Study of Kernel-based Methods for Chinese Relation Extraction

Ruihong Huang[1,2], Le Sun[1], Yuanyong Feng[1,2]

[1]Institute of Software, Chinese Academy of Sciences,
No.4, South Fourth Street, Zhongguancun, Haidian District, 100080, Beijing, China
[2]Graduate University of Chinese Academy of Sciences,
No.19, Yu Quan Street, Shijinshan, 100049, Beijing, China
{Ruihonghuang.china, lesunle}@gmail.com, yuanyong02@iscas.ac.cn

Abstract. In this paper, we mainly explore the effectiveness of two kernel-based methods, the convolution tree kernel and the shortest path dependency kernel, for Chinese relation extraction based on ACE 2007 corpus. For the convolution kernel, the performances of different parse tree spans involved in it for relation extraction are studied. Then, experiments with composite kernels, which are a combination of the convolution kernel and feature-based kernels are presented in order to discuss the complementary effects between tree kernel and flat kernels. For the shortest path dependency kernel, we improve it by replacing the strict same length requirement with finding the longest common subsequences between two shortest dependency paths. Experiments show kernel-based methods are effective for Chinese relation extraction.

Keywords: Chinese Relation Extraction, Convolution Tree Kernel, Shortest Path Dependency Kernel

1. Introduction

The aim of relation extraction as a subtask of information extraction is to find various predefined semantic relations between pairs of entities in text. The research of relation extraction has been advanced by the Message Understanding Conferences (MUC) [1] and the NIST Automatic Content Extraction (ACE) program (ACE, 2000-2007) [2] in Phase 2. As a subtask of Information Extraction, relation extraction can be utilized in many applications such as Question Answering and information retrieval.

To our knowledge, no work has been done to examine the performance of the tree kernel or the shortest path dependency kernel on Chinese corpus. Since more errors exist in Chinese syntax analysis compared with English, whether these kernel methods are applicable for Chinese relation extraction is uncertain.

2. Related Work

Since relation extraction task was first introduced in MUC6, many methods, such as feature-based [3, 4], tree kernel-based [11, 12, 13, 14] and composite kernel-based [5, 15, 16], have been proposed in literature.

Feature-based methods for relation extraction employ explicitly various linguistic features, from lexical features, syntactic information to dependency trees and semantic knowledge in Max Entropy model [3] or SVM model [4]. The feature-based methods have achieved the state-of-art performances. However, the feature selection is heuristic, so it needs much manual efforts, besides, it is difficult to improve since the features in nearly all linguistic levels have been examined [5, 6], furthermore, feature-based methods lack the ability to explore the structural syntactic or dependency information which should be quite important for relation extraction in our first sight.

In contrast, kernel-based methods [7, 8] has the potential for further performance improvements as it gives us an elegant way to explore structural features implicitly by computing the similarity between two objects via a kernel function, thus give us a good chance to explore the parsing or dependency information of the sentence where the entity pair occur. In recent years, different kernel types have been proposed for English relation extraction, from the hierarchical tree kernels [11] and [12] defined on shallow parse tree or dependency trees, shortest path dependency kernel [13] to the current convolution parse tree kernel[14,16]. Moreover, composite kernel which generally is a combination of a tree kernel and a feature-based kernel has advanced the performance further [15, 16]. Up to now, the kernel-based methods have achieved and recently exceeded the best performance of the feature-based methods for relation extraction.

For Chinese entity relation extraction, various features and different classification algorithms, for example, SVM [17] and bootstrapping [18], have been proposed in feature-based framework, and the reported results are usually alluring just on certain types of relations or on non-standard dataset. Besides, [19] proposed a novel improved Edit kernel for Chinese relation extraction, in which, the author improved the edit distance algorithms considering the Chinese word property and applied it to the Chinese relation extraction. However, as to structural kernels, which have been explored extensively recently for English, few work has been done up to now.

