing" rules, which are designed primarily to prevent **I** tags from becoming stranded, so that arguments which contain a large number of words can successfully be identified. These rules allow the argument label of a word to be changed based on the argument labels of the previous two words, the next two words, and the chunk tags of these words. This sample rule marks a word as being argument-internal, if both its neighbors are already so marked:

$$\begin{array}{l} \text{label}_{w_{-1}} = \textbf{I} \\ \text{label}_{w_0} = \textbf{O} \\ \text{label}_{w_1} = \textbf{I} \end{array} \rightarrow \text{label}_{w_0} = \textbf{I}.$$

3.3.10 Path rules (5)

Finally, we include a number of rule templates using the highly-specific **Path** feature. These rules allow the argument label of a word to be changed based on its current value, as well as the value of the feature **Path** in combination with **L/R**, **Indent**, **V-Tag**, **Verb**, and **Word**.

3.4 Tag interpretation

The final step in our transformation-based approach to semantic role labeling is to map the word-by word IOB tags predicted by the TBL model back to the format of the original data set, which marks only argument boundaries, so that we can calculate precision and recall statistics for

each argument type. The simplest method of performing this mapping is to consider an argument as consisting of an initial labeled boundary tag (such as **B-A0**, followed by zero or more argument-internal (**I**) tags, ignoring anything which does not conform to this structure (in particular, strings of **I**s with no initial boundary marker).

In fact, this method works quite well, and it is used for the results reported below.

Finally, there is a post-processing step in which adjuncts may be re-labeled if the same sequence of words is found as an adjunct in the training data, and always bears the same role. This affected fewer than twenty labels on the development data, and added only about 0.1 to the overall f-measure.

4 Results

The results on the test section of the CoNLL 2004 data are presented in Table 1 below. The overall result, an f-score of 60.66, is considerably below results reported for systems using a parser on a comparable data set. However, it is a reasonable result given the simplicity of our system, which does not make use of the additional information found in the PropBank frames themselves.

It is an interesting question to what extent our results depend on the use of the **Path** feature (which Pradhan et al. (2003) found to be essential to their models' performance). Since this **Path** feature is also likely to be one of the model's most brittle features, depending heavily on the accuracy of the syntactic analysis, we might hope that the system does not depend too heavily on it. In fact, the overall f-score on the development set drops from 62.75 to 61.33 when the **Path** feature is removed, suggestig that it is not essential to our model, though it does help performance to some extent.

	Precision	Recall	$F_{\beta=1}$
Overall	64.17%	57.52%	60.66
A0	72.48%	68.94%	70.67
A1	63.57%	61.88%	62.72
A2	51.32%	40.90%	45.52
A3	51.58%	32.67%	40.00
A4	36.07%	44.00%	39.64
A5	0.00%	0.00%	0.00
AM-ADV	41.08%	32.25%	36.13
AM-CAU	63.33%	38.78%	48.10
AM-DIR	31.58%	24.00%	27.27
AM-DIS	56.93%	53.99%	55.42
AM-EXT	70.00%	50.00%	58.33
AM-LOC	26.34%	21.49%	23.67
AM-MNR	46.90%	26.67%	34.00
AM-MOD	96.24%	91.10%	93.60
AM-NEG	90.98%	95.28%	93.08
AM-PNC	37.93%	12.94%	19.30
AM-PRD	0.00%	0.00%	0.00
AM-TMP	51.81%	38.42%	44.12
R-A0	82.00%	77.36%	79.61
R-A1	78.26%	51.43%	62.07
R-A2	100.00%	22.22%	36.36
R-A3	0.00%	0.00%	0.00
R-AM-LOC	50.00%	25.00%	33.33
R-AM-MNR	0.00%	0.00%	0.00
R-AM-PNC	0.00%	0.00%	0.00
R-AM-TMP	100.00%	7.14%	13.33
V	98.15%	98.15%	98.15

Table 1: Results on test set: closed challenge

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Xavier Carreras and Lluís Màrquez. 2003. Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of RANLP 2003*.

Hai Leong Chieu and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL 2003*.

Jesús Giménez and Lluís Màrquez. 2003. Fast and accurate part-of-speech tagging: the SVM approach revisited. In *Proceedings of RANLP 2003*.

Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin, and Dan Jurafsky. 2003. Shallow semantic parsing using support vector machines. Technical Report CSLR-2003-01, Center for Spoken Language Research, University of Colorado at Boulder.

Grace Ngai and Radu Florian. 2001. Transformation-based learning in the fast lane. In *Proceedings of NAACL 2001*, pages 40–47, June.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2003. Support vector learning for semantic argument classification. Technical Report CSLR-2003-03, Center for Spoken Language Research, University of Colorado at Boulder.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of EACL 1999*, pages 173–179.