Hybrid learning of Syntactic and Semantic Dependencies

Lin Yao
Computer Science Department, HIT Campus of ShenZhen Univeristy Town
ShenZhen, 518055, China
Tel: 86-0755-2603-3789 E-mail: yaolin@hit.edu.cn

Chengjie Sun, Lu Li, Zhixin Hao & Xiaolong Wang
Computer Science Department, HIT Campus of ShenZhen Univeristy Town
ShenZhen, 518055, China
Tel: 86-0755-2603-3789 E-mail: {cjsun, lli, xlwang}@insun.hit.edu.cn, haozhix@cs.hitsz.edu.cn

Abstract

This paper presents our solution for jointly parsing of syntactic and semantic dependencies. The Maximum Entropy (ME) classifier is selected in this system. Also the Mutual Information (MI) model was utilized into feature selection of dependency labeling. Results show that the MI model allows the system to get better performance and reduce training hours.

Keywords: Maximum Entropy, Mutual Information, Semantic Role Labeling (SRL), Syntactic Parser

1. Introduction

Since 2002, semantic role labeling which focus on recognizing and labeling semantic arguments has received considerable interests because of its big contribution to many Natural Language Processing (NLP) applications, such as information extraction, question answering, machine translation, paraphrasing, and etc.

In the past four years, the Conference on Computational Natural Language Learning (CoNLL) featured an associated share task every year which allow the participants to train and test their Semantic Role Labeling (SRL) or Syntactic systems on the same date sets and share their experiences. In 2004 and 2005, the shared tasks of CoNLL were focus on SRL. In CoNLL-2006 and CoNLL-2007, the shared tasks were dedicated to the syntactic dependency parsing.

In 2008, the CoNLL consolidates the last four years endeavor and unites the tasks into one, and proposes a new challenge of the merging of both syntactic dependencies (extracted from the Penn Treebank) and semantic dependencies (extracted both from PropBank) under an unique unified representation.

After more than 7 years endeavor on semantic and syntactic areas, many different learning techniques have been practiced on this area, such as ME, Support Vector Machine (SVM), Relevance Vector Machine(RVM),SNoW, Tree-Conditional Random Field(T-CRF), DT, CPM.

Several variants of the basic approaches using different features and different classifiers have been introduced. (Carreras, 2007; Fleischman & Hovy, 2003; Gildea, Hockenmaier, & PHILADELPHIA, 2003; Gildea & Jurafsky, 2002; Pradhan, Ward, Hacioglu, Martin, & Jurafsky, 2004; Surdeanu, Harabagiu, Williams, & Aarseth, 2003; Surdeanu, Màrquez, Carreras, & Comas, 2007; Thompson, Levy, & Manning, 2003; Xue & Palmer, 2004)

As the most popular Classifier used in semantic role labeling currently, Maximum Entropy model(Berger, Della Pietra, & Della Pietra, 1996) is proper for Multiclass Categorization task. It could use relative shorter training time to get more accurate probability for each output category. Not just that, the maximum entropy framework pales others also by integrating facts without features independence hypothesis. Only in 2005 share task, up to 8 teams used the Maximum Entropy (ME) statistical framework (Che, Liu, Li, Hu, & Liu, 2005; Haghighi, Toutanova, & Manning, 2005; Park & Rim, 2005; Sang, Canisius, van den Bosch, & Bogers, 2005; Sutton & McCallum, 2005; Tsai, Wu, Lin, & Hsu, 2005; Venkatapathy, Bharati, & Reddy, 2005; Yi & Palmer, 2005). For these reasons,we propose a solution that selects Maximum Entropy model. (Liu, Che, Li, Hu, & Liu, 2005) and (Li, Fan, & Wang, 2008) have applied the maximum entropy classier in semantic task successfully. Therefore plentiful feature templates will rocket the training time and bring features mutual interference which induces the decline of system performance. Consequently different from prior systems, mutual information method is selected to decrease the number of features and reduce the train time efficiently with little impact on result.

2. Problem Definition

SRL concerned the recognition of semantic roles in a given sentence. As defined in the Shared Tasks of CoNLL-2004, the task includes analyzing the propositions expressed by some target verbs of the sentence. Based on PropBank predicate-argument structures, for each target verb, all the constituents in the sentence which fill a semantic role of the verb have to be recognized(Carreras, Màrquez, & Processing, 2005).

In each sentence a word (most time is a verb) will be specified as a predicate and its accompanying arguments will be recognized and labeled with different categories depending on the roles they played.

