A Minimum Error Weighting Combination Strategy for Chinese Semantic Role Labeling

Tao Zhuang and **Chengqing Zong**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{tzhuang, cqzong}@nlpr.ia.ac.cn

Abstract

Many Semantic Role Labeling (SRL) combination strategies have been proposed and tested on English SRL task. But little is known about how much Chinese SRL can benefit from system combination. And existing combination strategies trust each individual system's output with the same confidence when merging them into a pool of candidates. In our approach, we assign different weights to different system outputs, and add a weighted merging stage to the conventional SRL combination architecture. We also propose a method to obtain an appropriate weight for each system's output by minimizing some error function on the development set. We have evaluated our strategy on Chinese Proposition Bank data set. With our minimum error weighting strategy, the F_1 score of the combined result achieves 80.45%, which is 1.12% higher than baseline combination method's result, and 4.90% higher than the best individual system's result.

1 Introduction

In recent years, Chinese Semantic Role Labeling has received much research effort (Sun and Jurafsky, 2004; Xue, 2008; Che et al., 2008; Ding and Chang, 2008; Sun et al., 2009; Li et al., 2009). And Chinese SRL is also included in CoNLL-2009 shared task (Hajič et al., 2009). On the data set used in (Xue, 2008), the F_1 score of the SRL results using automatic syntactic analysis is still in low 70s (Xue, 2008; Che et al., 2008; Sun et al., 2009). As pointed out by Xue (Xue, 2008), the SRL errors are mainly caused by the errors in automatic syntactic analysis. In fact, Chinese SRL suffers from parsing errors even more than English SRL, because the state-of-the-art parser for Chinese is still not as good as that for English. And previous research on English SRL shows that combination is a robust and effective method to alleviate SRL's dependency on parsing results (Màrquez et al., 2005; Koomen et al., 2005; Pradhan et al., 2005; Surdeanu et al., 2007; Toutanova et al., 2008). However, the effect of combination for Chinese SRL task is still unknown. This raises two questions at least: (1) How much can Chinese SRL benefit from combination? (2) Can existing combination strategies be improved? All existing combination strategies trust each individual system's output with the same confidence when putting them into a pool of candidates. But according to our intuition, different systems have different performance. And the system that have better performance should be trusted with more confidence. We can use our prior knowledge about the combined systems to do a better combination.

The observations above motivated the work in this paper. Instead of directly merging outputs with equal weights, different outputs are assigned different weights in our approach. An output's weight stands for the confidence we have in that output. We acquire these weights by minimizing an error function on the development set. And we use these weights to merge the outputs. In this paper, outputs are generated by a full parsing based Chinese SRL system and a shallow parsing based SRL system. The full parsing based system

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1362–1370,
Beijing, August 2010

use multiple parse trees to generate multiple SRL outputs. Whereas the shallow parsing based system only produce one SRL output. After merging all SRL outputs, we use greedy and integer linear programming combination methods to combine the merged outputs.

We have evaluated our combination strategy on Chinese Propbank data set used in (Xue, 2008) and get encouraging results. With our minimum error weighting (MEW) strategy, the F_1 score of the combined result achieves 80.45%. This is a significant improvement over the best reported SRL performance on this data set, which is 74.12% in the literature (Sun et al., 2009).

2 Related work

A lot of research has been done on SRL combination. Most of them focused on English SRL task. But the combination methods are general. And they are closely related to the work in this paper.

Punyakanok et al. (2004) formulated an Integer Linear Programming (ILP) model for SRL. Based on that work, Koomen et al. (2005) combined several SRL outputs using ILP method. Màrquez et al. (2005) proposed a combination strategy that does not require the individual system to give a score for each argument. They used a binary classifier to filter different systems' outputs. Then they used a greedy method to combine the candidates that pass the filtering process. Pradhan et al. (2005) combined systems that are based on phrase-structure parsing, dependency parsing, and shallow parsing. They also used greedy method when combining different outputs. Surdeanu et al. (2007) did a complete research on a variety of combination strategies. All these research shows that combination can improve English SRL performance by 2$\sim$5 points on F_1 score. However, little is known about how much Chinese SRL can benefit from combination. And, as we will show, existing combination strategies can still be improved.

