An Improved Semantic Role Labeling System for Chinese

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Third Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

We present an improved semantic role labeling system for Chinese parsed sentences. Our new system outperformed the previous reported one from two aspects: knowledge acquisition{utilization} and model design. As to the former, the semantic knowledge obtained from E-Hownet were utilized to solve the data sparseness issue; as to the latter, a combination of back-off models was proposed for semantic role classification. They enhanced the performance by ?? and ??, respectively. Further performance gains through post-processing lead to an overall accuracy improvement from 92.71% to 94.81%.

1 Introduction

A semantic role labeling (SRL) system for Chinese was reported by (You and Chen, 2004). Input to the system was a parse tree, parsed by Sinica Chinese parser[1], and output was its semantically labeled counterpart. Sinica Treebank (Chen et al., 2003), a semantically annotated Chinese treebank, was used to train simple probabilistic models, which were used for semantic role classification. As far feature set, the system relied on a number of conventional lexical (e.g. `head_word`, `head_word_pos`, `target_word`, `target_word_pos`) as well as syntactic features (e.g. `phrase_type`, `position`). The overall accuracy of the system was reported to be 92.71%. Even though this accuracy can be considered on the higher side, there was room for improvement, especially in two subtasks: data sparseness handling and classification approach.

[1]An online demo of the parser is available at http://parser.iis.sinica.edu.tw/

Data sparseness handling first, the old system used a back-off strategy to tackle this problem (see (You and Chen, 2004) for more details). This considerably improved the baseline performance, but data sparseness still remained an issue. A major reason was dependency of the system on non-generalized lexical features (e.g. `head_word`, `target_word`). In this paper, we use E-HowNet (**?**), a semantic knowledge-base for Chinese, to generalize two lexical features (i.e. `head_word` and `target_word`), and hence, to better address the data sparseness concern.

Classification methodology next. In the old system, simple probabilistic models, together with a back-off strategy, were used for semantic role assignment. As shown below, a particular feature combination was used for each model, and the backing-off was based on number of examples seen in the training data.

```
if#of(h,h_pos,t,t_pos,pt,position)
                          >threshold
 P(r|constituent)=
 P(r|h,h_pos,t,t_pos,pt,position)
else
 if#of(h_pos,t,t_pos,pt,position)
                          >threshold
  P(r|constituent)=
  P(r|h_pos,t,t_pos,pt,position)
else
......
```

A critical observation is that backing-off based on a constant threshold value did not allow to exploit the statistical knowledge in the best possible way. The reason is that if enough examples, with a more constrained feature combination, have been seen in the training data, it did not allow to utilize the less constrained statistical information, which may suggest a more probable role (see Section 4.3 for an example). We propose a different strategy that is based on a combination of weighted simple

probabilistic models. In our method, both more and less constrained probabilities are ranked by learned weights, and then the top ranked information is used for final decision making. The probabilities are calculated using the same training data (i.e. Sinica treebank), and optimal weights, which encode the worth of a particular feature combination in determining the semantic role, are found using genetic algorithms. To show the effectiveness of our idea, we build a number of systems that are based on other well established classification approaches (e.g. NaiveBayes, Decision Trees, Maximum Entropy, Linear Interpolation), and compare their outcomes to our system. The experimental results show that our strategy outperformed all other systems including the previous system, and lead to a considerable improvement in the accuracy of our final SRL system.

The rest of the paper is organized as follows: Section 2 briefly outlines our feature set, and reports how we have generalized the lexical features. This is followed by a description of probabilistic models in Section 3. Section 4 explains our classification method with the help of an example. Experiments and evaluation results are given in Section 5, which is followed by conclusion and future work section.

2 Feature Set

In addition to the set of lexical and syntactic features, mentioned in the introduction section and used by (You and Chen, 2004), we have used the following features:

- `pos_left_right_child`: part of speech tags of the immediate left and right siblings of a test node.
- `passive`: A sentence-level boolean feature indicating weather the sentence, containing the test node, is passive or not.
- `all_pos`: A set of part of speech tags of all nodes under a test node including the test node itself.
- `all_semType`: A set of semantic types of all nodes of the tree, the test node is a child of.

As pointed out by (Gildea and Jurafsky, 2002), lexical statistics, though very useful for semantic role labeling, often becomes a source of data sparseness. This is because, for a particular test case, the lexical values may not have been seen in the training data due to a large vocabulary size. In our system, we have generalized two lexical features (i.e. `head_word`, and `target_word`) by replacing them with more general features (i.e. `semanticType_head_word` and `semanticType_target_word` respectively) extracted from the Chinese semantic knowledgebase (E-HowNet).

E-HowNet is an entity-relation model in which words have been grouped together based on their semantic relationships. This grouping, in other words, provides a way to generalize the lexical features, and hence can be used to solve the data sparseness issue.

Table 1 gives coverage statistics of each model (see Section 3 for details about the models) before and after adding these generalizations. As can be seen, after generalization, the coverage improved considerably for each model, which ultimately resulted in better performance.