For example:

[$_{A0}$ He] [$_{AM\text{-}MOD}$ would] [$_{AM\text{-}NEG}$ not] [$_{V}$ **receive**] [$_{A1}$ anything valuable] from [$_{A2}$ who he known].

Here, the roles for the predicate **receive** (that is, the *roleset* of the predicate) are defined in the PropBank Frames scheme as:

V: verb
A0: receiver
A1: thing gotten
A2: receive-from
AM-MOD: modal
AM-NEG: negation

The addressing of SRL problem task takes for quite a long time. Since CoNLL-2004 shared task, the SRL problem is come up with machine learning algorithm and based on partial syntactic information without using full parsers and external lexico-semantic knowledge bases(Blunsom, 2004). In CoNLL-2005, following the earlier year's initiative, the task still interest in the recognition of semantic roles for the English language. Novelty of this shared task is that the complete syntactic trees were provided as input to improve the performance in SRL. And the training corpus has been substantially enlarged. In CoNLL-2006 and CoNLL-2007, the shared tasks were dedicated to the syntactic dependency parsing. In 2008, the CoNLL combined past four years endeavor and merged the tasks into one and proposed a new challenge of the merging of both syntactic dependencies (extracted from the Penn Treebank) and semantic dependencies (extracted both from PropBank) under a unique unified representation. Compared with the earlier CoNLL share task, this time SRL using a dependency-based representation and the predicated is not limited to verbal and nominal. Researchers try to prove the dependency-based representation better for SRL than the constituent-based formalism and the merged representation more helpful than the individual ones. After merge, this combination is easier adoptable for most NLP technology and linear time processing is better fit for many applications.

3. Maximum Entropy Model

In Shannon's Information Theory, entropy is defined as a measure of the uncertainty associated with a random variable. Maximum Entropy model allows people to make an optimal choice among various states which all satisfy prior constraints by choosing the state with the maximum entropy, in other words, all unknown possibilities have equally opportunity.

Maximum entropy allows us to restrict the model distribution to have the constraints which are represented as the same expected value for feature as seen in the training data(Nigam, Lafferty, & McCallum, 1999). The motivating idea behind maximum entropy is to derive the uniform model with maximum H(p) value and shown in Eq. (3.1).

$$p = \arg\max_{p\in C}\ \mathrm{H}(p) \tag{3.1}$$

The maximum entropy framework makes different features be integrated into the same model without concerning the relationships among them. These advantages make the ME model has been used in many NLP areas. ME model is defined in E.q (3.2).

$$\begin{aligned} p_\lambda(y\mid \mathrm{X}) &= \frac{1}{Z(\mathrm{X})}\exp(\sum_{\mathrm{i}}\lambda_{\mathrm{i}} f_{\mathrm{i}}(\mathrm{X},\mathrm{y})) \\ \mathrm{Z}(\mathrm{X}) &= \sum_{\mathrm{y}}\exp(\sum_{\mathrm{i}}\lambda_{\mathrm{i}} f_{\mathrm{i}}(\mathrm{X},\mathrm{y})) \end{aligned} \tag{3.2}$$

Where y is a class label (in our case: meaningful or unmeaning), X is an input vector containing predicates on the matched text, and Z(X) is a normalization term. Each feature function f_i(X, y) maps an input vector and class to a binary value, for example:

$$f_i(\mathrm{X}, y) = \begin{cases} 1, \textit{ if currentword= SYT in X and y=meaningful} \\ 0, \textit{ otherwise} \end{cases} \tag{3.3}$$

The parameters of the model are the feature weights λ_i. They are determined in such a way that the parameters maximize the conditional log-likelihood of the training data $\sum_{k=1}^{N} \log p_\lambda(y^{(k)} \mid \mathrm{X}^{(k)})$, where N is the number of training samples. We use Zhangle's ME tool package(Zhang, 2004) to train the ME model.

For the semantic role labeling, the main task will be picking up the right label from the semantic label sets. $P(y/X)$ denotes the conditional probability of getting the output y with the context X. f_i(X, y) describes the feature constrain with weighting parameter λ_i.

The ME model will choose the value of $P(y/X)$ which makes H(p) the gets the maximum value.

