Improved-Edit-Distance Kernel for Chinese Relation Extraction

Wanxiang CheJianmin Jiang, Zhong Su, Yue PanTing LiuSchool of Computer Sci. and Tech.IBM CRLSchool of Computer Sci. and Tech.Harbin Institute of TechnologyBeijing China, 100085Harbin Institute of TechnologyHarbin China, 150001{jiangjm, suzhong, liu@ir.hit.edu.cnHarbin China, 150001

Abstract

In this paper, a novel kernel-based method is presented for the problem of relation extraction between named entities from Chinese texts. The kernel is defined over the original Chinese string representations around particular entities. As a kernel function, the Improved-Edit-Distance (IED) is used to calculate the similarity between two Chinese strings. By employing the Voted Perceptron and Support Vector Machine (SVM) kernel machines with the IED kernel as the classifiers, we tested the method by extracting person-affiliation relation from Chinese texts. By comparing with traditional feature-based learning methods. we conclude that our method needs less manual efforts in feature transformation and achieves a better performance.

1 Introduction

Relation extraction (RE) is a basic and important problem in information extraction field. It extracts the relations among the named entities. Examples of relations are *person-affiliation*, *organization-location*, and so on. For example, in the Chinese sentence "郭士纳是IBM公司的主 席。" (Gerstner is the chairman of IBM Corporation.), the named entities are 郭士纳 (*people*) and IBM公司 (*organization*). The relation between them is *person-affiliation*. Usually, we can regard RE as a classification problem. All particular entity pairs are found from a text and then decided whether they are a relation which we need or not.

At the beginning, a number of manually engineered systems were developed for RE problem (Aone and Ramos-Santacruz, 2000). The automatic learning methods (Miller et al., 1998; Soderland, 1999) are not necessary to have someone on hand with detailed knowledge of how the RE system works, or how to write rules for it.

Usually, the machine learning method represents the NLP objects as feature vectors in the feature extraction step. The methods are named feature-based learning methods. But in many cases, data cannot be easily represented explicitly via feature vectors. For example, in most NLP problems, the feature-based representations produce inherently local representations of objects, for it is computationally infeasible to generate features involving long-range dependencies. On the other hand, finding the suitable features of a particular problem is a heuristic work. Their acquisition may waste a lot of time.

Different from the feature-based learning methods, the kernel-based methods do not need to extract the features from the original text, but retain the original representation of objects and use the objects in algorithms only via computing a kernel (similarity) function between a pair of objects. Then the kernel-based methods use existing learning algorithms with dual form, e.g. the Voted Perceptron (Freund and Schapire, 1998) or SVM (Cristianini and Shawe-Taylor, 2000), as kernel machine to do the classification task. Haussler (1999) and Watkins (1999) proposed a new kernel method based on discrete structures respectively. Lodhi et al. (2002) used string kernels to solve the text classification problem. Zelenko et al. (2003) used the kernel methods for extracting relations from text. They defined the kernel function over shallow parse representation of text. And the kernel method is used in conjunction with the SVM and the Voted Perceptron learning algorithms for the task of extracting *person-affiliation* and *organization-location* relations from text.

As mentioned above, the discrete structure kernel methods are more suitable to RE problems than the feature-based methods. But the stringbased kernel methods only consider the word forms without their semantics. Shallow parser based kernel methods need shallow parser systems. Because the performance of shallow parser systems is not high enough until now, especially for Chinese text, we cannot depend on it completely.

To cope with these problems, we propose the Improved-Edit-Distance (IED) algorithm to calculate the kernel (similarity) function. We consider the semantic similarity between two words in two strings and some structure information of strings.

The rest of the paper is organized as follows. In Section 2, we introduce the kernel-based machine learning algorithms and their application in natural language processing problems. In Section 3, we formalize the relation extraction problem as a machine learning problem. In Section 4, we give a novel kernel method, named the IED kernel method. Section 5 describes the experiments and results on a particular relation extraction problem. In Section 6, we discuss the reason why the IED based kernel method yields a better result than other methods. Finally, in Section 7, we give the conclusions and comments on the future work.

2 Kernel-based Machine Learning

Most machine learning methods represent an object as a feature vector. They are well-known feature-based learning methods.

Kernel methods (Cristianini and Shawe-Taylor, 2000) are an attractive alternative to feature-based methods. The kernel methods retain the original

representation of objects and use the object only via computing a *kernel function* between a pair of objects. As we know, a kernel function is a similarity function satisfying certain properties.

