A Local Tree Alignment-based Soft Pattern Matching Approach for Information Extraction

Seokhwan Kim, Minwoo Jeong, and Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Science and Technology
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, Korea
{megaup, stardust, gblee}@postech.ac.kr

Abstract

This paper presents a new soft pattern matching method which aims to improve the recall with minimized precision loss in information extraction tasks. Our approach is based on a local tree alignment algorithm, and an effective strategy for controlling flexibility of the pattern matching will be presented. The experimental results show that the method can significantly improve the information extraction performance.

1 Introduction

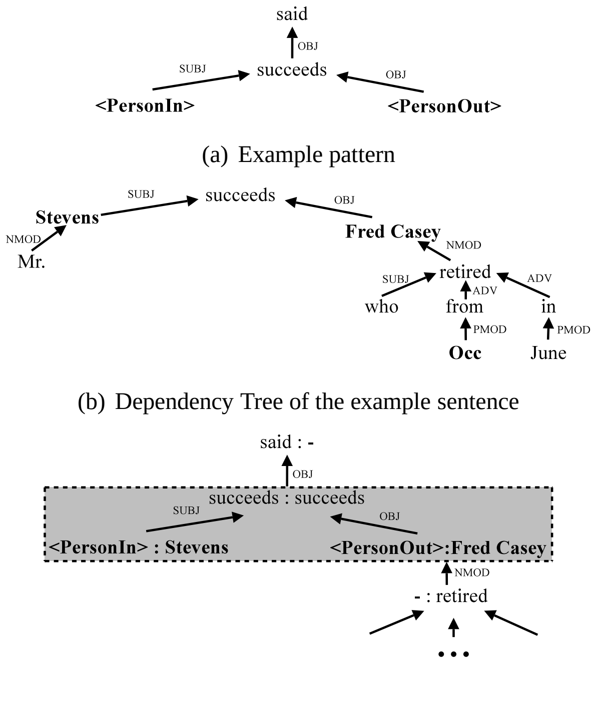
The goal of information extraction (IE) is to extract structured information from unstructured natural language documents. Pattern induction to generate extraction patterns from a number of training instances is one of the most widely applied approaches for IE.

A number of pattern induction approaches have recently been researched based on the dependency analysis (Yangarber, 2003) (Sudo et al., 2001) (Greenwood and Stevenson, 2006) (Sudo et al., 2003). The natural language texts in training instances are parsed by dependency analyzer and converted into dependency trees. Each subtree of a dependency tree is considered as a candidate of extraction patterns. An extraction pattern is generated by selecting the subtree which indicates the dependency relationships of each labeled slot value in the training instance and agrees on the selection criteria defined by each pattern representation model. A number of dependency tree-based pattern representation models have been proposed. The predicate-argument (SVO) model allows subtrees containing only a verb and its direct subject and object as extraction pattern candidates (Yangarber, 2003). The chain model represents extraction patterns as a chain-shaped path from each target slot value to the root node of the dependency tree (Sudo et al., 2001). A couple of chain model patterns sharing the same verb are linked to each other and construct a linked-chain model pattern (Greenwood and Stevenson, 2006). The subtree model considers all subtrees as pattern candidates (Sudo et al., 2003).

Regardless of the applied pattern representation model, the methods have concentrated on extracting only exactly equivalent subtrees of test instances to the extraction patterns, which we call hard pattern matching. While the hard pattern matching policy is helpful to improve the precision of the extracted results, it can cause the low recall problem. In order to tackle this problem, a number of soft pattern matching approaches which aim to improve recall with minimized precision loss have been applied to the linear vector pattern models by introducing a probabilistic model (Xiao et al., 2004) or a sequence alignment algorithm (Kim et al., 2008).

In this paper, we propose an alternative soft pattern matching method for IE based on a local tree alignment algorithm. While other soft pattern matching approaches have been able to handle the matching among linear vector instances with features from tree structures only, our method aims to directly solve the low recall problem of tree-to-tree pattern matching by introducing the local tree alignment algorithm which is widely used in bioinformatics to analyze RNA secondary structures. Moreover,

Proceedings of NAACL HLT 2009: Short Papers, pages 169–172,
Boulder, Colorado, June 2009.



(a) Example pattern

(b) Dependency Tree of the example sentence

(c) Local alignment-based tree pattern matching

Figure 1: An example of local alignment-based tree pattern matching

we present an effective policy for controlling degree of flexibility in the pattern matching by setting the optimal threshold values for each extracted pattern.

2 Methods

The low recall problem of information extraction based on hard pattern matching is caused by lack of flexibility in pattern matching. For example, the tree pattern in Figure 1(a) cannot be matched with the tree in Figure 1(b) by considering only exactly equivalent subtrees, because the first tree has an additional root node 'said' which is not in the second one. However, the matching between two trees can be performed by omitting just a node as shown in Figure 1(c).