4. Corpus and Evaluation

Large corpora that allow to automatically extracting information about language are beginning to serve researchers in NLP as critical tools. Since there is no large-scale dependency Treebank published, the corpus we adopted is provided by the CoNLL shared task. It is generated from several well-known corpora. These corpora will be briefly introduced in the follows: The Penn Treebank(Marcus, Marcinkiewicz, & Santorini, 1993), Proposition Bank (PropBank)(Palmer, Kingsbury, & Gildea, 2005), NomBank (Meyers, et al., 2004).The prior corpora have been merged and converted into dependency formalism which were used in shared task evaluation(Surdeanu, Johansson, Meyers, Màrquez, & Nivre, 2008). The data format defined (Surdeanu, et al., 2008) as 12 columns. They are ID, FORM, LEMMA, GPOS, PPOS, SPLIT_FORM, SPLIT_LEMMA, PPOSS, HEAD, DEPREL, PRED, ARG. The HEAD means syntactic head of the current token and DEPREL means syntactic dependency relation to the HEAD.

The official evaluation of 2008 shared task is concerned from three aspects(Surdeanu, et al., 2008):

syntactic dependencies

semantic dependencies

the overall task

The labeled attachment score (LAS) is used for evaluating the syntactic dependencies and defined as the percentage of tokens which have correct HEAD and DEPRELvalues. Labeled F1 score is applied for the evaluation of semantic dependencies. The macro average of the two previous scores is used to score the whole task(Surdeanu, et al., 2008).

5. System Architecture:

Comparing with traditional Semantic Role Label, our system will produce a joint rich syntactic-semantic output to allow people getting semantic role annotation and syntactic structure at the same time. The main components of our system:

Syntactic parsing.

Predicate tagging.

Features Selection using Mutual Information.

Semantic dependency labeling.

During syntactic parser stage, syntactic dependency tree are achieved. In the predicated tagging module, the predicates in the sentence will be marked with tag numbers, 00 is for non-predicate. The results generated by earlier two stages are served as input of the last component. With the filtered features, the Semantic dependency labeling generates the classification of semantic roles.

5.1 Syntactic parsing

In our system, the first stage is to create a labeled syntactic dependency parse y for input sentence x including words and their part of speeches (POS). Inspired by the parsing model presented in (McDonald, Pereira, Ribarov, & Hajic, 2005), we equates the problem of dependency parsing to finding maximum spanning trees in directed graphs. As Figure 1, the following flow chart presents the sequence of syntactic parsing.

Our previous work (Li et. al. 2008) detailed the features used for the syntactic parsing.

5.2 Predicate Tagging

Usually an event in a sentence can be described using predicate and its arguments. The predicate, which most time is a verb, reveals the type of event. Arguments are the sentence constituents holding semantic roles and combine with the predicate to complete the meaning of the sentence event. Accordingly the semantic task includes two subtasks:

Predicate tagging.

Semantic dependency labeling

Predicated tagging task is not only binary classification problem, we treated it as multiple classification question since we include role sets as predicate type which corresponding to different meanings. The framework for Predication Tagging is shown as Figure 2.

In both PropBank(Palmer, et al., 2005) and NomBank (Meyers, et al., 2004), predicates usually have assigned several rolesets corresponding to different meanings. For example, the verb abandon has three rolesets marked as ordinal numbers 01, 02 and 03 as described below.

```
<frameset>
<predicate lemma="abandon">
<roleset id="abandon.01" name="leave
behind" vncls="51.2">
. . .
</roleset>
<roleset id="abandon.02"
name="exchange" vncls="51.2">
. . .
</roleset>
<roleset id="abandon.03"
name="surrender, give over" vncls="-
">
. . .
</roleset>
</predicate>
</frameset>
```

Choosing different rolesets means different meanings, which will affect estimation of event. Consequently, these numbers of role set are treated as tags for predicates; some statistical properties will be obtained.

The tag set was chosen as {01, 02... 22} which corresponds to numbers of role set. And a special condition shall be considered that when the word is not predicate. 00 will be added in to indicate non-predicate.

The predicate tagging feature template includes 3 categories features: content features, context features and compound features (shown in Table 1).

The content features give the information about current word. The POS framework of current word was presented as neighbor's POS, called context features. With study of the corpus, we found out compound words such as short-term affect the predicate tagging performance distinctly. The feature template for compound words will included several separate features which will be assigned null for non-compound words.

5.2.1 Semantic Dependency labeling

Shown in Figure 3, this part of system is based on the output of syntactic parser and predicate tagging mentioned earlier and to find out the dependents of a given predicate in a sentence and label them with the one from the set of semantic dependency labels.

Suppose (p, d) is a couple of predicate and one of its possible dependents, T is the dependency tree generated by syntactic parsing; L is the set of semantic dependency labels, null is included as a special tag for no semantic dependency between p and d. This task can also be recognized as classification issue and work out with ME model and be described as E.q (5.1).

$$l_{(\mathrm{p,d})} = \arg\max_{l \in L} \Pr\left(l \mid p, d, T\right) \tag{5.1}$$

For most language processing task, no matter which machine learning algorithm has been chosen, the selection of features set always was the crucial factors affect the performance of systems. We select features from three aspects: predicate, dependent, and properties between predicate and dependent.

