WUST SVM-Based System at NTCIR-9 RITE Task

Maofu Liu, Yan Li, Yu Xiao, Chunwei Lei
College of Computer Science and Technology, Wuhan University of Science and Technology
Wuhan 430065, Hubei, P.R.China
e_mfliu@163.com, liyan880923@sina.com

ABSTRACT
This paper describes our work in NTCIR-9 on RITE Binary-class (BC) subtask and Multi-class (MC) subtask in Simplified Chinese. We use classification method and SVM classifier to identify the textual entailment. We totally use thirteen statistical features as the classification features in our system. The system includes three parts: (1) Preprocessing, (2) Feature Extraction, (3) SVM Classifier. In these three parts, we mainly focus on the second one.

Categories and Subject Descriptors
I.2.7 [Artificial Intelligence]: Natural Language Processing - text analysis.

I.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing - linguistic processing.

General Terms
Experimentation.

Keywords
Textual Entailment, SVM Classifier, Classification Features

1. INTRODUCTION
RITE is a generic benchmark task that addresses major text understanding needed in various NLP/Information Access research areas. In NTCIR-9, RITE task has four subtasks, we focus on two of them, Binary-Class (BC) subtask and Multi-Class (MC) subtask.

BC subtask means to identify whether text t1 entails or infers hypothesis t2 or not given a text pair (t1, t2). For example, Text1 entailing Text2 means that Text1 has the same meaning with Text2 and Text1 also has more meaning than Text2.

Text1:大量使用类固醇对某些人容易引起高血压，特别是在原本就有肾脏疾病者，长期服用会降低免疫力。

Text2: 大量使用类固醇者，可能免疫功能会变差。

MC subtask means that 5-way labeling task to detect (forward/reverse/bidirection) entailment or non-entailment (contradiction/independence) in a text pair. Forward means that t1 entails t2 and t2 does not entail t1 given a text pair (t1, t2). For example, the relation between Text3 and Text4 is forward. Reverse means that t2 entails t1 and t1 does not entail t2 given a text pair (t1, t2). For example, the relation between Text5 and Text6 is reverse. If it is the case that t1 entails t2 and t2 entails t1, then t1 and t2 are true in exactly the same situations, and are thus equivalent. In other words, equivalence is the bidirectional entailment and we also call it bidirection[3]. The relation between Text7 and Text8 is bidirection.

Text3: 唐卡具有通史性、趣味性、知识性、宗教性、工艺性等特点，故被人们誉为藏族的“百科全书”。

Text4: 唐卡被称为藏族的百科全书。

Text5: 糖尿病与胆固醇有关。

Text6: 看上去各不相干的糖尿病和胆固醇之间其实有密不可分的关系。

Text7: 海湾战争开打，伊拉克和科威特的原油停止出口，油价上涨。

Text8: 海湾战争开始,因科威特及伊拉克原油停止输出，造成油价上涨。

In MC subtask, non-entailment contains contradiction and independence. For instance, the relation between Text9 and Text10 is contradiction. The contradiction means that t1 and t2 contradict or cannot be true at the same time given a text pair (t1, t2). The independence means that if the pair (t1, t2) can not be put into any of 4-way (forward/reverse/bidirection/contradiction), we put it into the independence class.

Text9: 何大一被称为“爱滋病之父”。

Text10: 何大一作为第四个发现爱滋病的科学家，仅比首位发现者晚了 3 个月，却令他错失“爱滋病之父”的称号。

In this paper, we choose classification method and SVM classifier to solve the entailment problem. BC subtask can be regarded as two types of classification problem, entailment and non-entailment. MC subtask can be looked on as five types of classification problem, i.e. forward, reverse, bidirection, contradiction and independence.

2. System Description
Our system includes three main modules, i.e. preprocessing, feature extraction and SVM Classifier. Figure 1 illustrates our system architecture in detail.

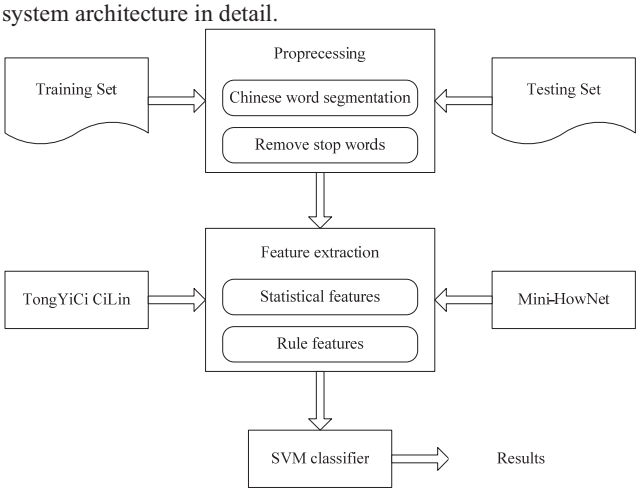

Figure 1. System Architecture

The following Figure 2 describes the SVM classifier in our system.