3 Individual SRL Systems

3.1 Full Parsing Based System

The full parsing based system utilize full syntactic analysis to perform semantic role labeling. We implemented a Chinese semantic role labeling system similar to the one described in (Xue, 2008). Our system consists of an argument identification stage and an argument classification stage. In the argument identification stage, a number of argument locations are identified in a sentence. In the argument classification stage, each location identified in the first stage is assigned a semantic role label. The features used in this paper are the same with those used in (Xue, 2008).

Maximum entropy classifier is employed for both the argument identification and classification tasks. And Zhang Le's MaxEnt toolkit[1] is used for implementation.

3.2 Shallow Parsing Based System

The shallow parsing based system utilize shallow syntactic information at the level of phrase chunks to perform semantic role labeling. Sun et al. (2009) proposed such a system on Chinese SRL and reported encouraging results. The system used in this paper is based on their approach. For Chinese chunking, we adopted the method used in (Chen et al., 2006), in which chunking is regarded as a sequence labeling task with IBO2 representation. The features used for chunking are the uni-gram and bi-gram word/POS tags with a window of size 2. The SRL task is also regarded as a sequence labeling problem. For an argument with label ARG*, we assign the label B-ARG* to its first chunk, and the label I-ARG* to its rest chunks. The chunks outside of any argument are assigned the label O. The features used for SRL are the same with those used in the one-stage method in (Sun et al., 2009).

In this paper, we employ Tiny SVM along with Yamcha (Kudo and Matsumoto, 2001) for Chinese chunking, and CRF++[2] for SRL.

3.3 Individual systems' outputs

The maximum entropy classifier used in full parsing based system and the CRF model used in shallow paring based system can both output classification probabilities. For the full parsing based system, the classification probability of the ar-

[1] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

[2] http://crfpp.sourceforge.net/

gument classification stage is used as the argument's probability. Whereas for the shallow parsing based system, an argument is usually comprised of multiple chunks. For example, an argument with label ARG0 may contain three chunks labeled as: B-ARG0, I-ARG0, I-ARG0. And each chunk has a label probability. Thus we have three probabilities p_1, p_2, p_3 for one argument. In this case, we use the geometric mean of individual chunks' probabilities $(p_1 \cdot p_2 \cdot p_3)^{1/3}$ as the argument's probability.

As illustrated in Figure 1, in an individual system's output, each argument has three attributes: its location in sentence loc, represented by the number of its first word and last word; its semantic role label l; and its probability p.

Sent:	外商	投资 企业	成为 中国 外贸 重要 增长点	
Args:	[ARG0]	[pred]	[ARG1	]
loc:	(0, 2)		(4, 7)	
l:	ARG0		ARG1	
p:	0.94		0.92	

Figure 1: Three attributes of an output argument: location loc, label l, and probability p.

So each argument outputted by a system is a triple (loc, l, p). For example, the ARG0 in Figure 1 is $((0, 2), \text{ARG0}, 0.94)$. Because the outputs of baseline systems are to be combined, we call such triple a **candidate** for combination.

4 Approach Overview

As illustrated in Figure 2, the architecture of our system consists of a candidates generation stage, a weighted merging stage, and a combination stage. In the candidates generation stage, the baseline systems are run individually and their outputs are collected. We use 2-best parse trees of Berkeley parser (Petrov and Klein, 2007) and 1-best parse tree of Bikel parser (Bikel, 2004) and Stanford parser (Klein and Manning, 2003) as inputs to the full parsing based system. The second best parse tree of Berkeley parser is used here for its good quality. So together we have four different outputs from the full parsing based system. From the shallow parsing based system, we have only one output.