To get better performance, a lot research work had been done on our train data and their properties. With the survey on the 39832 sentence and their 421039 semantic roles, the high-frequency PoS pairs between predicate and dependent and associated semantic roles have been survey, such as IN-NNS, RB-VBN,VBZ-VBN, RBS-VBN,CC-VB. Partial concurrence statistics results for POS pair and semantic roles were presented in the Figure 4.

Study the statistic result, the affiliation between POS pair and semantic role is obvious. And for the different semantic roles, the statistic property of the PoS pair is distinct. These characteristics show that the PoS pair is sustaining feature for semantic roles labeling.

At the same time, to discover the relevance between semantic dependency and syntactic dependency, huge amount word pairs have been studied. These words' family relationships on the syntactic tree also have been concerned. The family dependency relationship is defined as IS-A relationship between current word and predicate. The relationship set includes Son, Father, Brother, Self and Grandson relationship. The details are presented as Figure 5. Totally 421039 semantic dependency relationship were involved. 59.09% of them have directly syntactic dependency, which means the current word is the son node of predicate. If other family relationship were included into the consideration, this part occupies the 87.34% of total 421039 word pairs. These proves the syntactic dependency devote great contribution to the semantic recognition.

After plenty statistics and analysis, our system selects such as PoS pairs and family dependency relationships totally 24 features as the features template to build training model. The chosen features are listed in the Table 2.

5.2.2 Mutual Information

In dependency labeling, different features have different knowledge in distinguishing class labels. More features will contain more knowledge. Therefore the quantity of feature template is not proportion to the recognition ability of the model. Large numbers of features do not always signify high quality recognition. Too much information will bring interferes among themselves and decrease in performance. At the same time, a large number of feature templates mean huge time consuming and thereby limit the system practicability. How to reduce feature set and choose the optimal subset without compromising classification accuracy?

Mutual Information (MI) methods can be utilized as a pruning algorithm to choose a suitable subset from the earlier large candidate features. The mutual information represents the amount of uncertainty remaining about the system output Y that is resolved by observing the system input X. Mutual information (Al-Ani, 2003) between X and Y is given as Eq. (6.2).

$$I_{YX} = \sum_{x}\sum_{y} p\left(x, y\right) \log\left(\frac{p\left(y/x\right)}{p\left(y\right)}\right) \tag{6.2}$$

In our system, y is the semantic labels from label set; x is one of feature instances generated by a particular feature template. All the feature templates used in this system are listed in the Table 1. I_{YX} is the mutual information of a feature template with all the semantic labels. The mutual information of all feature templates will be calculated. The large value of mutual information between a feature template and its output means the closer associations. Usually a pruning value is set during the phrase of choosing feature templates; the feature template with the higher mutual information value will be kept. After calculation, the feature templates numbers in Table 1 have been reordered according its MI value in descending order. The first 18 feature templates are listed as follows:

(9, 6, 8, 5, 23, 11, 15, 4, 19, 18, 14, 21, 2, 16, 22, 24, 1, 10)

6. Experiments and Analysis

6.1 Predicate Tagging

The predicate identify model was tested on devel, wsj, brown and wsj+brown data separately. The experimental result is shown below in Table 3.

The performance on devel and wsj data is better than on brown data. That is because the training data is from the same corpus (Penn Treebank) as devel and wsj data, the brown data is from a different corpus (Brown). The results show that our system has limitations that the identification result might be worse on other test corpora.

6.2 Semantic Dependency labeling

Utilizing the MI model, the features are chosen according their contribution to the results. First all features are grouped, then we collect the first 18 features and those with less contribution are removed. The procedure is continued until the last group with 7 features. The test results with different feature combinations are included in the Figure 6. F24 means 24 features were selected, so as to analogize.

Figure 6 presents the best performance gotten with the 10 features combination. This result shows that more features did not give the better performance.

Figure 7 shows the 10 features selected recognition result which is correct. However, when 18 features (shown in Figure 8)were selected, the predicate "say" was not identified. This example demonstrates that excessive features go with redundant information which may lower performance, concise but more interrelated information may provide better performance.

Usage of the MI model decreases the template from 24 features to 10 and the training time is cut sharply, the performance with 10 features is better than others.