Model	Before	After
1	7.10%	12.68%
2	52.06%	73.05%
3	67.43%	82.68%
4	74.56%	93.35%
5	97.97%	97.97%
6	28.60%	64.20%
7	76.30%	94.23%
8	82.97%	96.85%
9	99.78%	99.78%
10	90.09%	99.00%

Table 1: Coverage Statistics

3 Probabilistic Models

Ten simple probabilistic models, each based on a particular feature combination, were build using the Sinical treebank labeled data. The probabilities for each model were estimated using the following simple formula.

$$\begin{aligned} & P(r|constituent) \\ = \; & P(r|f_c) \\ = \; & \#(r, f_c)/\#f_c \end{aligned}$$

Where f_c represents a particular feature combination. A number of combinations were tried, and for the final system we used the following set of ten combinations.

{(semType_h_word,h_pos,
semType_t_word,t_pos,pt,position,
all_pos,passive,all_semType,
left_right_child_pos),
(h_pos,semType_t_word,t_pos,pt,
position,passive),
(semType_h_word,h_pos,t_pos,pt,
position,passive),
(semType_t_word,t_pos,pt,passive,
position),
(h_pos,t_pos,pt,position,passive),
(semType_h_word,semType_t_word,
pt,position,passive),
(semType_t_word,t_pos,pt,passive),
(semType_t_word,t_pos,passive),
(t_pos,pt,position,passive),
(semType_t_word)}

4 Classification Method

4.1 Notations

- Let F be the set of feature combinations (given in the previous section), and f_c be an element of F
- P be the set of corresponding probabilistic models, and P_{f_c} be an element of P with feature combination f_c
- D_{f_c} be the probability distribution computed by P_{f_c}
- $M_{((p,r)|f_c)}$ be the most probable $(probability, role)$ pair from D_{f_c}
- W be a set of optimal weights, w be an element of W, and wp be a weighted probability (i.e. rank)

4.2 Algorithm

Our classification algorithm is as follows:

1. For a test candidate, extract values of all features (mentioned in section 2) using the parse tree and E-HowNet.
2. initialize $potential\_roles \leftarrow empty$
3. For each $f_c \in F$ do:
   - Find probability distribution D_{f_c} using the corresponding P_{f_c} model.
   - Select $M_{((p,r)|f_c)}$ from D_{f_c}.
   - Rank $M_{((p,r)|f_c)}$ by multiplying p with the corresponding w from W.
   - Append (wp, r) to $potential\_roles$.
4. return the top ranked r from $potential\_roles$.

4.3 An Example

Suppose we want to assign a semantic role to the circled node of the parse tree given in Figure 1. First, we have to extract the full set

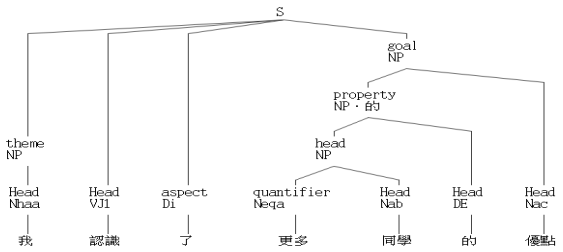

Figure 1: An example parsed sentence

of (feature:value) pairs for the target test node. The extracted set is given below:

{(semType_h_word:advantage),
(h_pos:Nac),(semType_t_word:human),
(t_pos:NP??),(pt:NP),(position:-1),
(all_pos:NP??-NP-Neqa-Nab-DE),
(passive:False),
(left_right_child_pos:empty-Nac),
(all_semType:
BecomeMore-human-tool-advantage)}

The probability distribution estimated by each of the ten probabilistic models is given in Table[2] 2. Next, from each distribution, we can select $M_{((p,r)|f_c)}$. The full list is given below:

[NULL,(0.5072,'possessor'),
(0.5,'property'),
(0.5035,'property'),
(0.8514,'property'),
(0.5,'property'),
(0.5035,'property'),
(0.5035,'property'),
(0.8598,'property'),
(0.1915,'agent')]

Note that a NULL is inserted for the models that do not have any probability distribution.
Finally, each role in this list is ranked by multiplying its probability to the corresponding weight from W. As mentioned previously, genetic algorithms and a held out development data-set were

[2]Note that due to the sparseness of the training data, their is no probability distribution for the feature combination of model 1.