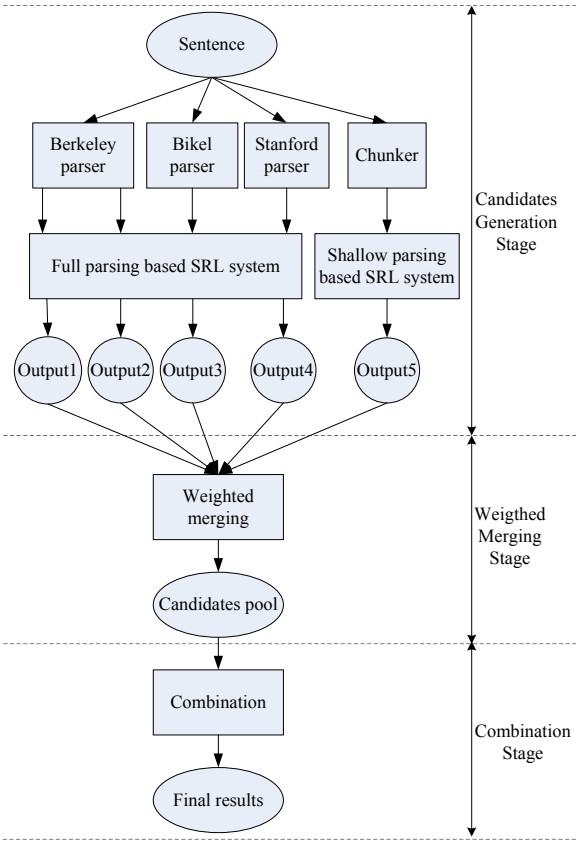

Figure 2: The overall architecture of our system.

In the weighted merging stage, each system output is assigned a weight according to our prior knowledge obtained on the development set. Details about how to obtain appropriate weights will be explained in Section 6. Then all candidates with the same loc and l are merged to one by weighted summing their probabilities. Specifically, suppose that there are n system outputs to be combined, with the i-th output's weight to be w_i. And the candidate in the i-th output with loc and l is (loc, l, p_i) (If there is no candidate with loc and l in the i-th output, p_i is 0.). Then the merged candidate is (loc, l, p), where $p = \sum_{i=1}^{n} w_i p_i$.

After the merging stage, a pool of merged candidates is obtained. In the combination stage, candidates in the pool are combined to form a consistent SRL result. Greedy and integer linear programming combination methods are experimented in this paper.

5 Combination Methods

5.1 Global constraints

When combining the outputs, two global constraints are enforced to resolve the conflict between outputs. These two constraints are:

1. *No duplication*: There is no duplication for key arguments: ARG0 $\sim$ ARG5.

2. *No overlapping*: Arguments cannot overlap with each other.

We say two argument candidates **conflict** with each other if they do not satisfy the two constraints above.

5.2 Two combination methods

Under these constraints, two methods are explored to combine the outputs. The first one is a greedy method. In this method, candidates with probability below a threshold are deleted at first. Then the remaining candidates are inspected in descending order according to their probabilities. And each candidate will be put into a solution set if it does not conflict with candidates already in the set. This greedy combination method is very simple and has been adopted in previous research (Pradhan et al., 2005; Màrquez et al., 2005).

The second combination method is integer linear programming (ILP) method. ILP method was first applied to SRL in (Punyakanok et al., 2004). Here we formulate an ILP model whose form is different from the model in (Punyakanok et al., 2004; Koomen et al., 2005). For convenience, we denote the whole label set as $\{l_1, l_2, \ldots, l_n\}$. And let $l_1 \sim l_6$ stand for the key argument labels ARG0 $\sim$ ARG5 respectively. Suppose there are m different locations, denoted as $loc_1, \ldots, loc_m$, among all candidates in the pool. And the probability of assigning l_j to loc_i is p_{ij}. A binary variable x_{ij} is defined as:

$$x_{ij} = \begin{cases} 1 & \text{if } loc_i \text{ is assigned label } l_j, \\ 0 & \text{otherwise.} \end{cases}$$

The objective of the ILP model is to maximize the sum of arguments' probabilities:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - T) x_{ij} \qquad (1)$$

where T is a threshold to prevent including too many candidates in solution. T is similar to the threshold in greedy combination method. In this paper, both thresholds are empirically tuned on development data, and both are set to be 0.2.