When we reflect over the two hierarchical kernels, the shortest path dependency kernel and the convolution parse tree kernel, we can see undoubtedly the convolution parse tree kernel has achieved better performance than the other kernel types and the shortest path dependency kernel is the most efficient since it is so fast and still achieves general performance. Therefore, in this paper, we explore different feature spaces involved in convolution kernel and improve the original shortest path dependency kernel with an aim to loosen its strict constraints of same lengths on shortest dependency paths.

3. Tree Spans in Convolution Kernel for Relation Extraction

A convolution kernel [7] aims to capture structural information in terms of substructures. As a special convolution kernel, the convolution tree kernel proposed in [9] counts the number of common sub-trees as the structural similarity between two parse trees. Furthermore, [10] has proved that the convolution tree kernel can be computed in O(|N1|*|N2|) where N1 and N2 equal to the number of nodes in two trees.

Same as most of previous work on kernel-based relation extraction, we first employ the parse tree segment within the entity pair (called Path-enclosed tree in [14]) in convolution kernel for entity relation extraction. Then, considering the limited parse tree spans within the nested entity pair, we extend the feature spaces for the nested relation instances to incorporate a verb factor in two strategies.

The first strategy (figure 1) is to extend the cases by including the highest level of verb (the main predicate) of the sub-sentence where the entity pair occurs when there are not any verb between the entity pair. The second strategy (figure 2) is to include the nearest verb to the entity pair (similar to the dynamic span expansion method proposed in [16]). The two strategies are out of our wonder that whether the main predicate which is powerful to determine the semantics of the whole sub-sentence or the nearest verb which is more relevant to the entity pair semantic relation will contribute more to the final entity relation identification.

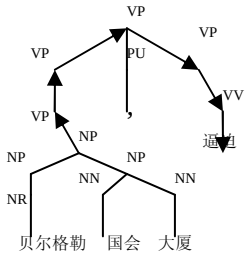

Fig. 1. strategy one

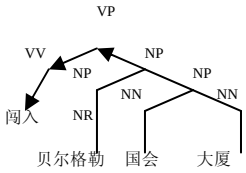

Fig. 2. strategy two

4. Shortest Path Kernel based on Longest Common Subsequence

The shortest path kernel proposed in [13] requires two shortest dependency paths to have the same length which may contribute to the low recall. To loosen the constraint, we improve the kernel by summing up the common word classes on the longest common subsequences of two shortest dependency paths other than the original shortest dependency paths.

In implementation (figure 3), the general part-of-speech features (the italic) of the nodes is used to match the longest common subsequences while the part-of-speech features, the word features and the entity type features of the nodes in the resulted longest common subsequences are utilized to compute the similarity. Same as in [13], no normalization is done in the improved shortest path dependency kernel.

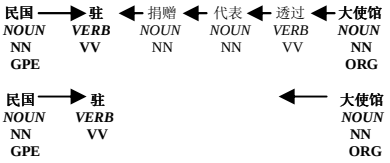

Fig. 3. two relations of type “PART-WHOLE” with the similarity equal to 4*3*4=48

Besides, in our experiments, the dependency information is generated by the same CFG (Context Free Grammar) parser employed to generate the parse trees, so the dependency links form a tree naturally. Thus, the shortest dependency path is somewhat different from the path used in [13], the meanings slightly differ accordingly. There is no unified direction distribution in the original shortest dependency path. In meaning, the shortest dependency paths of Bunescu and Mooney mainly describe the predicate-argument interactions, that is, the dependency relation of the two entities as the arguments to the predicates. In contrast, we always have a parent, pointed by dependency nodes on both sides, in our shortest dependency path. In meaning, our shortest dependency paths convey an ordered dependency series connecting the entity pair. Accordingly, the improved dependency path kernel will operate on the two longest common subsequences belonging to the two sides from the two ends to the central parent.