In order to improve and control the degree of flexibility in tree pattern matching, we have adopted a local tree alignment approach as the pattern matching method instead of hard pattern matching strategy. The local tree alignment problem is to find the most similar subtree between two trees.

We have adopted the Hochsmann algorithm (Hochsmann et al., 2003) which is a local tree alignment algorithm used in bioinformatics to analyze RNA secondary structures. The goal of the Hochsmann algorithm is to find the local closed forest alignment which maximizes the similarity score for ordered trees. The algorithm can be implemented by a dynamic programming approach which solves a problem based on the previous results of its subproblems. The main problem of Hochsmann algorithm is to compute the similarity score between two subforests according to the defined order from the single node level to the entire tree level. The similarity score is defined based on three tree edit operations which are insertion, deletion, and replacement (Tai, 1979). For each pair of subforests, the maximum similarity score among three edit operations is computed, and the kind and the position of performed edit operations are recorded.

The adaptation of Hochsmann algorithm to the IE problem is performed by redefining the σ-function, the similarity score function between two nodes, as follows:

$$\sigma(v,w) = \begin{cases} 1 & \text{if lnk}(v)\text{=lnk}(w), \\ & \text{and lbl}(v)\text{=lbl}(w), \\ \sigma(\text{p}(w),\text{p}(v)) & \text{if lbl}(v)\text{=<SLOT>}, \\ 0 & \text{otherwise.} \end{cases}$$

where v and w are nodes to be compared, lnk(v) is the link label of v, lbl(v) is the node label of v, and $p(v)$ denotes a parent node of v. While general local tree alignment problems consider only node labels to compute the node-level similarities, our method considers not only node labels, but also link labels to the head node, because the class of link to the head node is important as the node label itself for dependency trees. Moreover, the method should consider the alignment of slot value nodes in the tree patterns for adopting information extraction tasks. If the pattern node v is a kind of slot value nodes, the similarity score between v and w is inherited from parents of both nodes.

After computing for all pairs of subforests, the optimal alignment is obtained by trace-back based on the recorded information of edit operation which maximizes the similarity score for each subforest pair. On the optimal alignment, the target node aligned to a slot value node on the pattern is regarded as an argument candidate of the extraction. Each extraction candidate has its confidence score which is computed from the alignment score, defined as:

$$\mathrm{score}(T_{\mathrm{PTN}}, T_{\mathrm{TGT}}) = \frac{S(T_{\mathrm{PTN}}, T_{\mathrm{TGT}})}{|T_{\mathrm{PTN}}|}$$

where $|T|$ denotes the total number of nodes in tree T and $S(T_1, T_2)$ is the similarity score of both trees computed by Hochsmann algorithm.

Only the extraction candidates with alignment score larger than the given threshold value, θ, are accepted and regarded as extraction results. For the simplest approach, the same threshold value, θ, can be applied to all the patterns. However, we assumed that each pattern has its own optimal threshold value as its own confidence score, which is different from other patterns' threshold values. The optimal threshold value θ_i and the confidence score $conf_i$ for the pattern P_i are defined as:

$$\theta_i = \operatorname*{arg\,max}_{0.5<\theta\leq 1.0} \{\mathrm{eval}_{\mathrm{fscore}}\,(D_{\mathrm{train}}, P_i, \theta)\}$$

$$conf_i = \max_{0.5<\theta\leq 1.0} \{\mathrm{eval}_{\mathrm{fscore}}\,(D_{\mathrm{train}}, P_i, \theta)\}$$

where $\mathrm{eval}_{\mathrm{fscore}}(D, P, \theta)$ is the evaluation result in F-score of the extraction for the data set D using the pattern P with the threshold value θ. For each pattern, the threshold value which maximizes the evaluation result in F-score for the training data set and the maximum evaluation result in F-score are assigned as the optimal threshold value and the confidence score for the pattern respectively.

3 Experiment

In order to evaluate the effectiveness of our method, we performed an experiment for the scenario template extraction task on the management succession domain in MUC-6. The task aims to extract scenario template instances which consist of person-in, person-out, position, organization slot values from news articles about management succession events. We used a modified version of the MUC-6 corpus including 599 training documents and 100 test documents described by Soderland (1999). While the scenario templates on the original MUC-6 corpus are labeled on each document, this version has scenario templates for each sentence.

All the sentences in both training and test documents were converted into dependency trees by Berkeley Parser[1] and LTH Constituent-to-Dependency Conversion Tool[2]. From the dependency trees and scenario templates on the training data, we constructed pattern candidate sets for four types of pattern representation models which are SVO, chain, linked-chain, and subtree models. For each pattern candidate, corresponding confidence score and optimal threshold value were computed.