The inequalities in equation (2) make sure that each loc is assigned at most one label.

$$\forall 1 \leq i \leq m : \sum_{j=1}^{n} x_{ij} \leq 1 \qquad (2)$$

The inequalities in equation (3) satisfy the *No duplication* constraint.

$$\forall 1 \leq j \leq 6 : \sum_{i=1}^{m} x_{ij} \leq 1 \qquad (3)$$

For any location loc_i, let C_i denote the index set of the locations that overlap with it. Then the *No overlapping* constraint means that if loc_i is assigned a label, i.e., $\sum_{j=1}^{n} x_{ij} = 1$, then for any $k \in C_i$, loc_k cannot be assigned any label, i.e., $\sum_{j=1}^{n} x_{kj} = 0$. A common technique in ILP modeling to form such a constraint is to use a sufficiently large auxiliary constant M. And the constraint is formulated as:

$$\forall 1 \leq i \leq m : \sum_{k \in C_i} \sum_{j=1}^{n} x_{kj} \leq (1 - \sum_{j=1}^{n} x_{ij}) M \qquad (4)$$

In this case, M only needs to be larger than the number of candidates to be combined. In this paper, $M = 500$ is large enough. And we employ lpsolve[3] to solve the ILP model.

Note that the form of the ILP model in this paper is different from that in (Punyakanok et al., 2004; Koomen et al., 2005) in three aspects: (1) A special label class *null*, which means no label is assigned, was added to the label set in (Punyakanok et al., 2004; Koomen et al., 2005). Whereas no such special class is needed in our model, because if no label is assigned to loc_i, $\sum_{j=1}^{n} x_{ij} = 0$ would simply indicate this case. This makes our model contain fewer variables. (2) Without *null* class in our model, we need to use a different technique to formulate the *No-overlapping* constraint. (3) In order to compare

[3] http://lpsolve.sourceforge.net/

with the greedy combination method, the ILP model in this paper conforms to exactly the same constraints as the greedy method. Whereas many more global constraints were taken into account in (Punyakanok et al., 2004; Koomen et al., 2005).

6 Train Minimum Error Weights

The idea of minimum error weighting is straightforward. Individual outputs $O_1, O_2, \ldots, O_n$ are assigned weights $w_1, w_2, \ldots, w_n$ respectively. These weights are normalized, i.e., $\sum_{i=1}^{n} w_i = 1$. An output's weight can be seen as the confidence we have in that output. It is a kind of prior knowledge we have about that output. We can gain this prior knowledge on the development set. As long as the data of the development set and the test set are similar, this prior knowledge should be able to help to guide SRL combination on test set. In this section, we discuss how to obtain appropriate weights.

6.1 Training model

Suppose the golden answer and SRL result on development set are d and r respectively. An **error function** $Er(r, d)$ is a function that measures the error contained in r in reference to d. An error function can be defined as the number of wrong arguments in r. It can also be defined using precision, recall, or F_1 score. For example, $Er(r, d) = 1 - Precision(r, d)$, or $Er(r, d) = 1 - F_1(r, d)$. Smaller value of error function means less error in r.

The combination process can also be seen as a function, which maps the outputs and weights to the combined result r: $r = Comb(O_1^n, w_1^n)$. Therefore, the error function of our system on development set is:

$$Er(r, d) = Er(Comb(O_1^n, w_1^n), d) \qquad (5)$$

From equation (5), it can be seen that: Given development set d, if the outputs to be combined O_1^n and the combination method $Comb$ are fixed, the error function is just a function of the weights. So we can obtain appropriate weights by minimizing the error function:

$$\hat{w}_1^n = \arg\min_{w_1^n} Er(Comb(O_1^n, w_1^n), d) \qquad (6)$$

6.2 Training algorithm

Algorithm 1 Powell Training Algorithm.