5. Experiments

5.1 Experimental Setting

Data: we use the Chinese portion of ACE (Automatic Content Extraction) 2007 corpora from LDC to conduct our experiments. In the ACE 2007 data, the training set include 689 documents and 6900 relation instances while the testing set include 160 documents and 1977 relation instances. Specifically, there are 2030 non-nested relation instances in the training set and 620 in the testing set. The ACE 2007 data defines 7 major entity types: Facility, GPE, Location, Organization, Person, Vehicle and Weapon. In this paper, we assume that the entities and their types are already known. Besides, all pairs of entities occurring in the same sentence are treated as potential relation instances. The data imbalance and sparseness are potential problems in ACE corpus, for example, the “Employment” subtype has 1265 positive instances while the “Artifact” subtype has only 6 instances in the training data, besides, in both the training part and the testing part, the negative samples are 10 times more than the positive samples.

Data processing: We select the Stanford syntactic parser to generate the sentence parse tree and dependency list. For we don’t find any appropriate POS tagger preserving the Penn standard required by the above parser, we use the POS tagger internal to the parser. Besides, we utilize the PKU Chinese word segmenter. During parsing, we segment sentences which are too long to be parsed at one time. For

shortest path dependency kernel, we construct the sentence dependency tree utilizing the output dependency list and extract the shortest path from it.

Classifier: we select the LibSVM [20] as our classifier and insert into the convolution tree kernel [21] and our shortest dependency path kernel based on the longest common subsequences. Besides, we adopted one vs. one strategy for the multi-class classification. The parameters are selected using 5-fold cross-validation.

Kernel normalization: the convolution tree kernel, the linear entity kernel and its expansion occurring in all experiments are all normalized while the two shortest path dependency kernels are not. The normalizing method is:

$$\hat{K}(T_1, T_2) = K(T_1, T_2) / \sqrt{K(T_1, T_1) \bullet K(T_2, T_2)}$$

Evaluation: we adopt Recall (R), Precision (P) and F-measure (F) standards.

5.2 Experimental Results

(1) Table 1 compares the performances of three different parse tree spans involved in the pure convolution tree kernel. We threshold the output to get best performances. We can see the Path-enclosed parse tree achieves best performance, although much lower than 74.12/54.90/63.07 (P/R/F) attained in our feature-based experiments. Especially, strategy one gets the lowest F-measures which generally will involve larger part of the parse tree in the convolution tree kernel. Thus, on the one hand, we may infer that more noises have been introduced into the kernel computation which may counteract the benefits of involving somewhat more complete syntactic structures in the kernels. On the other hand, we are confirmed that more efforts are needed to find appropriate and effective feature spaces.

(2) With the aim to examine the complementary effect of the tree features and the flat features, we also experimented with two composite kernels which are combinations of the above convolution kernel with the linear entity kernel (Comp-linear) and its polynomial expansion (Comp-poly) as stated in [15]:

$$K_c(R_1, R_2) = \alpha \cdot K_T(R_1, R_2) + (1 - \alpha) K_{Ent}.$$

In table 2, the F-measures show both composite kernels (with α set to 0.5 in Comp-linear and 0.7 in Comp-poly) advance the performances to a extent not as evident as that on English dataset [15]. Especially, unlike [15, 16], the polynomial entity kernel embodying bi-gram entity information doesn't improve any performance compared with the linear entity kernel, ruins the evaluations reversely.

Table 1. Three different feature spaces involved in the pure convolution tree kernel

Feature spaces	Avg. P(%)	Avg. R(%)	Avg. F
Path-enclosed tree	40.05	33.04	35.03
Strategy one	22.99	26.68	22.06
Strategy two	29.31	40.19	32.66

Table 2. Performance comparison of different kernel setups on the path-enclosed parse trees

Path-enclosed	Avg. P(%)	Avg. R(%)	Avg. F
Pure Conv	40.05	33.04	35.03
Comp-linear	41.67	34.75	36.73

Comp-poly	41.59	34.75	36.72

(3) Our original motivation to study kernel-based methods is to improve the extraction performance of the non-nested relations, they have longer distances and more complex syntactic structures between entity pairs, thus are more difficult to identify using flat features in feature-based methods. The initial results of kernel method on non-nested relations are presented in table 3. Besides, we give our previous experimental results using feature-based methods. The comparison shows that structural kernel's performance utilizing only the parse tree information has exceeded the feature-based methods on non-nested relation instances a bit.

