

# Embedding the Semantic Knowledge in Convolution Kernels

Kebin Liu<sup>1</sup> Fang Li<sup>2</sup> Ying Han<sup>3</sup> Lei Liu<sup>4</sup>

*Dept. of Computer Science & Engineering,*

*Shanghai Jiaotong University*

*{<sup>1</sup>captainlkb2003, <sup>2</sup>fli, <sup>3</sup>hanying, <sup>4</sup>liu-lei}@sjtu.edu.cn*

## Abstract

*Convolution kernels, such as tree kernel and subsequence kernel are useful for natural language processing tasks. However, most of them ignore the semantic knowledge. In order to solve the problem, this paper proposes a new method to embed the semantic knowledge into kernel calculation. The new method has been applied to extract the ORG-affiliation relation from Chinese texts and achieves an average F-measure of 82.1%. Comparing with feature-based method and the traditional Word-sequence kernel, it provides significant improvement.*

## 1. Introduction

Over the past few years, kernel methods have been successfully applied on natural language processing tasks together with several different learning algorithms and achieved state-of-the-art performance. An intuitive idea of the kernel method is to find a mapping  $\Phi$  from original space to a new space, in which problems can be easily solved. The kernel calculates the similarity between two objects, defined as the dot-product in the new vector space (e.g.  $K(x, y) = \Phi(x) \cdot \Phi(y)$ ).

Natural language texts have the form of sequences of words. After preprocessing on the texts, they are discrete structures, such as parsing trees [1,2], sequences [3,4] or graphs [5]. In these cases, convolution kernels [6], which run on these discrete structures, show excellent performances. Convolution kernels do not construct the feature vectors explicitly. They keep the initial form of input objects. However, convolution kernels implicitly map the original space to the new space indexed by all substructures. During computation, they enumerate all substructures occurring in both inputs and then count the common ones. For example, the tree kernels find common

fragments of the shallow parsing trees, and the Word-sequence kernel [3,7] considers common word subsequences. Recently convolution kernels have been combined with syntax information [8] from various levels like tokenization, shallow parsing etc. However, few of them employ the semantic knowledge. To solve this problem, this paper presents an improved kernel function in which the semantic similarity between Chinese words is embedded.

The rest of this paper is organized as follows: section 2 discusses obtaining semantic knowledge and our kernel function. Section 3 shows relation extraction experiments and results. Finally, the conclusion and future work are presented.

## 2. Our method

Many kinds of convolution kernels have made state-of-the-art performances. According to the data structure they use, convolution kernels could be divided into three classes: tree kernel [1,2], sequence kernel [3,4] and graph kernel [5]. In the reminder sections, we restrict the discussion to sequence kernels and our improvements. We introduce our model in this section including semantic information acquisition and the improved kernel function.

### 2.1. Obtaining semantic knowledge from ontology

The semantic knowledge this paper introduced is the similarity between Chinese words. In the new kernel function, matched subsequences are not the word sequences but the POS sequences. The similarity of the corresponding word sequences are then measured by the semantic similarity between their words and embedded in to the kernel calculation.

Many prior approaches use the statistic methods to calculate the semantic information such as GVSM [3], LSI [9] and PCA [10] which depend much on the

training set. Without loss of generality, this paper derives semantic knowledge from hownet [11] which is an authoritative ontology for Chinese. Each word in hownet has several concepts [12]. The similarity between two words is defined as the maximum similarity of their concepts:

$$SIM(W_1, W_2) = \max_{i=1,2,\dots,n; j=1,2,\dots,m} sim(C_{1i}, C_{2j}) \quad (1)$$

Concept similarity is evaluated by two different measures, concept similarity [12] and concept relevance.

First, in hownet each concept is represented by several primitive expressions separated with commas. There are 4 parts of primitive expressions constructed by primitives with or without relation descriptors. The similarity of two concepts can be calculated by combining the similarities of their primitive expressions:

$$sim'(C_1, C_2) = \sum_{i=1}^4 \beta_i sim_i(C_1, C_2) \quad (2)$$

Where  $sim_i(C_1, C_2)$  is the similarity measure of the  $i$ th kind of primitive expression [12] and  $\beta_i$  is the parameter indicates the primitive expression's weight.

$$sim_p(p_1, p_2) = \frac{\alpha}{\alpha + dis(p_1, p_2)} \quad (3)$$

The similarity between primitive expressions is measured by their primitives using the equation (3). Where  $dis(p_1, p_2)$  means the semantic distance between two primitives. The semantic distance is defined as the path length of the two primitives in semantic tree. The semantic tree is constructed according to the hyponymy between primitives. Here  $\alpha$  is a parameter whose value is the distance between two primitives when they share a similarity of 0.5.