From the test results, we can also see that our system gets much better performance on WSJ corpus than Brown corpus. The reason is that the syntactic parser is constructed based on the WSJ corpus and we might get worse performance using other test corpora. The performances also imply that the error of syntactic parsing and predicate tagging could be probably augmented in semantic dependent labeling. In order to improve the performance of the whole system, the deep dependence between the two stages should be broken up in future research.

7. Conclusion

We present a semantic dependency system, which includes syntactic module, predicate tagging module and dependency labeling module. Through analyzing a large amount of corpus, we proved that there are the affiliations between POS pair and semantic role. And also we discover the relevance between semantic dependency and syntactic dependency. In the dependent role labeling subtask, the results show that the performance will not always be improved when we increase the number of features. MI model is applied to reduce the feature set; different feature combinations are evaluated, and get better performance and faster training speed.

References

Al-Ani, A. (2003). A new mutual information based measure for feature selection. *Intelligent Data Analysis,* 7(1), 43-57.

Berger, A., Della Pietra, V., & Della Pietra, S. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics,* 22(1), 39-71.

Blunsom, P. (2004). Maximum Entropy Markov Models for Semantic Role Labelling. Paper presented at the Proceedings of ALTA-2004.

Carreras, X. (2007). *Experiments with a higher-order projective dependency parser.*

Carreras, X., Màrquez, L., & Processing, N. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. Paper presented at the 9th Conference on Computational Natural Language Learning (CoNLL-2005).

Che, W., Liu, T., Li, S., Hu, Y., & Liu, H. (2005). Semantic role labeling system using maximum entropy classifier. Proceedings of Computational Natural Language Learning (CoNLL-2005).

Fleischman, M., & Hovy, E. (2003). *A maximum entropy approach to framenet tagging.*

Gildea, D., Hockenmaier, J., & PHILADELPHIA, P. U. (2003). Identifying Semantic Roles Using Combinatory Categorial Grammar: PENNSYLVANIA UNIV PHILADELPHIA.

Gildea, D., & Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics,* 28(3), 245-288.

Haghighi, A., Toutanova, K., & Manning, C. (2005). A joint model for semantic role labeling. Paper presented at the In Proceedings of CoNLL-2005.

Li, L., Fan, S., & Wang, X. (2008). Discriminative Learning of Syntactic and Semantic Dependencies. Paper presented at the Proceedings of Computational Natural Language Learning (CoNLL-2008)

Liu, T., Che, W., Li, S., Hu, Y., & Liu, H. (2005). *Semantic role labeling system using maximum entropy classifier.*

Marcus, M., Marcinkiewicz, M., & Santorini, B. (1993). Building a large annotated corpus of English: the penn treebank. *Computational Linguistics,* 19(2), 313-330.

McDonald, R., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., et al. (2004). *The NomBank Project: An Interim Report.*

Nigam, K., Lafferty, J., & McCallum, A. (1999). *Using maximum entropy for text classification.*

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

Park, K., & Rim, H. (2005). Maximum entropy based semantic role labeling. Paper presented at the In Proceedings of CoNLL-2005.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., & Jurafsky, D. (2004). Shallow semantic parsing using support vector machines.

Sang, E., Canisius, S., van den Bosch, A., & Bogers, T. (2005). Applying spelling error correction techniques for improving semantic role labelling. Paper presented at the In Proceedings of CoNLL-2005.

Surdeanu, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003). *Using predicate-argument structures for information extraction.*

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., & Nivre, J. (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. Proceedings of CoNLL–2008.

Surdeanu, M., Màrquez, L., Carreras, X., & Comas, P. (2007). Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research,* 29, 105-151.

Sutton, C., & McCallum, A. (2005). Joint parsing and semantic role labeling. Paper presented at the In Proceedings of CoNLL-2005.

Thompson, C., Levy, R., & Manning, C. (2003). A generative model for semantic role labeling. Lecture notes in computer science, 397-408.

Tsai, T., Wu, C., Lin, Y., & Hsu, W. (2005). Exploiting full parsing information to label semantic roles using an ensemble of me and svm via integer linear programming.

Venkatapathy, S., Bharati, A., & Reddy, P. (2005). Inferring semantic roles using subcategorization frames and maximum entropy model.

Xue, N., & Palmer, M. (2004). Calibrating features for semantic role labeling.

Yi, S., & Palmer, M. (2005). The integration of syntactic parsing and semantic role labeling.

Zhang, L. (2004). Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent.htm, 16

Table 1. Feature template for predicate tagging.