Table 3. Performance comparison on non-nested relations of ACE 2007 data

Non-nested	Avg. P(%)	Avg. R(%)	Avg. F
Conv Kernel	40.09	33.22	34.13
Feature-based	54.81	19.15	29.57

Table 4. Classification performance between the original shortest path dependency kernel and the improved version based on the longest common subsequences over real relations

Shortest-path dep-kernel	Avg. P(%)	Avg. R(%)	Avg. F
Original	4.52	16.67	7.12
Improved	17.89	18.62	15.20

(4) The experimental results on the shortest dependency path kernel are somewhat disappointing. The improved kernel shows poor performance on the relation detection while the original shortest dependency path kernel has even not any relation detection ability since the trained model treats all the potential relations as positive or negative instances under different parameters. So, in table 4, we only show their multi-classifying performances on all the 1977 positive instances. We can see although our improved kernel has better performance, in general, the classification performances are both too low. When we examined the dependency path representations of relation instances, we find that a large part of Chinese relations lack clear predict-argument dependencies which is presumed in [13] and we may reason that it's really difficult to extract Chinese relations using this type of kernel.

6. Conclusion

In this paper, we explore the effectiveness of kernel-based methods for Chinese relation extraction. The experimental results show that although the current tree kernels haven't achieved as good performance as the feature-based methods, it has given reasonable measures, especially it has overrun feature-based methods on non-nested relations which is difficult to deal with by just using flat features.

7. Acknowledgment

This work is partially supported by National Natural Science Foundation of China with the contract No. 60773027 and No. 60736044 and by the 863 project of China with the contract No. 2006AA010108.

References

1. MUC. 1987-1998. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.
2. ACE. 2002-2007. http://projects.ldc.upenn.edu/ace/.
3. Kambhatla Nanda. Combining lexical, syntactic and emantic features with Maximum Entropy models for extracting relations. ACL-2004 (poster).
4. Zhou G.D., Su J., Zhang J., and Zhang M.. *Exploring Various Knowledge in Relation Extraction*. ACL-2005.
5. Zhao Shubin and Grishman Ralph. Extracting Relations with Integrated Information Using Kernel Methods. ACL-2005.
6. Jiang, J., & Zhai, C.. A systematic exploration of the feature space for relation extraction. NAACL-HLT'07.
7. Haussler D.. *Convolution Kernels on Discrete Structures.* Technical Report UCS-CRL-99-10, University of California, Santa Cruz. 1999.
8. Schölkopf B. and Smola A. J. *Learning with Kernels:*
SVM, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA 407-423. 2001.
9. Collins M. and Duffy N. Convolution Kernels for Natural Language. NIPS'2001.
10. Collins M. and Duffy N. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. ACL' 2002.
11. Zelenko D., Aone C. and Richardella A. *Kernel Methods for Relation Extraction*. Journal of Machine Learning Research. 2003(2):1083-1106.
12. Culotta A., Sorensen J. Dependency Tree Kernel for Relation Extraction. ACL-2004.
13. Bunescu R. C. and Mooney R. J. A Shortest Path Dependency Kernel for Relation Extraction. EMNLP-2005.
14. Zhang M, Zhang J, and Su J. 2006a. Exploring syntactic features for relation extraction using a convolution tree kernel. In Proceedings of HLT/NAACL.
15. Zhang M., Zhang J., Su J. and Zhou G.D. 2006b. A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features . COLINGACL.
16. Zhou G.D., Zhang M, Ji DH, Zhu Q.M. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. ACL-2007.
17. Che W.X.. *Automatic Entity Relation Extraction.* Journal of Chinese Information Processing, Vol. 19 No.2. 2004.
18. Zhang S.X.. *Study about automatic entity relation extraction.* Journal of Harbin Engineering University. Jul. 2006.
19. Che W.X. . Improved-Edit-Distance Kernel for Chinese Relation Extraction. IJCNLP-2005.
20. LibSVM. website: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
21. Moschitti. Convolution Tree kernel. http://ai-nlp.info.uniroma2.it/moschitti/TK1.2-software/Tree-Kernel.htm