1: **Input** : Error function $Er(\boldsymbol{w})$.
2: Initialize n directions $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n$, and a start point $\boldsymbol{w}$ in R^n.
3: Set termination threshold δ.
4: **do**:
5: $\quad \boldsymbol{w}_1 \leftarrow \boldsymbol{w}$
6: $\quad$ **for** $i \leftarrow 1, \ldots, n$:
7: $\quad\quad \alpha_i \leftarrow \arg\min_{\alpha} f(\boldsymbol{w}_i + \alpha \boldsymbol{d}_i)$
8: $\quad\quad \boldsymbol{w}_{i+1} \leftarrow \boldsymbol{w}_i + \alpha_i \boldsymbol{d}_i$
9: $\quad \boldsymbol{d}_{n+1} \leftarrow \boldsymbol{w}_{n+1} - \boldsymbol{w}$
10: $\quad \alpha^* \leftarrow \arg\min_{\alpha} f(\boldsymbol{w} + \alpha \boldsymbol{d}_{n+1})$
11: $\quad \boldsymbol{w}' \leftarrow \boldsymbol{w} + \alpha^* \boldsymbol{d}_{n+1}$
12: $\quad \Delta Er \leftarrow Er(\boldsymbol{w}) - Er(\boldsymbol{w}')$
13: $\quad i \leftarrow \arg\max_{1 \le j \le n} Er(\boldsymbol{w}_j) - Er(\boldsymbol{w}_{j+1})$
14: $\quad$ **if** $(\alpha^*)^2 \ge \frac{\Delta Er}{Er(\boldsymbol{w}_i) - Er(\boldsymbol{w}_{i+1})}$:
15: $\quad\quad$ **for** $j \leftarrow i, \ldots, n$:
16: $\quad\quad\quad \boldsymbol{d}_j \leftarrow \boldsymbol{d}_{j+1}$
17: $\quad \boldsymbol{w} \leftarrow \boldsymbol{w}'$
18: **while** $\Delta Er > \delta$
19: **Output**: The minimum error weights $\boldsymbol{w}$.

There are two difficulties to solve the optimization problem in equation 6. The first one is that the error function cannot be written to an analytical form. This is because the $Comb$ function, which stands for the combination process, cannot be written as an analytical formula. So the problem cannot be solved using canonical gradient-based optimization algorithms, because the gradient function cannot be derived. The second difficulty is that, according to our experience, the error function has many local optima, which makes it difficult to find a global optima.

To resolve the first difficulty, Modified Powell's method (Yuan, 1993) is employed to solve the optimization problem. Powell's method is a heuristic search method that does not require the objective function to have an explicit analytical form. The training algorithm is presented in Algorithm 1. In Algorithm 1, the line search problem in steps 7 and 10 is solved using Brent's method (Yuan, 1993). And the temination threshold δ is empirically set to be 0.001 in this paper.

To resolve the second difficulty, we perform multiple searches using different start points, and then choose the best solution found.

7 Experiments

7.1 Experimental setup

We use Chinese Proposition Bank (CPB) 1.0 and Chinese Tree Bank (CTB) 5.0 of Linguistic Data Consortium corpus in our experiments. The training set is comprised of 648 files(chtb_081.fid to chtb_885.fid). The development set is comprised of 40 files(chtb_041.fid to chtb_080.fid). The test set is comprised of 72 files(chtb_001.fid to chtb_040.fid and chtb_900.fid to chtb_931.fid).