Catogories	Features	Description
Content Features	Postion	Position of the current word in sentence. It can be start word, middle word, and end word
	Punctuation	If the current word is a punctuation
	Letter	If the current word is an one-letter word
	Number	If the current word is a number
	Words_par	If the current word is a part of compound word
	Word and pos	The current word itself and its POS tag
	Prototype and POS	The prototype of current word and its POS tag
Context Features	POS of i, i+1	Pos of ±i means part-of-speech tag of the word n position after/before current word. i∈(-2,-1,0,1)
	POS of i-1, i, i+1	Pos of ±i means part-of-speech tag of the word n position after/before current word. i∈(1,0)
	POS of i-1,i, i+1, i+2	Pos of ±i means part-of-speech tag of the word n position after/before current word. i∈(0,-1)
Compound Features	Word and pos	Same as above
	Prototype and pos	Same as above
	Word_pos	POS tag of compound words it belongs to
	Pos 0,-2	Same as above
	Pos 0,+2	Same as above
	Pos -2,0,+2	Same as above

Table 2. Feature Template for semantic dependencies

No	Features	Description
1	Dependent Relationship	Dependent relationship between dependent anc predicate
2	Family Relation	Family relation between predicate and dependent ir the syntactic dependent tree: Son, Father, Brother o Self.
3	Position of Dependent	Position of dependent according predicate: before after or self.
4	POS pair	POS pair of predicate and dependent.
5	Dependency Path betweer Predicate and Dependent	Dependency label list of the Path from predicate tc dependent.
6	Predicate Information	Lemma and POS of predicate,
7	Predicate Voice	Active or passive.
8	Dependent Information	Lemma and POS of dependent
9	Path POS	POS list of all words appearing on the path fron predicate to dependent.
10	Dependency Path	Syntactic dependency label between predicate and it parent.
11	Predicate’s Parent Info.	Lemma and POS of predicate’s parent
12	POS List of Predicate’s Siblings	
13	Syntactic Dependency Framework of Predicate’s Parent	Syntactic dependency label list of the edges betweer predicate’s parent and its siblings
14	Dependent’s parent Info.	POS of dependent’s parent.
15	POS List of Dependent ’s Siblings	
16	Syntactic Dependency Framework of Dependent’s Parent	Syntactic dependency label list of the edges betweer dependent’s parent and its siblings.
17	Number of Dependent’s Children	
18	Dependent Category	VerbNet class number of Dependent
19	Path length	Path length between predicate and dependent.
20	POS List of Predicate’s Children	
21	POS of P-2	POS of the word 2 before Predict in the sentence
22	POS of P-1	POS of the word before predict in the sentence
23	POS of P+2	POS of the word 2 after predict in the sentence
24	POS of P+1	POS of the word after predict

Table 3. Predicate identification test result

Data	Precision	Recall	F-score
Devel	0.907848	0.914098	0.910962
Wsj	0.897038	0.914468	0.905669
Brown	0.844615	0.871429	0.857813
Wsj+Brown	0.89136	0.909856	0.900513

