Combining Constituent and Dependency Syntactic Views for Chinese Semantic Role Labeling

Shiqi Li[1], Qin Lu[2], Tiejun Zhao[1], Pengyuan Liu[3] and Hanjing Li[1]
[1]School of Computer Science and Technology,
Harbin Institute of Technology
{sqli,tjzhao,hjlee}@mtlab.hit.edu.cn
[2]Department of Computing,
The Hong Kong Polytechnic University
csluqin@comp.polyu.edu.hk
[3]Institute of Computational Linguistics,
Peking University
liupengyuan@pku.edu.cn

Abstract

This paper presents a novel feature-based semantic role labeling (SRL) method which uses both constituent and dependency syntactic views. Comparing to the traditional SRL method relying on only one syntactic view, the method has a much richer set of syntactic features. First we select several important constituent-based and dependency-based features from existing studies as basic features. Then, we propose a statistical method to select discriminative combined features which are composed by the basic features. SRL is achieved by using the SVM classifier with both the basic features and the combined features. Experimental results on Chinese Proposition Bank (CPB) show that the method outperforms the traditional constituent-based or dependency-based SRL methods.

1 Introduction

Semantic role labeling (SRL) is a major method in current semantic analysis which is important to NLP applications. The SRL task is to identify semantic roles (or arguments) of each predicate and then label them with their functional tags, such as 'Arg0' and 'ArgM' in PropBank (Palmer et al., 2005), or 'Agent' and 'Patient' in FrameNet (Baker et al., 1998).

The significance of syntactic analysis in SRL has been proven by (Gildea and Palmer, 2002; Punyakanok et al., 2005), and syntactic parsing has been applied by almost all current studies. In terms of syntactic representations, the SRL approaches are mainly divided into three categories: constituent-based, chunk-based and dependency-based. Constituent-based SRL has been studied intensively with satisfactory results. Chunk-based SRL has been found to be less effective than the constituent-based by (Punyakanok et al., 2005). In recent years, the dependency-based SRL has been greatly promoted by the CoNLL shared tasks on semantic parsing (Hajic et al., 2009). However, there is not much research on combined use of different syntactic views (Pradhan et al., 2005), on the feature level of SRL.

This paper introduces a novel method for Chinese SRL utilizing both constituent-based and dependency-based features. The method takes constituent as the basic unit of argument and adopts the labeling of PropBank. It follows the prevalent feature-based SRL methods to first turn predicate-argument pairs into flat structures by well-defined linguistic features, and then uses machine learning methods to predict the semantic labels. The method also involves two classification phases: semantic role identification (SRI) and semantic role classification (SRC). In addition, a heuristic-based pruning preprocessing (Xue and Palmer, 2004) is used to filter out a lot of apparently inappropriate constituents at the beginning.

Coling 2008: Poster Volume, pages 665–673,
Beijing, August 2010

And it has been widely reported that, in feature-based SRL, the performance can be improved by adding several combined features each of which is composed by two single features (Xue and Palmer, 2004; Toutanova et al., 2005; Zhao et al., 2009). Thus, in this work, we exploit combined use of both constituent-based and dependency-based features in addition to using features of singular types of syntactic view. We propose a statistical method to select effective combined features using both constituent-based and dependency-based features to make full use of two syntactic views.

2 Related Work

In recent years, many advances have been made on SRL using singular syntactic view, such as constituent (Gildea and Jurafsky, 2002; Xue and Palmer, 2004; Surdeanu et al., 2007), dependency (Hacioglu, 2004; Johansson and Nugues, 2008; Zhao et al., 2009), and CCG (Chen and Rambow, 2003; Boxwell et al, 2009). However, there are few studies on the use of multiple syntactic views. We briefly review the relevant studies of SRL using multiple syntactic views as follows.

Pradhan et al. (2005) built three semantic role labelers using constituent, dependency and chunk syntactic views, and then heuristically combined them at the output level. The method was further improved in Pradhan et al. (2008) which trains two semantic role labelers for constituents and dependency separately, and then uses the output of the two systems as additional features in another labeler using chunk parsing. The result shows an improvement to each labeler alone. A possible reason for the improvement is that the errors caused by different syntactic parsers are compensated. Yet, the features of different syntactic views can hardly complement each other in labeling. And the complexity of using multiple syntactic parsers is extremely high. Hacioglu (2004) proposed a SRL method to combine constituent and dependency syntactic views where the dependency parses are ob-tained through automatic mapping of constitu-ent parses. It uses the constituent parses to get candidates and then, the dependency parses to label them.