The same data setting has been used in (Xue, 2008; Ding and Chang, 2008; Sun et al., 2009). Sun et al. (2009) used sentences with golden segmentation and POS tags as input to their SRL system. However, we use sentences with only golden segmentation as input. Then we perform automatic POS tagging using Stanford POS tagger (Toutanova et al., 2003). In (Xue, 2008), the parser used by the SRL system is trained on the training and development set plus 275K words of broadcast news. In this paper, all parsers used by the full parsing based system are trained on the training set plus the broadcast news portion of CTB6.0. And the chunker used in the shallow parsing based system is trained just on the training set.

7.2 Individual outputs' performance

In this paper the four outputs of the full parsing based system are represented by FO1 $\sim$ FO4 respectively. Among them, FO1 and FO2 are the outputs using the first and second best parse trees of Berkeley parser, FO3 and FO4 are the outputs using the best parse trees of Stanford parser and Bikel parser respectively. The output of the shallow parsing based system is represented by SO. The individual outputs' performance on development and test set are listed in Table 1.

From Table 1 we can see that the performance of individual outputs are similar on development set and test set. On both sets, the F_1 scores of individual outputs are in the same order: FO1 $>$ FO2 $>$ SO $>$ FO3 $>$ FO4.

Data set	Outputs	$P(\%)$	$R(\%)$	F_1
development	FO1	**79.17**	**72.09**	**75.47**
	FO2	77.89	70.56	74.04
	FO3	72.57	67.02	69.68
	FO4	75.60	63.45	69.00
	SO	73.72	67.35	70.39
test	FO1	**80.75**	**70.98**	**75.55**
	FO2	79.44	69.37	74.06
	FO3	73.95	66.37	70.00
	FO4	75.89	63.26	69.00
	SO	75.69	67.90	71.59

Table 1: The results of individual systems on development and test set.

7.3 Combining outputs of full parsing based system

In order to investigate the benefit that the full parsing based system can get from using multiple parsers, we combine the four outputs FO1 $\sim$ FO4. The combination results are listed in Table 2. In tables of this paper, "Grd" and "ILP" stand for greedy and ILP combination methods respectively, and "+MEW" means the combination is performed with MEW strategy.

	$P(\%)$	$R(\%)$	F_1
Grd	**82.68**	73.36	77.74
ILP	82.21	73.93	77.85
Grd+MEW	81.30	75.38	78.23
ILP+MEW	81.27	**75.74**	**78.41**

Table 2: The results of combining outputs of full parsing based system on test set.

	Er	FO1	FO2	FO3	FO4
Grd	$1-F_1$	0.31	0.16	0.30	0.23
ILP	$1-F_1$	0.33	0.10	0.27	0.30

Table 3: The minimum error weights for the results in Table 2.

From Table 2 and Table 1, we can see that, without MEW strategy, the F_1 score of combination result is about 2.3% higher than the best individual output. With MEW strategy, the F_1 score is improved about 0.5% further. That is to say, with MEW strategy, the benefit of combination is improved by about 20%. Therefore, the effect of MEW is very encouraging.

Here the error function for MEW training is chosen to be $1 - F_1$. And the trained weights for greedy and ILP methods are listed in Table 3

separately. In tables of this paper, the column Er corresponds to the error function used for MEW strategy.

7.4 Combining all outputs

We have also combined all five outputs. The results are listed in Table 4. Compared with the results in Table 2, we can see that the combination results is largely improved, especially the recall.

	$P(\%)$	$R(\%)$	F_1
Grd	**83.64**	75.32	79.26
ILP	83.31	75.71	79.33
Grd+MEW	83.34	77.47	80.30
ILP+MEW	83.02	**78.03**	**80.45**

Table 4: The results of combining all outputs on test set.

From Table 4 and Table 1 we can see that without MEW strategy, the F_1 score of combination result is about 3.8% higher than the best individual output. With MEW, the F_1 score is improved further by more than 1%. That means the benefit of combination is improved by over 25% with MEW strategy.

Here the error function for MEW training is still $1 - F_1$, and the trained weights are listed in Table 5.

	Er	FO1	FO2	FO3	FO4	SO
Grd	$1 - F_1$	0.23	0.12	0.23	0.20	0.22
ILP	$1 - F_1$	0.24	0.08	0.22	0.21	0.25

Table 5: The minimum error weights for the results in Table 4.

