LETTER

Incorporating Frame Information to Semantic Role Labeling*

Joo-Young LEE†, **Young-In SONG**†, ***Nonmembers***, **Hae-Chang RIM**†a), ***Member***, ***and*** **Kyoung-Soo HAN**††, ***Nonmember***

SUMMARY In this paper, we suggest a new probabilistic model of semantic role labeling, which uses the frameset of the predicate as explicit linguistic knowledge for providing global information on the predicate-argument structure that local classifier is unable to catch. The proposed model consists of three sub-models: role sequence generation model, frameset generation model, and matching model. The role sequence generation model generates the semantic role sequence candidates of a given predicate by using the local classification approach, which is a widely used approach in previous research. The frameset generation model estimates the probability of each frameset that the predicate can take. The matching model is designed to measure the degree of the matching between the generated role sequence and the frameset by using several features. These features are developed to represent the predicate-argument structure information described in the frameset. In the experiments, our model shows that the use of knowledge about the predicate-argument structure is effective for selecting a more appropriate semantic role sequence.

key words: *semantic role labeling, predicate-argument structure, frame information, propbank, frameset*

1. Introduction

Semantic Role Labeling (SRL) is a task of identifying the shallow semantic structure of a natural language sentence. Most previous work ([1]–[6]) has used a local classification method which assigns a semantic role to each parse node by assuming that a role of a parse node is statistically independent to roles of other nodes.

Although the independence assumption has clear advantages in making an SRL model tractable and robust for parameter estimation, it can produce an invalid predicate-argument structure. For example, the sentence *"The cavalry broke."* including the predicate *break* is usually realized as `AGENT-break-PATIENT` or `PATIENT-break` form. In this example, the role of the parse node *the cavalry* is likely to be the PATIENT role, rather than the AGENT role, because there is no argument that can be assigned as the PATIENT, which is necessary for forming an `AGENT-break-PATIENT` joint structure. However, the local classification method cannot distinguish between the two different predicate-argument structures for the given sentence.

There have been two types of approaches to solving this problem. The first approach uses constraint rules to filter out invalid predicate-argument structures from candidate results ([4], [7]), and the second approach models a joint argument structure directly by relaxing the independence assumption between roles ([8], [9]). In the first approach, the argument candidates are generated by using local classifiers, and are then verified by using manually composed constraint rules. Although it has improved the SRL performance in many experiments, this approach has the limitation that it cannot give any preference for multiple candidates, if they satisfy the constraints.

The second approach integrates a likelihood of an argument-predicate structure in the SRL model. It automatically induces joint structures of predicate-arguments from an annotated corpus and estimates the likelihood of candidates from local classifiers. The experiments of [8] have shown that using the likelihood of the joint structure can bring additional improvements in the SRL performance.

Generally, these kinds of methods do not use explicit linguistic knowledge about a predicate-argument structure**. From the linguistic point of view, the number and the kind of semantic arguments of each predicate are determined. Clearly, such types of information can be useful features in measuring the validity of a predicate-argument structure and can be acquired from the existing lexicon resources that are used for constructing the semantic role annotated corpus, such as *frameset* of Propbank ([10]). One example of the frameset is given in Fig. 1. Here, we can easily notice that the predicate `break` can take four arguments; `A0, A1, A2,` and `A3`.

Based on this observation, we propose a new SRL approach that utilizes linguistic knowledge from the frame information to estimate the likelihood of a predicate-argument structure. Our approach clearly differs from the previous constraint-based approach. We use the linguistic knowledge to estimate the likelihood rather than to filter out. Our approach also differs from the second approach, that uses the likelihood of the joint structure, because our approach explicitly utilizes linguistic knowledge.

The remainder of this paper is organized as follows. Section 2 will briefly introduce the Propbank corpus, and

Manuscript received June 16, 2009.
Manuscript revised September 14, 2009.

†The authors are with the Dept. of Computer and Radio Communications Engineering, Korea Univ., Seoul, Korea.
††The author is with Div. of Computer Engineering, Sungkyul University, Anyang-City, Gyeonggi-do, Korea.