Boxwell et al. (2009) proposed a SRL method using features of three syntactic views: CCG, CFG and dependency. It primarily uses CCG-based features associated with 4 CFG-based and 2 dependency-based features. The combination of these syntactic views leads to a substantial performance improvement. Nguyen et al. (2009) proposed a composite kernel based on both constituent and dependency syntactic views and achieved a significant improvement in a relation extraction application.

3 Design Principle and Basic Features

Compared to related work, the proposed method integrates the constituent and dependency views in a collaborative manner. First, we define a basic feature set containing features from constituent and dependency syntactic views. Then, to make better use of two syntactic views, we introduce a statistical method to select effective combined features from the basic feature set. Finally we use both the basic features and the combined features to identify and label arguments. One of the drawbacks of the related work is the considerable complexity caused by multiple syntactic parsing processes. In our method, the cost of syntactic parsing will increase only slightly as we derive dependency parsing from constituent parsing using a constituent-to-dependency converter instead of using an additional dependency parser.

In our method, the feature set used for SRL consists of two parts: the basic feature set and the combined feature set built upon the basic feature set. The basic feature set can be further divided into constituent-based features and dependency-based features. Constituent features focus on hierarchical relations between multi-word constituents whereas dependency features focus on dependencies between individual words, as shown in Figure 1. Take the predicate '提高' (increased) as an example, in Figure 1(a), the NP constituent '中国的地位' (China's position) is labeled as 'Arg0'. The argument and the predicate are connected by the path of node types: 'NP-IP-VP-VP'. But in Figure 1(b), the individual word '地位' (position) is labeled as 'Arg0'. And the connection between the argument and the predicate is only one edge with the relation 'nsubj', which is more explicit than the path in the constituent structure. So the two syntactic views can complement each other on different linguistic units.

3.1 Constituent-Based Features

As a prevalent syntactic feature set for SRL, constituent-based features have been extensively studied by many researchers. In this work, we simply take 26 constituent-based features tested by existing studies, and add 8 new features define by us. Firstly, the 26 constituent-based features used by others are:

- The seven "standard" features: *predicate* (c1), *path* (c2), *phrase type* (c3), *position* (c4), *voice* (c5), *head word* (c6) and *predicate subcategorization* (c7) features proposed by (Gildea and Jurafsky, 2002).
- *Syntactic frame* (c8) feature from (Xue and Palmer, 2004).
- *Head word POS* (c9), *partial path* (c10), *first/last word in constituent* (c11/c12), *first/last POS in constituent* (c13/c14), *left/right sibling constituent* (c15/c16), *left/right sibling head* (c17/c18), *left/right sibling POS* (c19/c20), *constituent tree distance* (c21) and *temporal cue words* (c22) features from (Pradhan et al., 2004).
- *Predicate POS* (c23), *argument's parent constituent* (c24), *argument's parent constituent head* (c25) and *argument's parent constituent POS* (c26) inspired by (Pradhan et al., 2004).

Secondly, the 8 new features that we define are (we take the 'Arg0' node in Figure 1(a) as the example to illustrate them):

- *Locational cue words* (c27): a binary feature indicating whether the constituent contains location cue words, similar to the *temporal cue words* (c22). This feature is defined to distinguish the arguments with the 'ArgM-LOC' type from others.
- *POS pattern of argument's children* (c28): the left-to-right chain of the POS tags of the argument's children, e.g. 'NR-DEG-NN'.
- *Phrase type pattern of argument's children* (c29): the left-to-right chain of the phrase type labels of the argument's children, similar with the *POS pattern of argument's children* (c28), e.g. 'DNP-NP'.
- *Type of LCA and left child* (c30): The phrase type of the Lowest Common Ancestor (LCA) combined with its left child, e.g. 'IP-NP'.
- *Type of LCA and right child* (c31): The phrase type of the LCA combined with its right child, e.g. 'IP-VP'.

Three features: *bag of words of path* (c32), *bag of words of POS pattern* (c33) and *bag of words of type pattern* (c34), for generalizing three sparse features: *path* (c2), *POS pattern of argument's children* (c28) and *phrase type pattern of argument's children* (c29) by the bag-of-words representation.