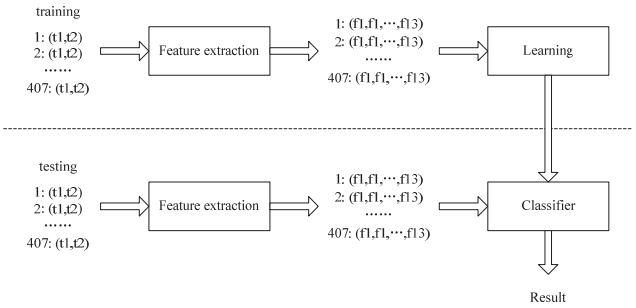

Figure 2. SVM Classifier

2.1 Preprocessing

We choose ICTCLAS[1] as the tool to segment the Chinese word. The following example is the preprocessing result of the original training data given by NTCIR-9 after Chinese word segmentation and removing stop words. Our system does not take the POS tagging into accounts.

Example 1:

<pair id=21 label=F>

<t1> Xbox 微软 公司 产品</t1>

<t2>Xbox 不 索尼 公司产品</t2>

</pair>

2.2 Feature extraction

In this subsection, we mainly focus on 13 features used in our system.

(1) Vocabulary overlapping

This feature considers how many same words existing in t1 and t2. We use this feature because the more of the same words, the higher similarity between them, and the pair (t1,t2) is more likely to express the same meaning.

$$Sim(t1,t2)=\frac{Words(t1)\bigcap Words(t2)}{Words(t1)\bigcup Words(t2)}$$

Where, Words(t1) expresses the set of the words in text t1.

(2) Length difference

This feature considers length difference between t1 and t2. We use this feature because if text t1 entails hypothesis t2, t1 will be more informative than t2. And this rule shows in the surface that if t1 entails t2, the length of t1 is longer than the length of t2. And the most important thing is that if the lengths of t1 and t2 are almost equal, maybe the pair(t1, t2) is bidirection.

$$Sim(t1,t2)=|\,Length(t1)-Length(t2)\,|$$

(3) Length ratio

If the difference in the length is too large, the similarity of the pair(t1, t2) is low.

$$Sim(t1,t2)=\frac{Length(t1)}{Length(t2)}$$

(4) Manhattan distance

Manhattan distance defines the distance between two points measured along axes at right angles. In a plane with point p1 at (x1, y1) and point p2 at (x2, y2), it is |x1 - x2| + |y1 - y2|. Manhattan distance can be used to calculate the similarity between text strings.

$$L(\vec{t1},\vec{t2})=\sum_{i=1}^{n}|\,t1_i-t2_i\,|$$

Where $\vec{t1},\vec{t2}$ is the result of vectors t1 and t2 and we use TF-IDF method to calculate the vectors t1 and t2.

(5) Euclidean distance

Euclidean distance also defines the distance between two points and can be used to calculate the similarity between vectors.

$$L(\vec{t1},\vec{t2})=\sqrt{\sum_{i=1}^{n}(t1_i-t2_i)^2}$$

(6) The cosine similarity

The cosine similarity considers the similarity between t1 and t2 in vector space.

$$Sim(\vec{t1},\vec{t2})=\frac{\sum_{i=1}^{n}t1_i*t2_i}{\sqrt{\sum_{i=1}^{n}t1_i\sum_{j=1}^{n}t2_j}}$$

(7) Jaro-Winkler distance

The Jaro–Winkler distance is a measure of similarity between two strings. The higher the Jaro-Winkler distance for two strings is, the more similar the strings are. The Jaro-Winkler distance metric is designed and best suited for short strings such as person names. The score is normalized such that 0 means no similarity and 1 is an exact match.

$$L(t1,t2)=\frac{m}{3*Length(t1)}+\frac{m}{3*Length(t2)}+\frac{m-t}{3*m}$$

$$L_{JW}=\frac{\max(Length(t1),Length(t2))}{2}-1$$

Where m is the number of strings that text t1 matching text t2. Matching here means a string appearing in the t1 and t2 at same time and the position interval no more than L_{JW}.

(8) LCS similarity

The longest common subsequence (LCS) is to find the longest subsequence common to all sequences in a set of sequences.

$$Sim(t1,t2)=\frac{Length(LCS(t1,t2))}{\min(Length(t1),Length(t2))}$$

Where, LCS(t1, t2) refers to the longest common subsequence between t1 and t2.

