Semantic Role Labeing System using Maximum Entropy Classifier *

Ting Liu, Wanxiang Che, Sheng Li, Yuxuan Hu and Huaijun Liu
Information Retrieval Lab
School of Computer Science and Technology
Harbin Institute of Technology
China, 150001
{tliu, car, ls, yxhu, hjliu}@ir.hit.edu.cn

Abstract

A maximum entropy classifier is used in our semantic role labeling system, which takes syntactic constituents as the labeling units. The maximum entropy classifier is trained to identify and classify the predicates' semantic arguments together. Only the constituents with the largest probability among embedding ones are kept. After predicting all arguments which have matching constituents in full parsing trees, a simple rule-based post-processing is applied to correct the arguments which have no matching constituents in these trees. Some useful features and their combinations are evaluated.

1 Introduction

The semantic role labeling (SRL) is to assign syntactic constituents with semantic roles (arguments) of predicates (most frequently verbs) in sentences. A semantic role is the relationship that a syntactic constituent has with a predicate. Typical semantic arguments include Agent, Patient, Instrument, etc. and also adjunctive arguments indicating Locative, Temporal, Manner, Cause, etc. It can be used in lots of natural language processing application systems in which some kind of semantic interpretation is needed, such as question and answering, information extraction, machine translation, paraphrasing, and so on.

*This research was supported by National Natural Science Foundation of China via grant 60435020

Last year, CoNLL-2004 hold a semantic role labeling shared task (Carreras and Màrquez, 2004) to test the participant systems' performance based on shallow syntactic parser results. In 2005, SRL shared task is continued (Carreras and Màrquez, 2005), because it is a complex task and now it is far from desired performance.

In our SRL system, we select maximum entropy (Berger et al., 1996) as a classifier to implement the semantic role labeling system. Different from the best classifier reported in literatures (Pradhan et al., 2005) – support vector machines (SVMs) (Vapnik, 1995), it is much easier for maximum entropy classifier to handle the multi-class classification problem without additional post-processing steps. The classifier is much faster than training SVMs classifiers. In addition, maximum entropy classifier can be tuned to minimize over-fitting by adjusting gaussian prior. Xue and Palmer (2004; 2005) and Kwon et al. (2004) have applied the maximum entropy classifier to semantic role labeling task successfully.

In the following sections, we will describe our system and report our results on development and test sets.

2 System Description

2.1 Constituent-by-Constituent

We use syntactic constituent as the unit of labeling. However, it is impossible for each argument to find its matching constituent in all auto parsing trees. According to statistics, about 10% arguments have no matching constituents in the training set of 245,353

Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL),
pages 189–192, Ann Arbor, June 2005.

constituents. The top five arguments with no matching constituents are shown in Table 1. Here, Charniak parser got 10.08% no matching arguments and Collins parser got 11.89%.

Table 1: The top five arguments with no matching constituents.

Args	Cha parser	Col parser	Both
AM-MOD	9179	9205	9153
A1	5496	7273	3822
AM-NEG	3200	3217	3185
AM-DIS	1451	1482	1404
A0	1416	2811	925

Therefore, we can see that Charniak parser got a better result than Collins parser in the task of SRL. So we use the full analysis results created by Charniak parser as our classifier's inputs. Assume that we could label all AM-MOD and AM-NEG arguments correctly with simple post processing rules, the upper bound of performance could achieve about 95% recall.

At the same time, we can see that for some arguments, both parsers got lots of no matchings such as AM-MOD, AM-NEG, and so on. After analyzing the training data, we can recognize that the performance of these arguments can improve a lot after using some simple post processing rules only, however other arguments' no matching are caused primarily by parsing errors. The comparison between using and not using post processing rules is shown in Section 3.2.

Because of the high speed and no affection in the number of classes with efficiency of maximum entropy classifier, we just use one stage to label all arguments of predicates. It means that the "NULL" tag of constituents is regarded as a class like "ArgN" and "ArgM".

2.2 Features

The following features, which we refer to as the basic features modified lightly from Pradhan et al. (2005), are provided in the shared task data for each constituent.