3.2 Dependency-Based Features

The dependency parse can effectively represent the head-dependent relationship between words, yet, it lacks constituent information. If we want to label constituents using dependency-based features, we should firstly map each constituent to one or more appropriate words in the dependency tree. In this paper, we use the head word of a constituent to represent the constituent in the dependency parses.

The selection method of dependency-based features is similar to the method of constituent-based features. The 35 selected dependency-based features include:

- *Predicate/Argument relation type* (d1/d2), *relation path* (d3), *POS pattern of predicate's children* (d4) and *relation pattern of predicate's children* (d5) features from (Hacioglu, 2004).
- *Child relation set* (d6), *child POS set* (d7), *predicate/argument parent word* (d8/d9), *predicate/argument parent POS* (d10/d11), *left/right word* (d12/d13), *left/right POS* (d14/d15), *left/right relation* (d16/d17), *left/right sibling word* (d18/d19), *left/right sibling POS* (d20/d21) and *left/right sibling relation* (d22/d23) features as described in (Johansson and Nugues, 2008).
- *Dep-exists* (d24) and *dep-type* (d25) features from (Boxwell et al., 2009).
- *POS path* (d26), *POS path length* (d27), *REL path length* (d28) from (Che et al., 2008).
- *High/low support verb* (d29/d30), *high/low support noun* (d31/d32) features from (Zhao et al., 2009).
- *LCA's word/POS/relation* (d33/d34/d35) inspired by (Toutanova et al., 2005).

To maintain the consistency between two syntactic views, the dependency parses are generated by a constituent-to-dependency converter (Marneffe et al., 2006), which is suitable for semantic analysis as it retrieves the semantic head rather than the general syntactic head, using a set of modified Bikel's head rules.

4 Selection of Combined Features

The combined features, each of which consists of two different basic features, have proven to be positive for SRL. Several combined features have been widely used in SRL, such as '*predicate+head word*' and '*position+voice*'. But to our knowledge, there is no prior report about the selection method of combined features for SRL. The common entropy-based criteria are invalid here because the combined features always take lots of distinct values. And the greedy method is too complicated to be practical due to the large number of combinations.

In this paper, we define two statistical criteria to efficiently estimate the classification performance of each combined feature on the corpus. Inspired by Fisher Linear Discriminant Analysis (FLDA) (Fisher, 1938) in which the separation of two classes is defined as the ratio of the variance between the classes to the variance within the classes, namely larger ratio can lead to better separation between two classes, and the discriminant plane can be achieved by maximizing the separation. Therefore, in this paper, we adopt the ratio of inter-class distance to intra-class distance to measure to what extent a combined feature can partition the data.

Initially, the feature set contains only the N basic features. We construct one combined feature f_{ab} at each iteration by combining two basic features f_a and f_b, where $a,b \in [1,N]$ and $a \neq b$. We push f_{ab} into the feature set and take it as the $N+1\,th$ feature. Then, all the training instances are represented by feature vectors using the new feature set, and we then quantize the feature vectors of positive and negative data orderly to keep their intrinsic statistical difference. If the training dataset is denoted as $D:\{D_{pos}, D_{neg}\}$, then the separation criterion, namely the ratio of inter-class to intra-class distance for feature f_i can be given as

$$g(f_i) = \frac{InterDist_{f_i}(D_{pos}, D_{neg})}{IntraDist_{f_i}(D_{pos}, D_{neg})} \quad (1)$$

where the inter-class and the intra-class distance between D_{pos} and D_{neg} for feature f_i are specified by (2) and (3), respectively.

$$InterDist_{f_i}(D_{pos}, D_{neg}) = \left(Mean_{f_i}(D_{pos}) - Mean_{f_i}(D_{neg})\right)^2 \quad (2)$$

$$IntraDist_{f_i}(D_{pos}, D_{neg}) = S^2_{f_i}(D_{pos}) + S^2_{f_i}(D_{neg}) \quad (3)$$

$Mean_{f_i}(D)$ in (2) and $S_{f_i}(D)$ in (3) represents the sample mean and the corresponding sample standard deviation of feature f_i in dataset D as given in (4) and (5).