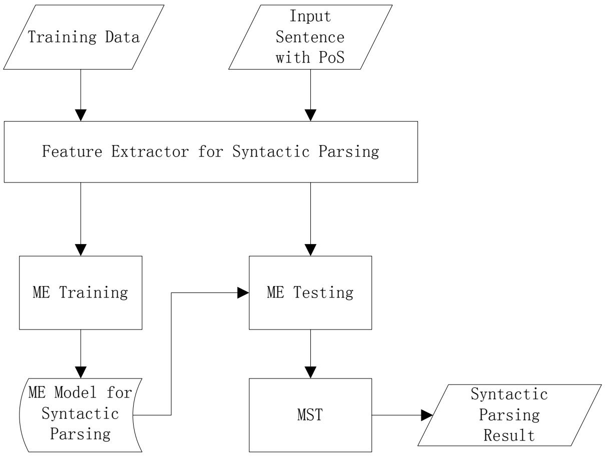


Figure 1. Framework for Syntactic Parser

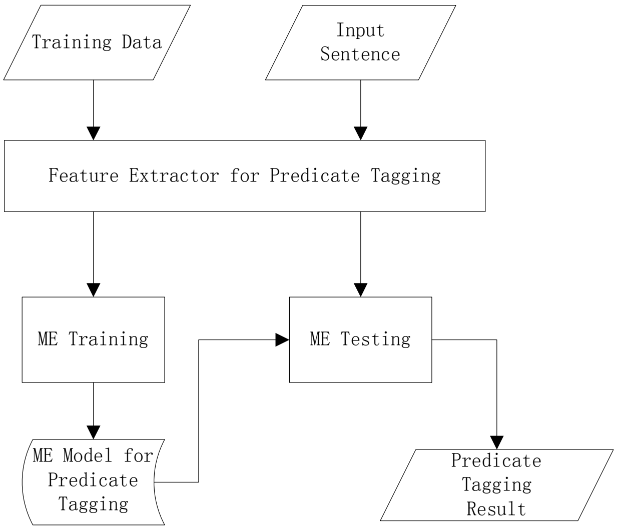


Figure 2. Framework for Predication Tagging

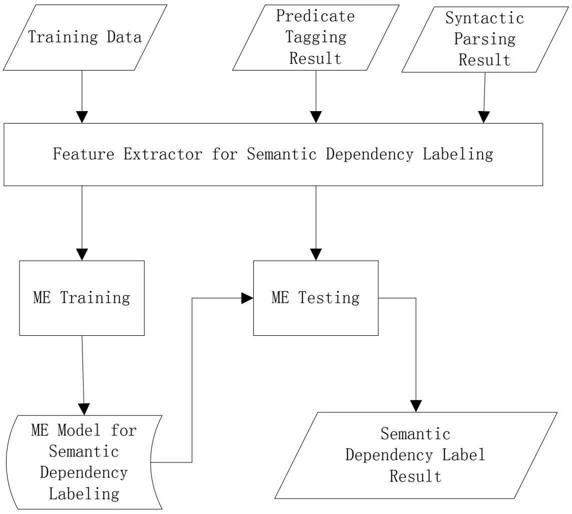
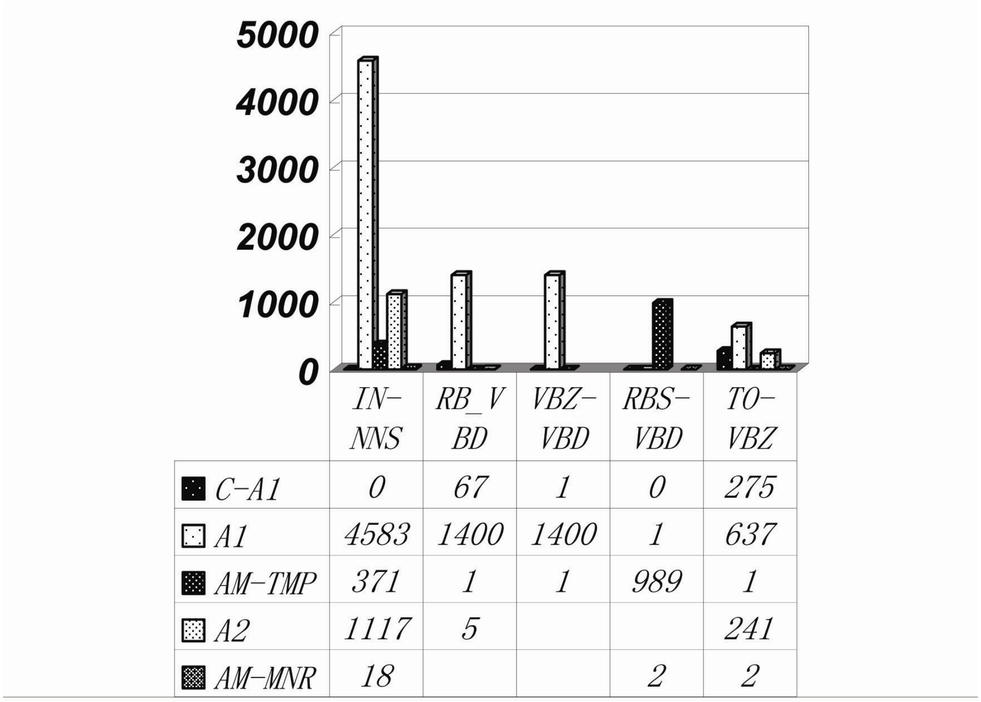

Training Data
Predicate Tagging Result
Syntactic Parsing Result
Feature Extractor for Semantic Dependency Labeling
ME Training
ME Testing
ME Model for Semantic Dependency Labeling
Semantic Dependency Label Result

Figure 3. Framework for Semantic Dependency Label



	IN-NNS	RB_VBD	VBZ-VBD	RBS-VBD	TO-VBZ
■ C-A1	0	67	1	0	275
□ A1	4583	1400	1400	1	637
▩ AM-TMP	371	1	1	989	1
▨ A2	1117	5			241
▩ AM-MNR	18			2	2