- **Predicate lemma**
- **Path:** The syntactic path through the parse tree from the parse constituent to the predicate.
- **Phrase type**
- **Position:** The position of the constituent with respect to its predicate. It has two values, "before" and "after", for the predicate. For the situation of "cover", we use a heuristic rule to ignore all of them because there is no chance for them to become an argument of the predicate.
- **Voice:** Whether the predicate is realized as an active or passive construction. We use a simple rule to recognize passive voiced predicates which are labeled with part of speech – VBN and sequences with AUX.
- **Head word stem:** The stemming result of the constituent's syntactic head. A rule based stemming algorithm (Porter, 1980) is used. Collins Ph.D thesis (Collins, 1999)[Appendix. A] describs some rules to identify the head word of a constituent. Especially for prepositional phrase (PP) constituent, the normal head words are not very discriminative. So we use the last noun in the PP replacing the traditional head word.
- **Sub-categorization**

We also use the following additional features.

- **Predicate POS**
- **Predicate suffix:** The suffix of the predicate. Here, we use the last 3 characters as the feature.
- **Named entity:** The named entity's type in the constituent if it ends with a named entity. There are four types: LOC, ORG, PER and MISC.
- **Path length:** The length of the path between a constituent and its predicate.
- **Partial path:** The part of the path from the constituent to the lowest common ancestor of the predicate and the constituent.
- **Clause layer:** The number of clauses on the path between a constituent and its predicate.
- **Head word POS**
- **Last word stem:** The stemming result of the last word of the constituent.
- **Last word POS**

We also use some combinations of the above features to build some combinational features. Lots of combinational features which were supposed to contribute the SRL task of added one by one. At the same time, we removed ones which made the performance decrease in practical experiments. At last, we keep the following combinations:

- Position + Voice
- Path length + Clause layer
- Predicate + Path
- Path + Position + Voice
- Path + Position + Voice + Predicate
- Head word stem + Predicate
- Head word stem + Predicate + Path
- Head word stem + Phrase
- Clause layer + Position + Predicate

All of the features and their combinations are used without feature filtering strategy.

2.3 Classifier

Le Zhang's Maximum Entropy Modeling Toolkit [1], and the L-BFGS parameter estimation algorithm with gaussian prior smoothing (Chen and Rosenfeld, 1999) are used as the maximum entropy classifier. We set gaussian prior to be 2 and use 1,000 iterations in the toolkit to get an optimal result through some comparative experiments.

2.4 No Embedding

The system described above might label two constituents even if one embeds in another, which is not allowed by the SRL rule. So we keep only one argument when more arguments embedding happens. Because it is easy for maximum entropy classifier to output each prediction's probability, we can label the constituent which has the largest probability among the embedding ones.

2.5 Post Processing Stage

After labeling the arguments which are matched with constituents exactly, we have to handle the arguments, such as `AM-MOD`, `AM-NEG` and `AM-DIS`, which have few matching with the constituents described in Section 2.1. So a post processing is given by using some simply rules:

- Tag target verb and successive particles as V.
- Tag "not" and "n't" in target verb chunk as `AM-NEG`.
- Tag modal verbs in target verb chunk, such as words with POS of "MD", "going to", and so on, as `AM-MOD`.
- Tag the words with POS of "CC" and "RB" at the start of a clause which include the target verb as `AM-DIS`.

3 Experiments

3.1 Data and Evaluation Metrics

The data provided for the shared task is a part of PropBank corpus. It consists of the sections from the Wall Street Journal part of Penn Treebank. Sections 02-21 are training sets, and Section 24 is development set. The results are evaluated for precision, recall and $F_{\beta=1}$ numbers using the *srl-eval.pl* script provided by the shared task organizers.

3.2 Post Processing

After using post processing rules, the final $F_{\beta=1}$ is improved from 71.02% to 75.27%.

[1] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

3.3 Performance Curve

Because the training corpus is substantially enlarged, this allows us to test the scalability of learning-based SRL systems to large data set and compute learning curves to see how many data are necessary to train. We divide the training set, 20 sections Penn Treebank into 5 parts with 4 sections in each part. There are about 8,000 sentences in each part. Figure 1 shows the change of performance as a function of training set size. When all of training data are used, we get the best system performance as described in Section 3.4.