$$Mean_{f_i}(D) = \frac{\sum_{x \in D} x(i)}{|D|}, i \in [1, N+1] \quad (4)$$

$$S_{f_i}(D) = \sqrt{\frac{\sum_{x \in D}\left(Mean_{f_i}(D) - x(i)\right)^2}{N}}, i \in [1, N+1] \quad (5)$$

Essentially, the inter-class distance reflects the distance between the center of positive dataset and the center of negative dataset, and the intra-class distance indicates the intensity of all instances relative to the corresponding center. Therefore, larger ratio will lead to a better partition for a feature, as has been pointed out by FLDA. In order to compare the ratio between different combined features, we further standardize the value of $g(f_i)$ by computing its *z*-score $Z(f_i)$ which indicates how many standard deviations between a sample and its mean, as given in (6).

$$Z(f_i) = \frac{g(f_i) - \overline{g(f_i)}}{S_G} \quad (6)$$

where $\overline{g(f_i)}$ represents the sample mean as given in (7), and S_G represents the sample standard deviation of the sequence $g(f_i)$ where i ranges from *1* to *N+1* as given in (8).

$$\overline{g(f_i)} = \frac{\sum_{i=1}^{N+1} g(f_i)}{N+1}, i \in [1, N+1] \quad (7)$$

$$S_G = \sqrt{\frac{\sum_{i=1}^{N+1}(g(f_i) - \overline{g(f_i)})^2}{N}}, i \in [1, N+1] \quad (8)$$

After figuring out the $Z(f_a)$ and $Z(f_b)$ for the basic feature f_a and f_b, and $Z(f_{ab})$ for the combined feature f_{ab} by (6), we define the other criterion, namely the improvement $I(f_{ab})$ of the combined feature, as the smaller difference between the *z*-score of the combined feature and its two corresponding basic features as given in (9).

$$I(f_{ab}) = Z(f_{ab}) - \mathrm{Max}\big(Z(f_a),\ Z(f_b)\big) \qquad (9)$$

Finally, the combined feature with a negative $I(f_{ab})$ value is eliminated. Then, we will rank the combined features in terms of their *z*-score, and use the top *N* of them for later classification. The selection method based on the two criteria can effectively filter out combined features whose means have no significant difference between positive and negative data, and hence retain the potentially useful combined features for the separation. Meanwhile, it has a relatively fast speed when dealing with a large number of features in comparison to the greedy method due to its simplicity.

5 Performance Evaluation

5.1 Experimental Setting

In our experiments, we adopt the three-step strategy proposed by (Xue and Palmer, 2004). First, argument candidates are generated from the input constituent parse tree using the prevalent heuristic-based pruning algorithm in (Xue and Palmer, 2004). Then, each predicate-argument pair is converted to a flat feature structure by which the similarity between two instances can be easily measured. Finally we employ the Support Vector Machines (SVM) classifier to identify and classify the arguments. It is noteworthy that we use the same basic features, but different combined features for the identification and classification of arguments. We present the result comparison between using gold-standard parsing and automatic parsing, and also offer an analysis of the contribution of the combined features.

To evaluate the proposed method and compare it with others, we use the most commonly used corpus in Chinese SRL, Chinese Proposition Bank (CPB) version 1.0, as the dataset. The CPB corpus contains 760 documents, 10,364 sentences, 37,183 target predicates and 88,134 arguments. In this paper, we focus on six main types of semantic roles: Arg0, Arg1, Arg2, ArgM-ADV, ArgM-LOC and ArgM-TMP. The number of semantic roles of the six types accounted for 95% of all the semantic roles in CPB. For SRC, we use the one-versus-all approach, in which six SVMs will be trained to separate each semantic type from the remaining types. We divide the corpus into three parts: the first 99 documents (chtb_001.fid to chtb_099.fid) serve as the test data, the last 32 documents (chtb_900.fid to chtb_931.fid) serve as the development data and the left 629 documents (chtb_100.fid to chtb_899.fid) serve as the training data.

We use the SVM-Light Toolkit version 6.02 (Joachims, 1999) for the implementation of SVM, and use the Stanford Parser version 1.6 (Levy and Manning, 2003) as the constituent parser and the constituent-to-dependency converter. In classifications, we employ the linear kernel for SVM and set the regularization parameter to the default value which is the reciprocal of the average Euclidean norm of training data. The performance metrics are: accuracy (A), precision (P), recall (R) and *F*-score (F).