(9) Sentence similarity

Assuming we have any two sentences t1 and t2. The words of t1 are w11,w12,…,w1m. The words of t2 are w21,w22,…,w2n. The words similarity between w1i(i>=1 and i<=m) and w2j(j>=1 and j<=n) is expressed by Sim(w1i,w2j). max{Sim(w1i,w2j)|j>=1 and j<=n} express that the maximum in Sim(w1i,w21), Sim(w1i,w22), …, Sim(w1i,w2j). And the sentences similarity based on CiLin can be calculated by the formula below.

[1] http://ictclas.org/

$$Sim(t1,t2)=\frac{1}{2}(\frac{1}{m}\sum_{i=1}^{m}\max\{Sim(w_{1i},w_{2j})\,|\,1\le j\le n\}+\frac{1}{n}\sum_{j=1}^{n}\max\{Sim(w_{1i},w_{2j})\,|\,1\le i\le m\})$$

We use tool in paper [1] to calculate words similarity.

(10) Semantic distance similarity

The sentences distance similarity based on HowNet can be calculated by the formula below.

$$Sim(t1,t2)=\frac{1}{2}(\frac{1}{m}\sum_{i=1}^{m}\max\{Sim(w_{1i},w_{2j})\,|\,1\le j\le n\}+\frac{1}{n}\sum_{j=1}^{n}\max\{Sim(w_{1i},w_{2j})\,|\,1\le i\le m\})$$

Here, we use tool in paper [2] to calculate the semantic distance similarity based on HowNet.

(11) Antonyms

To calculate the pair number of the antonyms in a pair(t1,t2), we must create a antonyms table. The n is the pair number of antonyms in a pair(t1,t2). If n is 0, we think there is no any antonym in the pair(t1,t2). If t1 and t2 are the same, the similarity of t1 and t2 is 1. And if n is not 0, the pair have one or more pairs antonyms may be contradiction.

$$Sim(t1,t2)=\begin{cases}0\,(n!=0)\\1\,(n=0)\end{cases}$$

The Sim(t1,t2) is the similarity of t1 and t2.

(12) Negative words

To calculate the number of negative word in each sentence, we must create a negative table. The n1 and n2 are the number of negative word in t1 and t2 respectively.

$$Sim(t1,t2)=\begin{cases}0\,(n1=n2 \text{ or } n1\%2=n2\%2)\\1\,(\text{otherwise})\end{cases}$$

The Sim(t1,t2) is the similarity of t1 and t2.

(13) Same words ratio in shorter sentence

According to our observation, we found the pair (t1, t2) in the following example should be classified to bidirection.

Example 2:

<pair id=21 label=F>

<t1>网易 首席 技术 总裁 丁磊</t1>

<t2>丁磊 网易 公司 首席 架构 师</t2>

</pair>

In this pair, the same words in the sentences are not much. If it is judged into bidirection, we can easily find that it is not reasonable. So I wish the following feature can be used to solve this problem.

$$Sim(t1,t2)=\frac{Words(\min(t1,t2))\,|_{similarity=1}}{\min(Length(t1),Length(t2))}$$

Where Words(min(t1,t2))|similarity=1 refers to the number of words which words similarity equals 1 in the shorter sentence between t1 and t2. Words similarity here is calculated based on CiLin.

2.3 SVM classifier

Here, we choose LIBSVM[2] as the classifier. LIBSVM is a library for support vector classification (SVM) and regression. After preparing and scaling data set in LIBSVM form, our system chooses the RBF kernel function to do the cross-validation.

Figure 3 and Figure 4 are the results of BC and MC respectively.