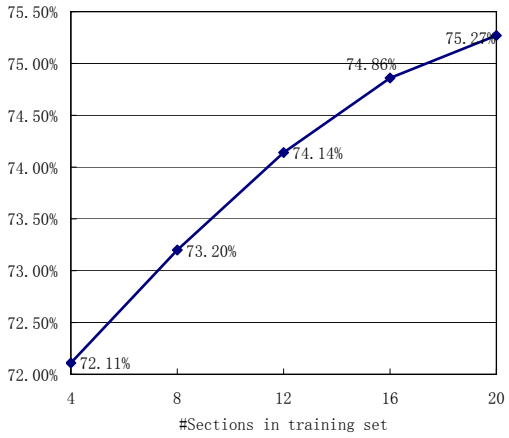

Figure 1: Our SRL system performance curve (of $F_{\beta=1}$) effecting of the training set size.

We can see that as the training set becomes larger and larger, so does the performance of SRL system. However, the rate of increase slackens. So we can say that at present state, the larger training data has favorable effect on the improvement of SRL system performance.

3.4 Best System Results

In all the experiments, all of the features and their combinations described above are used in our system. Table 2 presents our best system performance on the development and test sets.

From the test results, we can see that our system gets much worse performance on Brown corpus than WSJ corpus. The reason is easy to be understood for the dropping of automatic syntactic parser performance on new corpus but WSJ corpus.

The training time on PIV 2.4G CPU and 1G Mem machine is about 20 hours on all 20 sections, 39,832-

	Precision	Recall	$F_{\beta=1}$
Development	79.65%	71.34%	75.27
Test WSJ	80.48%	72.79%	76.44
Test Brown	71.13%	59.99%	65.09
Test WSJ+Brown	79.30%	71.08%	74.97

Test WSJ	Precision	Recall	$F_{\beta=1}$
Overall	80.48%	72.79%	76.44
A0	88.14%	83.61%	85.81
A1	79.62%	72.88%	76.10
A2	73.67%	65.05%	69.09
A3	76.03%	53.18%	62.59
A4	78.02%	69.61%	73.58
A5	100.00%	40.00%	57.14
AM-ADV	59.85%	48.02%	53.29
AM-CAU	68.18%	41.10%	51.28
AM-DIR	56.60%	35.29%	43.48
AM-DIS	76.32%	72.50%	74.36
AM-EXT	83.33%	46.88%	60.00
AM-LOC	65.31%	52.89%	58.45
AM-MNR	58.28%	51.16%	54.49
AM-MOD	98.52%	96.37%	97.43
AM-NEG	97.79%	96.09%	96.93
AM-PNC	43.68%	33.04%	37.62
AM-PRD	50.00%	20.00%	28.57
AM-REC	0.00%	0.00%	0.00
AM-TMP	78.38%	66.70%	72.07
R-A0	81.70%	85.71%	83.66
R-A1	77.62%	71.15%	74.25
R-A2	60.00%	37.50%	46.15
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	100.00%	25.00%	40.00
R-AM-EXT	0.00%	0.00%	0.00
R-AM-LOC	83.33%	47.62%	60.61
R-AM-MNR	66.67%	33.33%	44.44
R-AM-TMP	77.27%	65.38%	70.83
V	98.71%	98.71%	98.71

Table 2: Overall results (top) and detailed results on the WSJ test (bottom).

sentences training set with 1,000 iterations and more than 1.5 million samples and 2 million features. The predicting time is about 160 seconds on 1,346-sentences development set.

4 Conclusions

We have described a maximum entropy classifier is our semantic role labeling system, which takes syntactic constituents as the labeling units. The fast training speed of the maximum entropy classifier allows us just use one stage of arguments identification and classification to build the system. Some useful features and their combinations are evaluated. Only the constituents with the largest probability among embedding ones are kept. After predicting all arguments which have matching constituents in full parsing trees, a simple rule-based post-processing is applied to correct the arguments which have no matching constituents. The constituent-based method depends much on the syntactic parsing performance. The comparison between WSJ and Brown test sets results fully demonstrates the point of view.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL-2004*, pages 89–97, Boston, MA, USA.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Pennsylvania University.

Namhee Kwon, Michael Fleischman, and Eduard Hovy. 2004. Framenet-based semantic parsing using maximum entropy models. In *Proc. Coling 2004*.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3).

Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *Machine Learning Journal*.

Vladamir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proc. EMNLP 2004*.

Nianwen Xue and Martha Palmer. 2005. Automatic semantic role labeling for chinese verbs. In *Proc. IJCAI 2005*.