P_{f_c}	**Probability Distribution** (D_{f_c})
1	
2	[(0.5072, 'possessor'), (0.4783, 'property'), 0.0145, 'quantifier')]
3	[(0.5, 'property'), (0.5, 'possessor')]
4	[(0.5035, 'property'), (0.4894, 'possessor'), (0.0071, 'quantifier')]
5	[(0.8514, 'property'), (0.1361, 'possessor'), (0.0083, 'quantifier'), (0.0042, 'apposition')]
6	[(0.5, 'property'), (0.5, 'possessor')]
7	[(0.5035, 'property'), (0.4894, 'possessor'), (0.0071, 'quantifier')]
8	[(0.5035, 'property'), (0.4894, 'possessor'), (0.0071, 'quantifier')]
9	[(0.8598, 'property'), (0.1240, 'possessor'), (0.0132, 'quantifier'), (0.0015, 'apposition'), (0.0011, 'predication'), (0.0004, 'frequency')]
10	[(0.1915, 'agent'), (0.1461, 'theme'), (0.1361, 'goal'), (0.1355, 'property'), (0.0984, 'range'), (0.0775, 'DUMMY'), (0.0540, 'possessor'), (0.0533, 'apposition'), (0.0414, 'experiencer'), (0.0267, 'DUMMY2'), (0.0230, 'DUMMY1'), (0.0073, 'topic'), (0.0024, 'location'), (0.0021, 'quantifier'), (0.0021, 'causer'), (0.0007, 'predication'), (0.0007, 'manner'), (0.0003, 'time'), (0.0003, 'particle'), (0.0002, 'complement'), (0.0002, 'comparison'), (0.0001, 'target'), (0.0001, 'source'), (0.0001, 'hypothesis'), (0.0001, 'companion')]

Table 2: Probability distributions for the test constituent

used to find W. The observed optimal set after 100 generations is given below:

```
{(w1:0.9),(w2:1.0),(w3:0.9),
(w4:0.7),(w5:0.8),(w6:0.4),
(w7:0.5),(w8:0.4),(w9:0.5),
(w10:0.4)}
```

The final list of ranked roles is:

```
[NULL,(0.5072,'possessor'),
(0.45,'property'),
(0.505307,'property'),
(0.68112,'property'),
(0.2,'property'),
(0.25175,'property'),
(0.2014,'property'),
(0.4299,'property'),
(0.0766,'agent')]
```

The top ranked `(probability,role)` pair in this list is `(0.68112,'property')`, hence the role `'property'` will be assigned to the test constituent by our classification method.

One can note that, in the above example, model 5, which is less constrained compared to model 2, suggested a more probable and correct role. This is where our model differed and outperformed the previously reported model. However, one can argue that more constrained models are supposed to be more reliable. In our approach, this factor is encoded by weights, and we can see that models with more feature constraints tend to have higher weights, while it is the other way around for the less constrained models.

5 Experiments and Evaluation

To evaluate the performance of our system, and to show the usefulness of our approach, we have build the following five semantic role labeling systems:

- **System 1:** Based on Decision Tree classifier
- **System 2:** Based on Naive Bays classifier
- **System 3:** Based on Maximum Entropy classifier
- **System 4:** Based on simple probabilistic models together with linear interpolation
- **System 5:** Based on our classification approach

We used Sinica Treebank as our taring and testing data, and a 10-fold cross validation scheme to test each system. Results are given in Table 3. From the results, we can see that our system outperformed all other systems with system 4 being the most competitive one.

System	**Accuracy**	**Precision**
1	91.13%	
2	92.54%	
3	92.70%	
4	94.32%	
5	94.81%	

Table 3: Evaluation results

To further enhance the performance of our system, we added a post-processing component to fix some of the obvious mistakes made by the probabilistic models. One such example is disambiguation between possessor/property roles. We used a heuristic rule that if the semantic type of a target word is `human`, it is more likely to be a possessor than a property. With few other such rules, the scores were improved from 94.58% to 94.81%.

6 Conclusion and Future Work

We have presented two major revisions to an already existing semantic role labeling system for Chinese. One is related to generalization of the lexical features using a semantic knowledge-base, which in turn helped to handle the data sparseness. The other is in the classification method. We have presented a model that uses the statistical knowledge in a more efficient way, and hence improves the performance. The final accuracy of our revised system is 94.81%, which is statistically ??% better than the previously reported score, which was 94.71%.

In future we would like to test our classification method on Chinese Propbank data and see if we can improve on the state-of-the-art SRL scores.

References

Keh-Jiann Chen, Chi-Ching Luo, Ming-Chung Chang, Feng-Yi Chen, ChaoJan Chen, and Chu-Ren Huang. 2003. Sinica treebank: Design criteria, representational issues and implementation. *Anne Abeille (Ed.) Treebanks Building and Using Parsed Corpora. Language and Speech series. Dordrecht:Kluwer*, pages pp231–248.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288, September.

Xiaojun Lin, Meng Zhang, and Xihong Wu. 2010. Chinese semantic role labeling with hierarchical semantic knowledge. In *Proceedings of the 2010 International Conference on Electrical and Control Engineering*, ICECE '10, pages 583–586, Washington, DC, USA. IEEE Computer Society.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, and James H. Martin. 2004. Shallow semantic parsing using support vector machines.

Jia-Ming You and Keh-Jiann Chen. 2004. Automatic semantic role assignment for a tree structure. In Oliver Streiter and Qin Lu, editors, *ACL SIGHAN Workshop 2004*, pages 109–115, Barcelona, Spain, July. Association for Computational Linguistics.

Beñat Zapirain, Eneko Agirre, and Lluís Màrquez. 2007. Ubc-upc: Sequential srl using selectional preferences: An aproach with maximum entropy markov models. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 354–357, Stroudsburg, PA, USA. Association for Computational Linguistics.