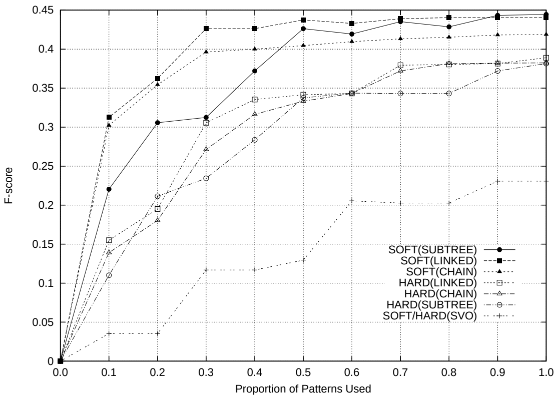


Figure 2: Comparison of soft pattern matching strategy with the hard pattern matching

The pattern candidates for each pattern representation model were arranged in descending order of confidence score. According to the arranged order, each pattern was matched with test documents and the extracted results were accumulated. Extracted templates for test documents are evaluated by comparing with the answer templates on the test corpus.

The curves in Figure 2 show the relative performance of the pattern matching strategies for each pattern representation model. The results suggest that soft pattern matching strategy with optimal threshold values requires less number of patterns for the performance saturation than the hard pattern matching strategy for all pattern models except the SVO model. For the SVO model, the result of soft pattern matching strategy is equivalent to that of hard pattern matching strategy. It is because most of patterns represented in SVO model are relatively shorter than those represented in other models.

In order to evaluate the flexibility controlling strategy, we compared the result of optimally determined threshold values with the cases of using

[1] http://nlp.cs.berkeley.edu/pages/Parsing.html

[2] http://nlp.cs.lth.se/pennconverter/

θ	SVO			Chain			Linked-Chain			Subtree		
	P	R	F	P	R	F	P	R	F	P	R	F
0.7	32.1	18.0	23.1	27.6	55.0	36.8	26.8	57.0	36.4	26.6	58.0	36.5
0.8	32.1	18.0	23.1	43.8	35.0	38.8	43.4	36.0	39.3	44.7	34.0	38.6
0.9	32.1	18.0	23.1	45.2	33.0	38.1	43.8	35.0	38.9	45.2	33.0	38.2
1.0 (hard)	32.1	18.0	23.1	45.2	33.0	38.1	43.8	35.0	38.9	45.2	33.0	38.2
optimal	32.1	18.0	23.1	36.0	49.0	**41.5**	40.7	48.0	**44.0**	43.0	46.0	**44.4**

Table 1: Experimental Results

various fixed threshold values. Table 1 represents the final results for all pattern representation models and threshold values. For the SVO model, all the results are equivalent regardless of the threshold strategy because of extremely short length of the patterns. For the other pattern models, precisions are increased and recalls are decreased by increasing the threshold. The maximum performances in F-score are achieved by our optimal threshold determining strategy for all pattern representation models. The experimental results of our method show the better recall than the cases of hard pattern matching and controlled precision than the cases of extremely soft pattern matching.

4 Conclusion

We presented a local tree alignment based soft pattern matching approach for information extraction. The softness of the pattern matching method is controlled by the threshold value of the alignment score. The optimal threshold values are determined by self-evaluation on the training data. Experimental results indicate that our soft pattern matching approach is helpful to improve the pattern coverage and our threshold learning strategy is effective to reduce the precision loss followed by the soft pattern matching method.

The goal of local tree alignment algorithm is to measure the structural similarity between two trees. It is similar to the kernel functions in the tree kernel method which is another widely applied approach to solve the IE problems. In the future, we plan to incorporate our alignment-based soft pattern matching method into the tree kernel method for IE.

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No. R01-2008-000-20651-0)

References

Mark A. Greenwood and Mark Stevenson. 2006. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of Workshop on Information Extraction Beyond The Document,* pp. 29–35.

Matthias Hochsmann, Thomas Toller, Robert Giegerich, and Stefan Kurtz. 2003. Local similarity in rna secondary structures. In *Proceedings of the IEEE Computer Society Bioinformatics Conference* , pp. 159–68.

Seokhwan Kim, Minwoo Jeong, and Gary Geunbae Lee. 2008. An alignment-based pattern representation model for information extraction. In *Proceedings of the ACM SIGIR '08,* pp. 875–876.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1):233–272.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2001. Automatic pattern acquisition for japanese information extraction. In *Proceedings of the first international conference on Human language technology research*, pp. 1–7.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the ACL '03,* pp. 224–231.

Kuo-Chung Tai. 1979. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433.

Jing Xiao, Tat-Seng Chua, and Hang Cui. 2004. Cascading use of soft and hard matching pattern rules for weakly supervised information extraction. In *Proceedings of COLING '04*, pp. 542–548.

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the ACL '03*, pp. 343–350.