There are a number of learning algorithms that can operate using only the dot product of examples. We call them *kernel machines*. For instance, the Perceptron learning algorithm (Cristianini and Shawe-Taylor, 2000), Support Vector Machine (SVM) (Vapnik, 1998) and so on.

3 Relation Extraction Problem

We regard the RE problem as a classification learning problem. We only consider the relation between two entities in a sentence and no relations across sentences. For example, the sentence"布什总统接见了IBM公司主席郭士 纳。" (President Bush met Gerstner, the chairman of IBM Corporation.) contains three entities, 布什 (people), 郭士纳 (people) and IBM公 司 (organization). The three entities form two candidate *person-affiliation* relation pairs: 布什-IBM公司 and 郭士纳-IBM公司. The contexts of the entities pairs produce the examples for the binary classification problem. Then, from the context examples, a classifier can decide 郭士 纳-IBM公司 is a real person-affiliation relation but 布什-IBM公司 is not.

3.1 Feature-based Methods

The feature-based methods have to transform the context into features. Expert knowledge is required for deciding which elements or their combinations thereof are good features. Usually these features' values are binary (0 or 1).

The feature-based methods will cost lots of labor to find suitable features for a particular application field. Another problem is that we can either select only the local features with a small window or we will have to spend much more training and test time. At the same time, the feature-based methods will not use the combination of these features.

3.2 Kernel-based Methods

Different from the feature-based methods, kernelbased methods do not require much labor on extracting the suitable features. As explained in the introduction to Section 2, we retain the original string form of objects and consider the similarity function between two objects. For the problem of the *person-affiliation* relation extraction, the objects are the context around *people* and *organization* with a fixed window size w. It means that we get w words around each entity as the samples in the classification problem. Again considering the example "布什总统接见了IBM公司主席郭 士纳。", with w = 2, the object for the pair 郭 士纳 (*people*) and IBM公司 (*organization*) can be written as "接见了 **ORG** 主席 **PEO**。" Through the objects transformed from the original texts, we can calculate the similarity between any two objects by using the kernel (similarity) function.

For the Chinese relation extraction problem, we must consider the semantic similarity between words and the structure of strings while computing similarity. Therefore we must consider the kernel function which has a good similarity measure. The methods for computing the similarity between two strings are: the same-word based method (Nirenburg et al., 1993), the thesaurus based method (Qin et al., 2003), the Edit-Distance method (Ristad and Yianilos, 1998) and the statistical method (Chatterjee, 2001). We know that the same-word based method cannot solve the problem of synonyms. The thesaurus based method can overcome this difficulty but does not consider the structure of the text. Although the Edit-Distance method uses the structure of the text, it also has the same problem of the replacement of synonyms. As for the statistical method, it needs large corpora of similarity text and thus is difficult to use for realistic applications.

For the reasons described above, we propose a novel Improved-Edit-Distance (IED) method for calculating the similarity between two Chinese strings.

4 IED Kernel Method

Like normal kernel methods, the new IED kernel method includes two components: the kernel function and the kernel machine. We use the IED method to calculate the semantic similarity between two Chinese strings as the kernel function. As for the kernel machine, we tested the Voted Perceptron with dual form and SVM with a customized kernel. In the following subsections,



Figure 1: The comparison between the Edit-Distance and the Improved-Edit-Distance

we will introduce the kernel function, the IED method, and kernel machines.

4.1 Improved-Edit-Distance

Before the introduction to IED, we will give a brief review of the classical Edit-Distance method (Ristad and Yianilos, 1998).

The edit distance between two strings is defined as: The *minimum* number of edit operations necessary to transform one string into another. There are three edit operations, Insert, Delete, and Replace. For example, in Figure 1(a), the edit distance between "爱吃苹果(like apples)" and "喜 欢吃香蕉(like bananas)" is 4, as indicated by the four dotted lines.

As we see, the method of computing the edit distance between two Chinese strings cannot reflect the actual situation. First, the Edit-Distance method computes the similarity measured in Chinese character. But in Chinese, most of the characters have no concrete meanings, such as "苹", "果" and so on. The single character cannot express the meanings of words. Second, the cost of the Replace operation is different for different words. For example, the operation of "爱(love)" being replace by "喜欢(like)" should have a small cost, because they are synonyms. At last, if there are a few words being inserted into a string, the meaning of it should not be changed too much. Such as "爱吃苹果(like apples)" and "爱吃甜苹 果(like sweet apples)" are very similar.

Based on the above idea, we provide the IED method for computing the similarity between two Chinese strings. It means that we will use Chinese words as the basis of our measurement (instead of characters). By using a thesaurus, the similarity between two Chinese words can be computed. At the same time, the cost of the Insert operation is reduced.