*This work was supported by Grant No. R01-2006-000-11162-0 (2008) from the Basic Research Program Korea Science and Engineering Foundation.

a) E-mail: rim@nlp.korea.ac.kr

DOI: 10.1587/transinf.E93.D.201

**In the rest of this paper, we will refer to this kind of knowledge as *frame information*.

```
<roleset id="break.01" name="break, cause to not
 be whole">
 <roles>
  <role descr="breaker" no="0">
  <role descr="thing broken" n="1">
  <role descr="instrument" n="2"/>
  <role descr="pieces" n="3"/>
 </roles>
 <example>
   <text>John broke the window with a rock</text>
   <arg n="0">John</arg>
   <rel>break</rel>
   <arg n="1">the window</arg>
   <arg f="with" n="2">a  rock</arg>
 </example>
</roleset>
```

Fig. 1 The example of the frameset “break.01” in PropBank.

the frameset used in this research. In Sect. 3 and 4, we will describe our proposed method and present the experimental results. Finally, we will conclude the paper and discuss future works in Sect. 5.

2. Propbank and Frameset

Propbank is one of the widely used semantic role annotated corpus. In Propbank, the semantic roles are divided into two folds: *core role* and *adjunct role*. The core roles are defined verb by verb, and are labeled as `A0, A1 ..., A5`. The adjunct roles can be taken by any verb, such as `AM-TMP(Temporal), AM-LOC(Location), AM-MNR(Manner)`, etc.

Propbank also provides a lexicon that specifies the semantic structures for verbs [10]. It consists of about 4,500 `frameset` describing a set of roles corresponding to a distinct usage of a verb. In essence, a frameset for a certain verb specifies its semantic role entries. In the case of a polysemous verb, there may be multiple framesets for each meaning of the verb. Based on the framesets, we can identify the number and type of core roles that the verb can take. The frameset also contains its annotation example sentences which include additional useful information, such as the functional word, which can be a head of an argument phrase. In the `<example>` of Fig. 1, we can see that the A2 argument (n=“2”) for the predicate *break* could be realized as a phrase whose head is the functional word *with*.

3. Proposed SRL Model

Formally, the goal of our SRL model is to find a valid sequence of roles $R^* = \{r_1, \cdots, r_n\}$ for a given predicate v and dependency parse tree D, where r_i is the semantic role of the ith word in D.

Our basic idea is to use the frame information from a frameset f as prior knowledge for judging whether a candidate role sequence R is appropriate or not. Because a predicate can have more than one frameset, our SRL model tries to find a pair of a valid role sequence and a frameset $< R, f >^*$ simultaneously:

Dependency relation
Family membership
Position
Current word lemma
Current word POS
Dependency path from predicate to word
Predicate Lemma
Predicate POS
POS pattern of predicate children
Dependency relation pattern of predicate children
POS pattern of predicate sibling
Dependency relation pattern of predicate sibling
POS+lemma of left and right word of current word

Fig. 2 Features for role sequence generation model.

$$\begin{aligned} < R, f >^* &= \underset{<R,f>}{\operatorname{argmax}} P(R, f|D, v) \\ &= \underset{<R,f>}{\operatorname{argmax}} P(R|D, v) \cdot P(f|R, D, v) \end{aligned} \tag{1}$$

Although this equation properly models our intuition, it has a serious problem; the probability $P(f|R, D, v)$ can require a large amount of parameter estimation for every possible v, so it may cause a serious data sparseness problem.

To avoid such a risk, we assume that the probability $P(f|R, D, v)$ can be reformulated as the probability that R is *matched* to a specification of f for a given D.

$$P(f|R, D, v) \approx P(f|v) \cdot P(m = 1|f, R, D) \tag{2}$$

where m is the indicator denoting *matching*, which becomes 1 when all roles assigned in R satisfy a specification of f. This equation can be viewed as the matching score of f and R for given D, weighted by the relative frequency of f for v. By using Eq. (2), our SRL model is modified as:

$$\begin{aligned} < R, f >^* = \underset{<R,f>}{\operatorname{argmax}} \{ & P(R|D, v) \cdot P(f|v) \\ & \cdot P(m = 1|f, R, D) \} \end{aligned} \tag{3}$$

We denote $P(R|D, v)$, $P(f|v)$, $P(m = 1|f, R, D)$ as the role sequence generation model, frameset generation model, and matching model, respectively. We will describe each model in the following subsections.

3.1 Role Sequence Generation Model

The role sequence generation model (RSGM), $P(R|D, v)$, estimates the probability for every possible role sequence candidate. We can rewrite the equation by assuming the independence between the roles:

$$P(R|D, v) = \prod_{w_i} P(r_i|D, v) \tag{4}$$

where r_i is the semantic role label assigned to the ith word w_i of D. Equation (4) is semantically equivalent to the general local classification model that have been used in the previous research ([1]–[4], [11]). To estimate this probability, we use the maximum entropy method with the features shown in Fig. 2, motivated from [12].