7.5 Using alternative error functions for minimum error weights training

In previous experiments, we use $1 - F_1$ as error function. As pointed out in Section 6, the definition of error function is very general. So we have experimented with two other error functions, which are $1 - Precision$, and $1 - Recall$. Obviously, these two error functions favor precision and recall separately. The results of combining all five outputs using these two error functions are listed in Table 6, and the trained weights are listed in Table 7.

From Table 6 and Table 4, we can see that when $1 - Precison$ is used as error function, the precision of combination result is largely improved. But the recall decreases a lot. Similar effect of the error function $1 - Recall$ is also observed.

	Er	$P(\%)$	$R(\%)$	F_1
Grd+MEW	$1 - P$	85.31	73.42	78.92
ILP+MEW	$1 - P$	**85.62**	72.76	78.67
Grd+MEW	$1 - R$	81.94	77.55	79.68
ILP+MEW	$1 - R$	79.74	**78.34**	79.03

Table 6: The results of combining all outputs with alternative error functions.

	Er	FO1	FO2	FO3	FO4	SO
Grd	$1 - P$	0.25	0.24	0.22	0.22	0.07
ILP	$1 - P$	0.30	0.26	0.20	0.15	0.09
Grd	$1 - R$	0.21	0.10	0.17	0.15	0.37
ILP	$1 - R$	0.24	0.04	0.10	0.22	0.39

Table 7: The minimum error weights for the results in Table 6.

The results of this subsection reflect the flexibility of MEW strategy. This flexibility comes from the generality of the definition of error function. The choice of error function gives us some control over the results we want to get. We can define different error functions to favor precision, or recall, or some error counts such as the number of misclassified arguments.

7.6 Discussion

In this paper, the greedy and ILP combination methods conform to the same simple constraints specified in Section 5. From the experiment results, we can see that ILP method generates slightly better results than greedy method.

In Subsection 7.4, we see that combining all outputs using ILP method with MEW strategy yields 4.90% improvement on F_1 score over the best individual output FO1. In order to understand each output's contribution to the improvement over FO1. We compare the differences between outputs.

Let C_O denote the set of correct arguments in an output O. Then we get the following statistics when comparing two outputs A and B: (1) the number of common correct arguments in A and B, i.e., $|C_A \cap C_B|$; (2) the number of correct arguments in A and not in B, i.e., $|C_A \setminus C_B|$; (3) the number of correct arguments in B and not in A, i.e., $|C_B \setminus C_A|$. The comparison results between

some outputs on test set are listed in Table 8. In this table, UF stands for the union of the 4 outputs FO1 $\sim$ FO4.

A	B	$\lvert C_A \cap C_B \rvert$	$\lvert C_A \setminus C_B \rvert$	$\lvert C_B \setminus C_A \rvert$
FO1	FO2	5498	508	372
	FO3	5044	962	552
	FO4	4815	1191	512
	SO	4826	1180	920
UF	SO	5311	1550	435

Table 8: Comparison between outputs on test set.

From Table 8 we can see that the output SO has 4826 common correct arguments with FO1, which is relatively small. And, more importantly, SO contains 920 correct arguments not in FO1, which is much more than any other output contains. Therefore, SO is more complementary to FO1 than other outputs. On the contrary, FO2 is least complementary to FO1. Even compared with the union of FO1 $\sim$ FO4, SO still contains 435 correct arguments not in the union. This shows that the output of shallow parsing based system is a good complement to the outputs of full parsing based system. This explains why recall is largely improved when SO is combined in Subsection 7.4. From the analysis above we can also see that the weights in Table 5 are quite reasonable. In Table 5, SO is assigned the largest weight and FO2 is assigned the smallest weight.

In Subsection 7.3, the MEW strategy improves the benefit of combination by about 20%. And in Subsection 7.4, the MEW strategy improves the benefit of combination by over 25%. This shows that the MEW strategy is very effective for Chinese SRL combination.