Figure 4. Examples of concurrence statistics for POS pair and semantic roles

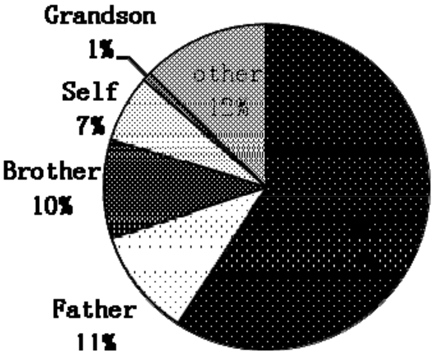


Figure 5. Examples of concurrence statistics for certain family dependency and semantic roles

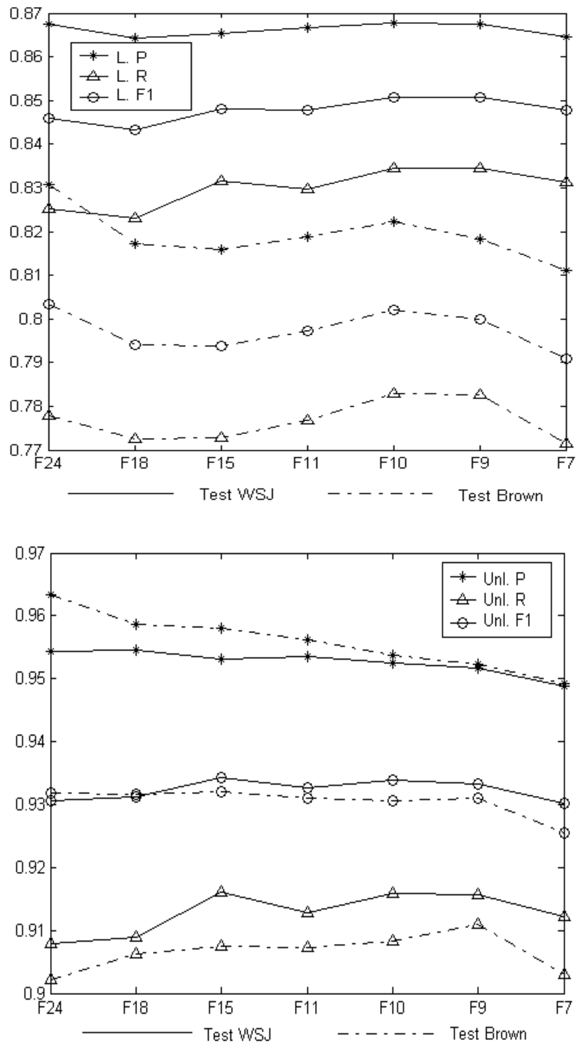


Figure 6. Testing Results and Features Combinations for Dependency labeling.

L= Labeled; Unl= Unlabeled; P= precision; R= recall;

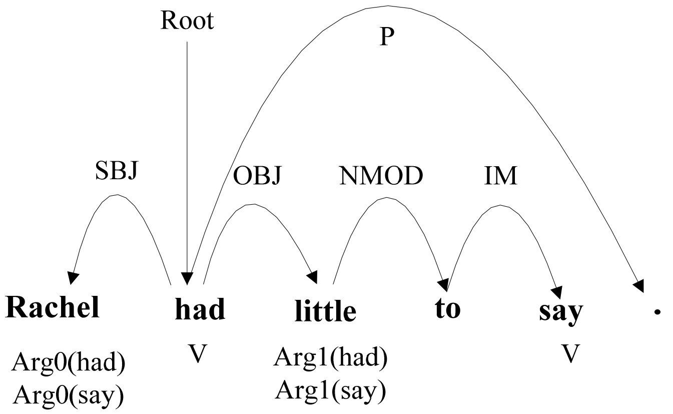


Figure 7. the recognition example with 10 features selected

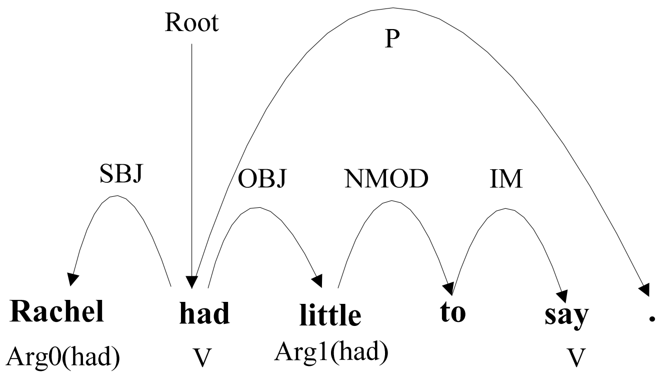


Figure 8. the recognition example with 18 features selected