While a predicate usually takes less than five arguments, average number of words in a Propbank sentence is about 25. This means that a majority of the words in a sentence is labeled as *NONE* indicating that the word is not an argument for a given predicate. Because such a bias can hurt the performance of statistical classification methods, we adopt the pruning strategy proposed by [12] to filter out a word that is probably *NONE* before performing the probability estimation. The probability of a pruned word is set to 1.0 for *NONE*.

Based on the semantic role probability distribution of each word, produced by the maximum entropy model, we generate a set of role sequence candidates, by applying the Viterbi decoding algorithm to the roles of each word. In addition, we apply the constraints used in [7], and filter out the invalid role sequences. To reduce the amount of computation, we only consider the top k results from the RSGM as candidates. k is empirically set to 8 in our experiments.

3.2 Frameset Generation Model

The frameset generation model (FGM), $P(f|v)$, represents the probability that the predicate v takes the frameset f in the sentence. It is simply estimated by MLE from an annotated corpus:

$$P(f|v) = \frac{count(f, v)}{count(v)} \tag{5}$$

The value of this model can be regarded as a weight for the matching model.

3.3 Matching Model

The matching model (MM), $P(m = 1|f, R, D)$, measures how likely a role sequence candidate R and a frameset f are matched. For this, we use the following assumptions:

- As R contains more core roles defined in f, R can be matched to f more likely.
- As the matched core roles in R are closer to the predicate v, R can be matched more likely to f.
- If the functional word of r_j in R appears in the `example` of the frameset f with the same role (see Fig. 1), R can be matched more likely to f.

Note that all of these assumptions are about core roles in the frameset, although a frameset can also specify adjunct roles. This is because a predicate can take other additional adjuncts, such as a temporal role (`TMP`), which are not described in its frameset.

Based on the assumptions, we define the following features:

- **Number of Overlapped Roles**: the number of core roles overlapped in f and R
- **Overlap Ratio**: the number of roles common in R and f over the number of roles defined in f
- **Distance**: the number of words between the predicate v and w, labeled as a core role r in a sentence, which is also in f
- **Functional Word Overlap**: boolean feature indicating that w that has r is the functional word existing in an example of f with the same role

We use the maximum entropy model with these features to estimate the matching probability. Since no data for training exists, we generate positive examples and negative examples from the framesets and the annotated corpus as follows. Suppose that there is a predicate v for which multiple framesets $\{f_1, \cdots, f_m\}$ are defined. We extract a correct role sequence R_v from a Propbank sentence whose predicate is v. If v takes the frameset f_i in this sentence, then we generate several pair of $\{< R_v, f_1 >, \cdots, < R_v, f_m >\}$ where $< R_v, f_i >$ is the correct pair and the others are not. We use the correct pair as a positive example and the other pairs are used as negative examples.

4. Experiments

We evaluated our SRL model based on the WSJ section of the CoNLL2008 shared task data [13]. We used gold parses as the input, and focused only on the semantic role labeling task. We assume that correct predicate is given, but frameset is unknown. Although we consider verbal predicates only, we think that our approach can be applied to nominal predicates by making a modification of features for the RSGM. As a baseline, we used the top 1 result of our RSGM which corresponds to the traditional local classification-based SRL model. To make our model flexible, we modified our proposed model as a log-linear form with weight factors α, β, and γ for the RSGM, FGM, and MM respectively†. In addition to the precision, recall, and F1 measure, we also employed *Perfect Proposition F1 (PPF1)*. It is the proportion of predicates of which all semantic roles are correctly antated by the system [13].

Table 1 shows the performances of the baseline and the proposed model. Here, *ALL* and *CORE* denotes all arguments and core arguments, respectively. Our experimental results show that taking the frame information into consideration can give additional benefits in the SRL task. The proposed approach shows better performances than the baseline, especially in terms of precision and PPF1. Particularly, the relatively bigger amount of improvement at PPF1 shows that our model considers the validity of the entire structure.

The proposed method, however, is not effective in recall. Based on our analysis, this seems to be the case, because the preference of our model for a role sequence candidate contains a small number of core role arguments; most of the training instances contains only a few core arguments, so the trained matching model generally has a tendency to give a higher score to the candidate with a smaller number of arguments. Additionally, our method frequently made errors in the misclassification of an adjunct argument as a core

† α, β, and γ are empirically set to 0.4, 0.1, and 0.5 respectively.