Here, we use the CiLin (Mei et al., 1996) as

the thesaurus resource to compute the similarity between two Chinese words. In CiLin, the semantics of words are divided into High, Middle, and Low classes to describe a semantic system from general to special semantic. For example: "苹果(apple)" is Bh07, "香蕉(banana)" is Bh07, "西红柿(tomato)" is Bh06, and so on.

The semantic distance between word A and word B can be defined as:

$$Dist(\mathbf{A}, \mathbf{B}) = \min_{a \in \mathbf{A}, b \in \mathbf{B}} dist(a, b)$$

where **A** and **B** are the semantic sets of word A and word **B** respectively. The distance between semantic *a* and *b* is: dist(a,b) = 2 *(3 - d), where *d* means that the semantic code is different from the *d*th class. If the semantic code is same, then the semantic distance is 0. Therefore, Dist(苹果, 香蕉) = 0 and Dist(苹果, 西红柿) = 2.

Table 1 defines the variations of the edit distance on string "AB" after doing various edit operations. Where, "*" denotes one to four words, "A" and "B" are two words which user inputs. X' denotes the synonyms of X.

Table 1: The Variations of Edit-Distance with AB

Rank	Pattern		
1	AB		
2	A*B		
3	AB'; A'B		
4	A*B'; A'*B		
5	A'; B'		

According to Table 1, we can define the cost of various edit operations in IED. See Table 2, where " \rightarrow " denotes the delete operation.

Table 2: The Cost of Edit Operation in IED

Edit Operation	Cost	
A→A	0	
Insert	0.1	
A→A'	Dist(A, A')/10 + 0.5	
Others	1	

By the redefinition of the cost of edit operations, the computation of IED between "爱吃苹 果" and "喜欢吃香蕉" is as shown Figure 1(b), where the Replace cost of "爱"→"喜欢" is 0.5 and "苹果"→"香蕉" is 0.7. Thus the cost of IED is 1.2. Compared with the cost of classical Edit-Distance, the cost of IED is much more appropriate in the actual situation.

We use dynamic programming to compute the IED similar with the computing of edit distance.

In order to compute the similarity between two strings, we should convert the distance value into a similarity. Empirically, the maximal similarity is set to be 10. The similarity is 10 minus the improved edit distance of two Chinese strings.

4.2 Kernel Machines

We use the Voted Perceptron and SVM algorithms as the kernel machines here.

The Voted Perceptron algorithm was described in (Freund and Schapire, 1998). We used SVM^{light} (Joachims, 1998) with custom kernel as the implementation of the SVM method. In our experiments, we just replaced the custom kernel with the IED kernel function.

5 Experiments and Results

In this section, we show how to extract the *person-affiliation* relation from text and give some experimental results. It is relatively straightforward to extend the IED kernel method to other RE problems.

The corpus for our experiments comes from Bejing Youth Daily¹. We annotated about 500 news with named entities of **PEO** and **ORG**. We selected 4,200 sentences (examples) with both **PEO** and **ORG** pairs as described in Section 3. There are about 1,200 positive examples and 3,000 negative examples. We took about 2,500 random examples as training data and the rest of about 1,700 examples as test data.

5.1 Infection of Window Size in Kernel Methods

The change of the performance of the IED kernel method varying while the window size w is shown in Table 3. Here the Voted Perceptron is used as the kernel machine.

Our experimental results show that the IED kernel method got the best performance with the highest F-Score when the window size w =

¹http://www.bjyouth.com/

2. As *w* grows, the *Precision* becomes higher. With smaller *w*'s, the *Recall* becomes higher.

5.2 Comparison between Feature and Kernel Methods

For the feature-based methods implementation, we use the words which are around the **PEO** and the **ORG** entities and their POS. The window size is w (See Section 3). All examples can be transformed into feature vectors. We used the regularized winnow learning algorithm (Zhang, 2001) to train on the training data and predict the test data. From the experimental results, we find that when w = 2, the performance of feature-based method is highest.

The comparison of the performance between the feature-based and the kernel-base methods is shown in Table 4.

Figure 2 displays the change of F-Score for different methods varying with the training data size.



Figure 2: The learning curve (of F-Score) for the person-affiliation relation, comparing IED kernel with feature-based algorithms

Table 3: The Performance Effected by w

w	Precision	Recall	F-Score			
1	66.67%	92.68%	77.55%			
2	93.55%	87.80%	90.85%			
3	94.23%	74.36%	83.12%			

Table 4: The Performance Comparison

	Precision	Recall	F- $Score$
Regularized Winnow	75.90%	96.92%	85.14%
Voted Perceptron	93.55%	87.80%	90.85%
SVM	94.15%	88.38%	91.17%

From Table 4 and Figure 2, we can see that the IED kernel methods perform better for the *person-affiliation* relation extraction problem than for the feature-based methods.