Second, another measure is the overlay of the two concepts' relevant set. Relevant set of a concept contains the concepts that share one or more common primitive expressions with it. The relevance between two concepts can be measured with the following equation:

$$sim''(C_1, C_2) = \frac{|relevant(C_1) \cap relevant(C_2)|}{|relevant(C_1) \cup relevant(C_2)|} \quad (4)$$

The integrated formula of the semantic similarity of two concepts is:

$$sim(C_1, C_2) = \lambda_1 \sum_{i=1}^4 \beta_i sim_i(C_1, C_2) + \lambda_2 sim''(C_1, C_2) \quad (5)$$

$\lambda_1, \lambda_2$  are weight parameters.

## 2.2. Embedding the semantic knowledge in sequence kernel

Word-sequence kernel processes gapped word sequences to yield the kernel value. Soft matching [3] is a prior attempt on add semantic information to convolution kernels. However this will significantly increase the computational cost. We introduce the POS sequence into kernel calculation and match the POS sequences instead of word sequences. The semantic similarity of corresponding word sequence is then embedded.

First of all, definitions needed are as follows: A two tuple  $X_i = (p, w)$  is used to store the paired information of a symbol in which  $p$  denotes a POS and  $w$  denotes the word of  $X_i$ . Let  $X = X_1 X_2 \dots X_{|X|}$ ,  $Y = Y_1 Y_2 \dots Y_{|Y|}$  denote the input sequences.

Input sequences include POS and words. They can be considered as two relative sequences, word sequence and POS sequence. The basic idea of this approach can be described as:

$$K_n(X, Y) = \sum_{u \in \Sigma^n} \sum_{i: u=X[i].p} \sum_{j: u=Y[j].p} \lambda_m^{2n} \prod_{k=1}^n SIM(X_{i_k}.w, Y_{j_k}.w) \prod_{i_1 < l < i_n, l \notin i} \lambda_g \prod_{j_1 < h < j_n, h \notin j} \lambda_g \quad (6)$$

To reduce the time complexity, we apply the recursive implementation:

$$K_n(Xa, Y) = K_n(X, Y) + \sum_{j: Y_j.p=a.p} \lambda_m^2 K'_{n-1}(X, Y[1: j-1]) SIM(a.w, Y_j.w) \quad (7)$$

Where  $\lambda_m$  is the decay factor of matches and the function  $SIM(W_1, W_2)$  is used to calculate the semantic similarity between two Chinese words.  $SIM(W_1, W_2)$  is described in section 2.1.

$$K'_i(Xa, T) = \lambda_g K'_i(X, Y) + K''_i(Xa, Y) \quad (8)$$

The  $\lambda_g$  is the distinct decay factor for gaps.

$$K''_i(Xa, Yb) = \lambda_g K''_i(Xa, Y) + \lambda_m^2 K'_{i-1}(X, Y) SIM(a.w, b.w) \delta(a.p, b.p) \quad (9)$$

Here SIM ( $W_1, W_2$ ) is also used to embed the semantic knowledge. Function  $\delta$  is used to determine if two inputs are equal to each other. Here inputs of  $\delta$  are symbols' POS. Function  $\delta$  returns 1 while inputs are equal and returns 0 otherwise.

$$K_n(X, Y) = 0, \text{ if } \min(|X|, |Y|) < n \quad (10)$$

$$K'_i(X, Y) = 0, \text{ if } \min(|X|, |Y|) < i, (i = 1, \dots, n-1) \quad (11)$$

$$K''_i(X, Y) = 0, \text{ if } \min(|X|, |Y|) < i, (i = 1, \dots, n-1) \quad (12)$$

$$K'_0(X, Y) = 1 \quad (13)$$

In some special cases such as any input sequence's size is lower than  $n$ , the kernel values are specified in equation (10) to (13).

### 3. Experiments

In this section, two experiments which extract the ORG-Affiliation relations from Chinese texts are carried out to evaluate the new kernel function in relation extraction tasks, comparing with the feature-based approach and prior subsequence kernels. Documents collected from the web are used to generate relation candidates automatically. From about 86k words, 6086 candidates are generated in which 1852 are positive.

Experiment 1 is raised to see if the new semantic kernel has a better ability to distinguish different relation examples. In this experiment, kernels are used to calculate the similarity between two relation examples. Similarities between two positive examples and between a positive example and a negative one are evaluated. For each condition, we calculate 200 pairs and the average kernel values are listed in Table 1. Here "Feature-based" means the Feature-based method which uses the dot product to compute similarity between objects and "Word-sequence kernel" denotes the traditional kernel function. "Semantic kernel" means the new sequence kernel with semantic knowledge proposed in this paper. The expression "P VS P" in Table 2 denotes that the inputs of kernel functions are two positive relation examples and the expression "P VS N" means a positive example and a negative example.

Table 1. Similarity between relation examples

	P VS P	P VS N

	(average kernel value)	(average kernel value)
Feature-based	0.61	0.26
Word-sequence kernel	0.67	0.27
Semantic kernel	0.82	0.31

Comparing with two other methods, the new kernel function's average value between positive examples increases a little while the kernel value between two positive examples increases much. Results show that the new kernel is superior to traditional ones on distinguishing examples.