To our best knowledge, no results on Chinese SRL combination has been reported in the literature. Therefore, to compare with previous results, the top two results of single SRL system in the literature and the result of our combination system on this data set are listed in Table 9. For the results in Table 9, the system of Sun et al. uses sentences with golden POS tags as input. Xue's system and our system both use sentences with automatic POS tags as input. The result of Sun et al. (2009) is the best reported result on this data set in the literature.

	POS	$P(\%)$	$R(\%)$	F_1
(Xue, 2008)	auto	76.8	62.5	68.9
(Sun et al., 2009)	gold	79.25	69.61	74.12
Ours	auto	**83.02**	**78.03**	**80.45**

Table 9: Previous best single system's results and our combination system's result on this data set.

8 Conclusions

In this paper, we propose a minimum error weighting strategy for SRL combination and investigate the benefit that Chinese SRL can get from combination. We assign different weights to different system outputs and add a weighted merging stage to conventional SRL combination system architecture. And we also propose a method to train these weights on development set. We evaluate the MEW strategy on Chinese Propbank data set with greedy and ILP combination methods.

Our experiments have shown that the MEW strategy is very effective for Chinese SRL combination, and the benefit of combination can be improved over 25% with this strategy. And also, the MEW strategy is very flexible. With different definitions of error function, this strategy can favor precision, or recall, or F_1 score. The experiments have also shown that Chinese SRL can benefit a lot from combination, especially when systems based on different syntactic views are combined. The SRL result with the highest F_1 score in this paper is generated by ILP combination together with MEW strategy. In fact, the MEW strategy is easy to incorporate with other combination methods, just like incorporating with the greedy and ILP combination methods in this paper.

Acknowledgment

The research work has been partially funded by the Natural Science Foundation of China under Grant No. 60975053, 90820303 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2006AA010108-4, and also supported by the China-Singapore Institute of Digital Media (CSIDM) project under grant No. CSIDM-200804.

References

Daniel Bikel. 2004. Intricacies of Collins Parsing Model. *Computational Linguistics*, 30(4):480-511.

Wanxiang Che, Min Zhang, Ai Ti Aw, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. Using a Hybrid Convolution Tree Kernel for Semantic Role Labeling. *ACM Transactions on Asian Language Information Processing*, 2008, 7(4).

Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. 2006. An empirical study of Chinese chunking. In *Proceedings of COLING/ACL-2006*.

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

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue and Yi Zhang. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of CoNLL-2009*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL-2003*.

Peter Koomen, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems. In *Proceedings of CoNLL-2005 shared task*.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with Support Vector Machines. In *Proceedings of NAACL-2001*.

Junhui Li, Guodong Zhou, Hai Zhao, Qiaoming Zhu, and Peide Qian. 2009. Improving Nominal SRL in Chinese Language with Verbal SRL Information and Automatic Predicate Recognition. In *Proceedings of EMNLP-2009*.

Lluís Màrquez, Mihai Surdeanu, Pere Comas, and Jordi Turmo. 2005. A Robust Combination Strategy for Semantic Role Labeling. In *Proceedings of EMNLP-2005*.

Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized parsing. In *Proceedings of ACL-2007*.

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

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic Role Labeling via Integer Linear Programming Inference. In *Proceedings of COLING-2004*.

Honglin Sun and Daniel Jurafsky. 2004. Shallow semantic parsing of Chinese. In *Proceedings of NAACL-2004*.

Weiwei Sun, Zhifang Sui, Meng Wang, and Xin Wang. 2009. Chinese Semantic Role Labeling with Shallow Parsing. In *Proceedings of EMNLP-2009*.

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

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2008. A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2): 145-159.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL-2003*.

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

Yaxiang Yuan. 1993. *Numerical Methods for Nonlinear Programming*. Shanghai Scientific and Technical Pulishers, Shanghai.