Figure 2 shows that the Voted Perceptron method gets close to, but not as good as, the performance to the SVM method on the RE problem. But when using the method, we can save significantly on computation time and programming effort.

6 Discussion

Our experimental results show that the kernelbased and the feature-based methods can get the best performance with the highest F-Score when the window size w = 2. This shows that for relation extraction problem, the two words around entities are the most significant ones. On the other hand, with w becoming bigger, the *Precision* becomes higher. And with w becoming smaller, the *Recall* becomes higher.

From Table 4 and Figure 2, we can see that the IED kernel methods perform very well for the *person-affiliation* relation extraction. Furthermore, it does not need an expensive feature selection stage like feature-based methods. Because the IED kernel method uses the semantic similarity between words, it can get a better extension. We can conclude that the IED kernel method requires much fewer examples than feature-based methods for achieving the same performance.

For example, there is a test sentence "胡锦 涛 主席 接见 了 IBM公司 总裁" (Chairman Hu Jintao met the CEO of IBM Corporation). The feature-based method judges the 胡锦涛-IBM公 司 as a *person-affiliation* relation, because the context around 胡锦涛 and IBM公司 is similar with the context of the *person-affiliation* relation. However, the IED kernel method does the correct judgment based on the structure information. For this case the IED kernel method gets a higher precision. At the same time, because the IED kernel method considers the extension of synonyms, its recall does not decrease very much.

The speed is a practical problem in applying kernel-based methods. Kernel-based classifiers are relatively slow compared to featurebased classifiers. The main reason is that the computing of kernel (similarity) function takes much time. Therefore, it becomes a key problem to improve the efficiency of the computing of the kernel function.

7 Conclusions

We presented a new approach for using kernelbased machine learning methods for extracting relations between named entities from Chinese text sources. We define kernels over the original representations of Chinese strings around the particular entities and use the IED method for computing the kernel function. The kernel-based methods need not transform the original expression of objects into feature vectors, so the methods need less manual efforts than the feature-based methods. We applied the Voted Perceptron and the SVM learning method with custom kernels to extract the *person-affiliation* relations. The method can be extended to extract other relations between entities, such as organization-location, etc. We also compared the performance of kernel-based methods with that of feature-based methods, and the experimental results show that kernel-based methods are better than feature-based methods.

Acknowledgements

This research has been supported by National Natural Science Foundation of China via grant 60435020 and IBM-HIT 2005 joint project.

References

- Chinatsu Aone and Mila Ramos-Santacruz. 2000. Rees: A large-scale relation and event extraction system. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 76–83.
- Niladri Chatterjee. 2001. A statistical approach for similarity measurement between sentences for EBMT. In *Proceedings of Symposium on Translation Support Systems STRANS-2001*, Indian Institute of Technology, Kanpur.
- N. Cristianini and J. Shawe-Taylor. 2000. An Introduction to Support Vector Machines. Cambridge University Press, Cambridge University.
- Yoav Freund and Robert E. Schapire. 1998. Large margin classification using the perceptron algorithm. In *Computational Learning Theory*, pages 209–217.

- David Haussler. 1999. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, 7,.
- Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98*, number 1398, pages 137–142, Chemnitz, DE.
- Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444.
- Jiaju Mei, Yiming Lan, Yunqi Gao, and Hongxiang Yin. 1996. *Chinese Thesaurus Tongyici Cilin (2nd Editon)*. Shanghai Thesaurus Press, Shanghai.
- Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and the Annotation Group. 1998. Algorithms that learn to extract information– BBN: Description of the SIFT system as used for MUC. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.
- S. Nirenburg, C. Domashnev, and D.J. Grannes. 1993. Two approaches to matching in example-based machine translation. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 47– 57, Kyoto, Japan.
- Bing Qin, Ting Liu, Yang Wang, Shifu Zheng, and Sheng Li. 2003. Chinese question answering system based on frequently asked questions. *Journal of Harbin Institute of Technology*, 10(35):1179– 1182.
- Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522– 532.
- Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272.
- Vladimir N. Vapnik. 1998. Statistical Learning Theory. Wiley.
- Chris Watkins. 1999. Dynamic alignment kernels. Technical Report CSD-TR-98-11, 1,.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. J. Mach. Learn. Res., 3:1083–1106.
- Tong Zhang. 2001. Regularized winnow methods. In Advances in Neural Information Processing Systems 13, pages 703–709.