5.2 Combined Feature Selection

First, we select the combined features for classifications of SRI and SRC using the method described in Section 4 on the training data with gold-standard parse trees. Due to the limit of this paper, we only list the top-10 combined features for SRI and SRC for the 6 different types, as shown in Table 1 in which each combined feature is expressed by the IDs of its two basic features with a plus sign between them.

Rank	**SRI**	**ARG0**	**ARG1**	**ARG2**	**ADV**	**LOC**	**TMP**
1	c1+c6	c1+c6	c1+c6	c1+c6	c1+c6	c5+c27	c1+c6
2	c1+d3	c32+c30	c30+d31	c1+d1	c30+d27	c9+d17	c22+c27
3	d25+d14	c7+c6	c30+d32	c1+c7	c30+d28	c9+d13	c7+c6
4	c4+d25	c1+c2	c5+c30	c7+c6	c1+c11	c9+c2	d26+d27
5	d25+d22	c1+c12	c30+d24	c1+c5	c24+d33	c23+c27	d26+d28
6	d25+d20	c23+c6	c30+c21	c1+c23	c30+d25	c9+c20	c23+d26
7	d25+d21	c1+c3	c5+c4	c23+c6	c24+d9	c14+c32	c5+d26
8	d25+d18	c10+d35	c1+c10	c1+c3	c27+c2	c14+c10	d26+d31
9	d25+d19	c10+d1	c30+d10	c5+c6	c22+c2	c9+c26	d26+d32
10	d25+d35	c10+d28	c4+c6	c1+d5	c24+d13	c14+c2	c23+c6

Table 1. Top-10 combined features for SRI and SRC ranked by *z*-score

Table 1 shows that the commonly used combined features, such as '*predicate+head word*' (c1+c6) and '*position+voice*' (c4+c5) proposed by (Xue and Palmer, 2004) are also included. In particular, the '*predicate+head word*' feature takes first place in all semantic categories except LOC, in which the combination of the new feature '*locational cue words*' (c27) and the '*voice* (c5)' feature performs the best. The results also show that the most frequently occurred basic features in the combined set are '*predicate*' (c1), '*head word*' (c6), '*type of LCA and left child*' (c30), '*dep-type*' (d25) and '*POS path*' (d26). These basic features should be more discriminative when combined with others. Additionally, we find some other latent effective combined features, such as '*predicate subcategorization+head word*' (c7+c6), '*predicate POS+head word*' (c23+c6) and '*predicate+phrase type*' (c1+c3), whose performance will be further validated and analyzed later in this section. It is obvious that the obtained combined features for SRI and SRC are different, and the obtained combined features for each type are also different as our selection method is based on positive and negative data which are completely different for each argument type. In SRI phase, we will use the combined features for all the six semantic types (after removing duplicates).

Then, we evaluate the performance of SRL based on the top-*N* combined features. The preliminary evaluation on the development set suggests that the performance becomes stable when *N* exceeds 20. Therefore, we vary the value of *N* to 5, 10 and 20 in the experiments to evaluate the performance of combined features. Corresponding to the three different values of *N*, we finally obtained 28, 60 and 114 combined features for the SRL, respectively.

5.3 SRL Using Gold Parses

To illustrate each component of the method, we constructed 6 SRL systems using 6 different feature sets: 'Constituent Only' (CO) − uses the constituent-based features, as presented in Section 3.1; 'Dependency Only' (DO) − uses the dependency-based features, as presented in Section 3.2; 'CD' − uses both the constituent-based features and the dependency-based features, but no combined features; 'CD+Top5' − obtained by adding the top-5 combined features to the 'CD' system; and similarly for the 'CD+Top10' and the 'CD+Top20' systems. And 'CO' serves as the baseline in our experiments.

First, we evaluate the performance of SRI using the held-out test set with gold-standard constituent parse trees. The corresponding dependency parse trees are automatically generated by the constituent-to-dependency converter included in the Stanford Parser. The testing results of the six systems on the SRI phase are shown in Table 2.