Experiment 2 uses relation examples to test three approaches in real relation extraction task.

Table 2. Relation extraction performance

	Precision	Recall	F-measure
Feature-based	86.0%	73.6%	79.3%
Word-sequence kernel	86.7%	<u>76.7%</u>	81.4%
Semantic kernel	<u>90.2%</u>	75.3%	<u>82.1%</u>

The results are shown in Table 2 from which it could be concluded that new sequence kernel with semantic knowledge achieves better performance than other prior approaches. Note that in this experiment, the training corpus is relatively large and providing enough examples for training.

Experiment 3 tests the generalization ability of the three methods. In fact, there are a series of experiments with different training set in experiment 2. The 5 separate experiments respectively use 100%, 80%, 60%, 40%, 20%, 10% amount of training examples which are chosen randomly from the initial training set. Testing set are the same in all experiments. Figure 1 shows the results in which the x-coordinate indicates the percentage of training examples used in the experiment and the y-coordinate indicates the F-measure of each approaches.

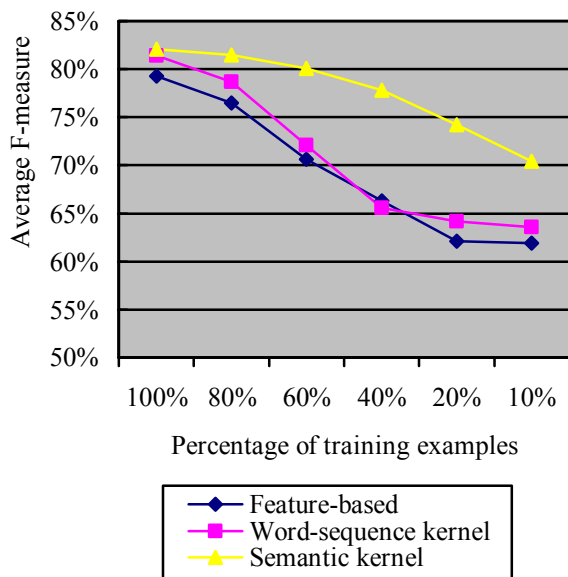


Figure 1. Relation extraction with different training sets

Results in Figure 1 show that, during the reduction of the training set, performance of the new kernel function descends slower than two other methods. Even with about 20% of the initial training examples, the new semantic kernel is also efficient. The feature-based approach and traditional sequence kernel have a rapid decrease on F-measure during the training set reducing from 80% to 40%.

## Conclusion

In this paper, we propose a method to embed the semantic similarity of Chinese words in to convolution kernel calculation. In order to achieve this, POS sequence and word sequence are both used. First, word sequence match in traditional Word-sequence kernel is replaced by POS sequence match. Then semantic similarity of corresponding word sequence for each matched POS sequence is embedded in the kernel calculation. Gaps are handled differently to the matches. Experiments show that, the new kernel has better generalization ability than other methods and achieve superior performances especially on small training sets. However, more other useful syntax and semantic information could be added into our work in the future. What's more, we would try to find more efficient methods to speed up the kernel calculation.

## References

- [1] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction", *J.Mach. Learn Res*, 2003, 3:1083-1106.
- [2] M. Collins, N. Duffy, "Convolution kernels for natural language", *Proc. of NIPS-2001*, 2001.
- [3] Nicola Cancedda, Eric Gaussier, Cyril Goutte, Jean-Michel Renders, "Word-Sequence Kernels", *Journal of Machine Learning Research*, 2003, 3:1059-1082.
- [4] Blaz Fortuna, "String Kernels", *SIKDD 2004 at multiconference IS 2004*, 2004.
- [5] J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda, "Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data", *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, 2003.
- [6] D. Haussler, "Convolution kernels on discrete structures", *Technical Report UCSC2CRL-99-10*, 1999.
- [7] J. Suzuki, H. Isozaki, and E. Maeda, "Convolution Kernels with Feature Selection for Natural Language Processing Tasks", 2003.
- [8] Shubin Zhao, R. Grishman, "Extracting Relations with Integrated Information Using Kernel Methods", *ACL 2005*, 2005.
- [9] N. Cristianini, H. Lodhi, J. Shawe-Taylor, "Latent Semantic Kernels", *Journal of Intelligent Information Systems*, 2002, 18:2-3.
- [10] B. Schölkopf, A. Smola, K-R. Müller, "Kernel Principal Component Analysis", *MIT Press*, 1999, 327-352.
- [11] Zhendong Dong, Qiang Dong, *About hownet*, <http://www.keenage.com>
- [12] Qun Liu, Sujian Li, "Word semantic similarity calculation based on Hownet", 2002.
- [13] H. Zhuge, "China's E-Science Knowledge Grid Environment", *IEEE Intelligent Systems*, 19 (1) (2004) 13-17.