Table 1 Performances of semantic role labeling.

	Baseline	Proposed
ALL Precision	88.02	89.16
ALL Recall	76.10	75.91
ALL F1	81.63	82.00
CORE Precision	88.66	90.39
CORE Recall	76.26	76.28
CORE F1	82.00	82.74
ALL PPF1	53.01	55.44

Table 2 Effect of using Frameset Generation Model (FGM) and Matching Model (MM). β and γ mean the weight factors of FGM and MM, respectively. α, the weight factor of the Role Sequence Generation Model (RSGM), is set to 0.4.

β	0.0	0.1	0.2	0.3	0.4	0.5	0.6
γ	0.6	0.5	0.4	0.3	0.2	0.1	0.0
PPF1	55.42	55.44	55.37	55.21	55.13	55.04	53.01

role argument when the argument is a functional word specified in an example of a frameset. The error seems to be a side-effect of the *functional word overlap* feature. Although the feature contributes to the precision improvement in our experiments, it is required to devise a better method for using the functional word matching information.

Furthermore, we conducted semantic role labeling with several combinations of weight factors in order to evaluate the contribution of the FGM and the MM. The performance of the RSGM is regarded as a baseline as shown in Table 1 and the FGM and the MM take roles of reranking role sequence candidates generated from the RSGM. In this experiment, we fix α to 0.4 and various values are assigned to β and γ. As shown in Table 2, the FGM and the MM improve the performance in terms of PPF1. It means that frameset information helps to select globally correct role sequence. The last column of Table 2 is same as the baseline in Table 1. Because FGM does not consider role sequences, it gives the same score to the every candidate. Compared with the MM, the effect of the FGM is not sufficient. It is because that current FGM only uses the relative frequency.

5. Conclusion

In this paper, we have presented a new approach for semantic role labeling that utilizes the linguistic knowledge on the frame information described in the frameset. Differing from the previous research, our approach estimates the likelihood of a predicate-argument structure on the basis of the explicit knowledge driven from the frameset, and uses this likelihood to find a legitimately structured answer in semantic role labeling.

For this purpose, we have devised a new statistical model that incorporates the matching probability between a candidate role sequence and a frameset of a predicate, and have proposed new features that can be useful. The experimental results have shown that the frame information can be helpful for semantic role labeling with the only a small number of simple features.

To achieve a better performance, it is necessary to discover new additional features for the matching model, and to find a way to solve the problems caused by the undesirable preference of our matching model and the functional word overlap features. Furthermore, investigating a more effective way to model the frame information will also be one of our future works.

References

[1] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," Computational Linguistics, vol.28, no.3, pp.245–288, 2002.
[2] S. Pradhan, W. Ward, K. Hacioglu, J.H. Martin, and D. Jurafsky, "Semantic role labeling using different syntactic views," Proc. ACL'05, 2005.
[3] N. Xue and M. Palmer, "Calibrating features for semantic role labeling," Proc. EMNLP 2004, 2004.
[4] M. Surdeanu, L. Marquez, X. Carreras, and P. Comas, "Combination strategies for semantic role labeling.," Journal of Artificial Intelligence Research (JAIR), vol.29, pp.105–151, 2007.
[5] X. Carreras and L. Màrques, "Introduction to the conll-2004 shared task: Semantic role labeling," Proc. CoNLL-2004, 2004.
[6] X. Carreras and L. Màrques, "Introduction to the conll-2005 shared task: Semantic role labeling," Proc. CoNLL-2005, 2005.
[7] V. Punyakanok, D. Roth, and W.t. Yih, "The importance of syntactic parsing and inference in semantic role labeling," Computational Linguistics, vol.34, no.2, pp.257–287, 2008.
[8] K. Toutanova, A. Haghighi, and C.D. Manning, "A global joint model for semantic role labeling," Computational Linguistics, vol.34, no.2, pp.161–191, 2008.
[9] R. Johansson and P. Nugues, "Dependency-based syntactic–semantic analysis with propbank and nombank," Proc. CoNLL-2008, 2008.
[10] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," Computational Linguistics, vol.31, no.1, pp.71–106, 2005.
[11] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky, "Shallow semantic parsing using support vector machines," Proc. HLT/NAACL'04, 2004.
[12] K. Hacioglu, "Semantic role labeling using dependency trees," Proc. CoNLL-2004, 2004.
[13] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre, "The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies," Proc. CoNLL-2008, 2008.