System	A (%)	P (%)	R (%)	F (%)
CO	97.87	97.04	97.30	97.17
DO	92.76	92.90	84.19	88.33
CD	97.98	97.44	97.25	97.34
CD+Top5	98.12	97.56	97.58	97.57
CD+Top10	98.15	97.61	97.62	97.61
CD+Top20	98.18	97.68	97.64	97.66

Table 2. Results of SRI using gold parses

It can be seen from Table 2 that 'CD' and 'CD+Top20' give only slightly improvement over 'CO' by less than 1% point. In other words, feature combinations do not seem to be very effective for SRI. Then we label all recognized constituents in the SRI phase with one of the six semantic role types. Table 3 displays the *F*-score of each semantic type and the overall SRC on the test set with gold-standard parses.

System	Arg0	Arg1	Arg2	ADV	LOC	TMP	ALL
CO	92.40	90.57	59.98	96.25	86.80	98.14	91.23
DO	90.70	88.22	56.95	94.54	81.23	97.37	89.14
CD	92.85	91.29	63.35	96.55	87.55	98.32	91.86
CD+Top5	93.96	92.79	73.48	97.13	88.63	98.31	93.22*[1]
CD+Top10	94.15	93.23	74.18	97.42	87.17	98.57	93.41*
CD+Top20	94.10	93.19	75.13	97.23	88.05	98.48	93.46*

Table 3. Results of SRC using gold parses

Table 3 shows that the proposed method performs much better in SRC. It improves the constituent-based method by more than 2% in SRC. The effectiveness of combined features can also be clearly seen because the overall *F*-scores of the three systems using combined features all exceed 93%, significant greater than the systems using singular features. The improvement is noticeable for all semantic role types except the 'TMP' type. It means that the dependency parses cannot provide additional information to the labeling of this type. The results of Table 2 and Table 3 together show

[1] The F-score value with an asterisk (*) indicates that there is a statistically significant difference between this system and the baseline ('CO') using the chi-square test ($p<0.05$).

that our method using combined features can effectively improve the performance of SRL on the SRC phases, when using gold parses.

5.4 SRL Using Automatic Parses

To measure the performance of the algorithm in practical conditions, we replicate the above experiments using Stanford Parser on the raw texts of the test set, without segmentation or POS tagging. The dependency parses are also generated from the automatic constituent parses, as described in Section 5.3. The results are shown in Table 4.

System	A (%)	P (%)	R (%)	F (%)
CO	71.54	68.72	70.62	69.66
DO	68.86	65.06	60.68	62.79
CD	73.53	70.63	72.75	71.67*
CD+Top5	73.62	70.69	72.98	71.82*
CD+Top10	73.65	70.71	73.08	71.88*
CD+Top20	73.67	70.70	73.16	71.91*

Table 4. Results of SRI using automatic parses

Table 4 shows that the proposed method is also effective when using automatic parses despite the dramatic decrease in *F*-scores in comparison to using gold-standard parses. The decline is mainly caused by the heuristic-based pruning strategy in which a number of real arguments are pruned when using the constituent parses with errors. Further analysis shows that, in SRI using gold parses, the ratio of incorrectly pruned arguments to the total is less than 2%, but the ratio jumps to 17% when using automatic parses. Next, on the basis of the SRI results, we test the performance of SRC using the automatic parses, as shown in Table 5.

System	Arg0	Arg1	Arg2	ADV	LOC	TMP	ALL
CO	89.20	88.90	54.47	93.93	81.80	94.38	88.24
DO	88.79	89.32	50.21	91.27	78.26	93.86	87.63
CD	89.75	89.87	57.71	95.28	84.22	94.71	89.16*
CD+Top5	90.75	90.97	65.64	95.53	84.45	94.45	90.16*
CD+Top10	90.96	91.37	67.25	95.31	84.49	94.61	90.45*
CD+Top20	90.94	91.29	67.42	95.22	84.39	94.65	90.42*

Table 5. Results of SRC using auto parses

Table 5 shows only a slight decline in comparison with the result of using gold-standard parses, and it maintains the same trend of performance for each semantic role in the Table 3, which proves the validity of the proposed method when using automatic parses. Table 6 shows the *F*-score of the overall SRL on both the gold-standard and the automatic parse data.

System	Gold Parse (F%)	Auto Parse (F%)
CO	89.29	63.13
DO	82.69	60.34
CD	90.01	65.56*
CD+Top5	91.47*	66.37*
CD+Top10	91.68*	66.61*
CD+Top20	91.76*	66.61*

Table 6. Results of overall SRL

Table 6 shows that the *F*-score of the 'CD+Top20' surpasses that of the 'CO' system by more than 2% on the gold parses, and more than 3% on the automatic parse. In other words, the method using constituent and dependency syntactic views performs even more effective for the automatic parses. The last three rows of Table 6 shows that the top-10 combined features perform better than the top-5 features by adding 32 more features, but the top-20 combined features obtain similar results to the top-10 features by adding 54 more features. It suggests that only several salient combined features can actually improve the performance.

5.5 Combined Feature Performance

To evaluate the performance of each combined feature to identify the salient combined features for SRL, we rank the 60 combined features used by the 'CD+Top10' system on the test data with gold-standard parses, according to the *F*-score improvement achieved by each combined feature. Here we list the top 20 of them which are shown in Table 7.

Rank	Feature	Δ F(%)	Rank	Feature	Δ F(%)
1	c1+c6	0.611	11	c10+d1	0.413
2	c1+c10	0.593	12	c5+d26	0.404
3	c4+c6	0.557	13	c24+d9	0.395
4	c9+c20	0.503	14	d25+d35	0.395
5	c23+c6	0.494	15	c30+d24	0.377
6	c1+c3	0.458	16	c9+c26	0.377
7	c9+d13	0.449	17	c10+d28	0.368
8	c14+c10	0.431	18	c30+d29	0.365
9	c1+c5	0.422	19	c30+d30	0.361
10	c24+d33	0.413	20	c7+c6	0.361

Table 7. Top-20 combined features

As can be seen from Table 7, a half of combined features are composed by constituent features only, and the other half contain at least one dependency-based feature. This indicates that dependency features can be helpful to construct combined features for SRL. Through analyzing the performance of each combined features, we have obtained some new and effective combined features which were not recognized before, such as '*predicate+partial path*' (c1+c10), '*position+head word*' (c4+c6), '*Head word POS+right sibling POS*' (c9+c20). Observation from these combined features suggests that not all combined features are composed by two significant basic features. Some not significant ones, such as '*partial path*' (c10) and '*Head word POS*' (c9) can also produce salient combined features.

Furthermore, we find that the relative order of the combined features in Table 7 is not exactly consistent with their orders in Table 1. The inconsistency indicates that the estimation criteria used for combined features selection is not perfect. In estimation, the effect of combined features is evaluated simply based on the distance between the positive and the negative dataset by considering the efficiency. But in practice, the effects of them are determined through one-by-one classification.

5.6 Comparison to Other Work

Finally, we compare the proposed method with other four representative Chinese SRL systems. First, the 'Xue^{1}' system (Xue and Palmer, 2005) is a typical feature-based system using 9 basic features, 2 combined features and the Maximum Entropy (ME) classifier. Second, the 'Liu' system (Liu et al. 2007) which uses 19 basic features, 10 combined features and also the ME classifier. Third, the 'Che' (Che, 2008) system use a hybrid convolution tree kernel to directly measure the similarity between two constituent structures. Fourth, the 'Xue^{2}' system described in (Xue, 2008), which is similar to 'Xue^{1}' on basic framework, but using a new feature set. The 'Xue^{2}' system evaluates the SRL of the verbal predicates and the nominalized predicates separately, and offers no consolidated evaluation in (Xue, 2008). So in the comparison, we refer to its performance on the verbal predicates and the nominalized predicates as 'Xue^{21}' and 'Xue^{22}'.

All the four systems mentioned above use the constituent as the labeling unit and use the CPB corpus as the data set, the same as our method. And we use the same training and test data splits as in the 'Xue^{1}' and 'Che' systems. Table 8 shows the comparison results in terms of *F*-score on both gold parses and auto parses.

System	Gold Parse (F%)	Auto Parse (F%)
Xue^{22}	69.6	57.3
Xue^{1}	91.3	61.3
Liu	91.31	—
Che	91.67	65.42
Ours	91.76	66.61
Xue^{21}	92.0	66.8

Table 8. Comparison to other work

Table 8 shows that our method performs better than the 'Xue^{1}', 'Liu' and 'Che' systems on both gold parses and automatic parses. It is only slightly worse than the 'Xue^{21}', namely the verbal predicates part of the 'Xue^{2}' system. But for the other part of the 'Xue^{2}' system for the nominalized predicates, namely the 'Xue^{22}', our method performs much better than it. The results further verify the validity of the method.

6 Conclusions

This paper presents a novel feature-based SRL approach for Chinese. Compared to the traditional feature-based methods, the method can effectively integrate the constituent and the dependency syntactic views at the feature level. The method provides an effective way to connect two syntactic views by a statistical selection method of combined features to substantially improve the feature-based SRL method. The complexity of the method will not increase significantly compared to the method using one syntactic view as we use a constituent-to-dependency conversion rather than additional dependency parsing. The effectiveness of the method has been proven by the experiments on CPB using SVM classifier with linear kernel.

Acknowledgments

This work is supported by the Key Program of National Natural Science Foundation of China under Grant No. 60736014, the Key Project of the National High Technology Research and Development Program of China under Grant No. 2006AA010108, and the Hong Kong Polytechnic University under Grant No. G-U297 and G-U596.

References

Collins F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. *Proceedings of Coling-ACL-1998.*

Stephen A. Boxwell, Dennis Mehay, and Chris Brew. 2009. Brutus: A Semantic Role Labeling System Incorporating CCG, CFG, and Dependency Features. *Proceedings of ACL-2009.*

Wanxiang Che. 2008. *Kernel-based Semantic Role Labeling.* Ph.D. Thesis. Harbin Institute of Technology, Harbin, China.

John Chen and Owen Rambow. 2003. Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments. *Proceedings of EMNLP-2003.*

Weiwei Ding and Baobao Chang. 2008. Improving Chinese Semantic Role Classification with Hierarchical Feature Selection Strategy. *Proceedings of EMNLP-2008.*

Ronald A. Fisher. 1938. The Statistical Utilization of Multiple Measurements. *Annals of Eugenics,* 8:376-386.

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

Daniel Gildea and Martha Palmer. 2002. The Necessity of Syntactic Parsing for Predicate Argument Recognition. *Proceedings of ACL-2002*.

Kadri Hacioglu. 2004. Semantic Role Labeling Using Dependency Trees. *Proceedings of COLING-2004.*

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, et al. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. *Proceedings of CoNLL-2009.*

Thorsten Joachims. 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods. Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed), MIT Press.

Richard Johansson and Pierre Nugues. 2008. Dependency-based Semantic Role Labeling of PropBank. *Proceedings of EMNLP-2008.*

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank. *Proceedings of ACL-2003.*

Huaijun Liu, Wanxiang Che, and Ting Liu. 2007. Feature Engineering for Chinese Semantic Role Labeling. *Journal of Chinese Information Processing,* 21(2):79-85.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of LREC-2006.*

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution Kernels on Constituent, Dependency and Sequential Structures for Relation Extraction. *Proceedings of EMNLP-2009.*

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71-106

Sameer Pradhan, Wayne Waed, Kadri Haciolgu, and James H. Martin. 2004. Shallow Semantic Parsing using Support Vector Machines. *Proceedings of HLT/NAACL-2004*

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2005. Semantic Role Labeling Using Different Syntactic Views. *Proceedings of ACL-2005.*

Sameer Pradhan, Wayne Ward, and James H. Martin. 2008. Towards Robust Semantic Role Labeling. *Computational Linguistics,* 34(2): 289-310.

Vasin Punyakanok, Dan Roth, Wentau Yih. 2005. The Necessity of Syntactic Parsing for Semantic Role Labeling. *Proceedings of IJCAI-2005*.

Mihai Surdeanu, Lluis Marquez, Xavier Carreras, and Pere R. Comas. 2007. Combination Strategies for Semantic Role Labeling. *Journal of Artificial Intelligence Research*, 29:105-151.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. *Proceedings of ACL-2005.*

Nianwen Xue and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. *Proceedings of EMNLP-2004.*

Nianwen Xue and Martha Palmer. 2005 Automatic semantic role labeling for Chinese verbs. *Proceedings of IJCAI-2005.*

Nianwen Xue. 2008. Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 34(2):225-255.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009. Semantic Dependency Parsing of NomBank and PropBank: An Efficient Integrated Approach via a Large-scale Feature Selection. *Proceedings of EMNLP-2009.*