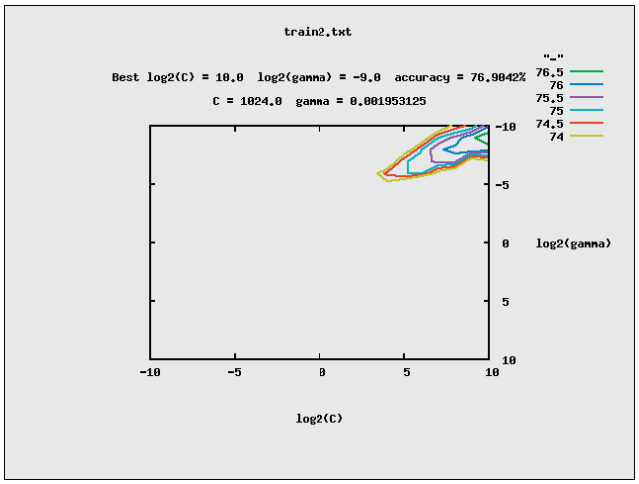

Figure 3. Training data for BC after cross-validation

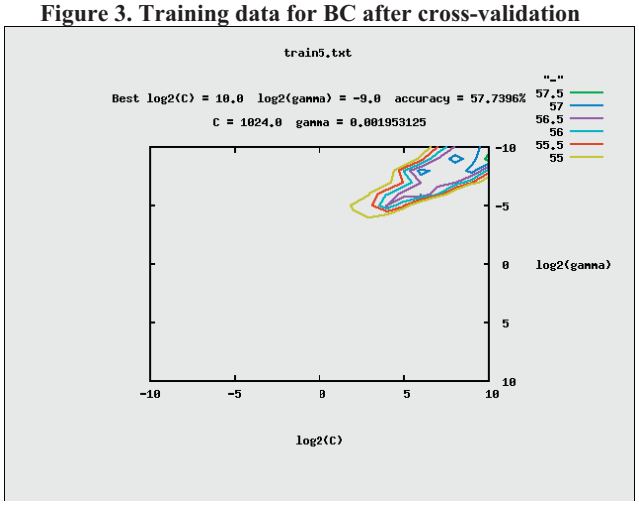

Figure 4. Training data for MC after cross-validation

3. Experiments

We submitted one result of BC and one result of MC to NTCIR-9. The official evaluation results of performance are listed in the Table 1. There is only one type of assessment, automatic assessment by accuracy.

Table1. Formal run experiment official results

Run	Subtask	Accuracy
RITE1-WUST-CS-BC-01	BC	0.725
RITE1-WUST-CS-MC-01	MC	0.582

Comparing with other groups, our system only achieves an intermediate official result. There are 12 groups participate in the BC subtask, and the results of 5 groups are better than us. There are 31 results of BC submitted, and 10 results are better than us. There are 11 groups participate in the MC subtask, and the results of 6 groups are better than us. There are 27 results of MC submitted, and 8 results are better than us.

3.1 BC subtask

Figure 8 is an experimental analysis of BC subtask. From the picture, we can get much information, and we put the analysis result in the Table 2.

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 2. Analysis of BC subtask

Label	Accuracy
Y	0.848
N	0.5

According to Table 2, we consider the accuracy of "Y" is good and the accuracy of "N" is bad. And we think that the most influence factors of accuracy are the judgment of the N mistakes.

```
---- Confusion Matrix ----

Cols: true labels
Rows: system labels

    |   Y    N |
----+----------+-----
 Y  | 223   72 | 295
 N  |  40   72 | 112
----+----------+-----
    | 263  144 |
```

Figure 5. Confusion Matrix of BC subtask

In BC subtask, we use the same features with MC subtask. And we have not experiment other features because we have no time. But BC and MC are different, we think if we choose some different features to BC, the accuracy of BC would be higher.

3.2 MC subtask

Figure 9 is an experimental analysis of MC subtask. From the picture, we can get much information, and we put the analysis result in the Table 3.

Table 3. Analysis of MC subtask

Label	Accuracy
F	0.683
R	0.824
B	0.887
C	0.094
I	0.329

According to Table 3, we can find that contradiction is almost wrong and independence is not ideal. And we think that the most influence factors of accuracy are the judgment of the C and I mistake.

```
---- Confusion Matrix ----

Cols: true labels
Rows: system labels

    |   F    R    B    C    I |
----+-----------------------+-----
 F  |  69    0    3    6   15 |  93
 R  |   0   75    4    5   21 | 105
 B  |  17    9   63   54    8 | 151
 C  |   3    1    1    7    3 |  15
 I  |  12    6    0    2   23 |  43
----+-----------------------+-----
    | 101   91   71   74   70 |
```

Figure 6. Confusion Matrix of MC subtask

In the experiment, we have found that contradiction is difficult to judge. We added two features, antonyms and negative words, to solve this problem, but the result is not better yet. We need to try more features to solve the contradiction problem.

According to Figure 6, we misjudge 21.4% of independence to forward, 30% to reverse and 11.4% to bi-direction. Forward, reverse and bi-direction belong to entailment, so 62.8% of "I" classified to entailment. We think the reason is that we narrowed the scope of the independent. We think "I" is the pair which the sentence similarity is lower. But actually the sentence similarity of many independence pairs is high. So this outcome caused. If we consider more features to expand the boundary of independence, the more improvement we will get.

4. Conclusions

Our RITE system depends on the method of classification. The key to improve the accuracy we consider is features. In our system, we use the same feature in BC and MC, but BC and MC are different, we think if we choose different features to BC, the accuracy of BC would be higher. In the MC subtask, many pairs would like be judged into bidirection. So we think we should add some features to limit the judgment of bidirection. By experiments, this method can let the accuracy improve. In the system, we almost consider only statistical features, we think if we add some rule features; the accuracy will be significantly improved.

ACKNOWLEDGMENTS

The work in this paper was supported partially by the National Natural Science Foundation of China (No. 61100133 and 61070243), the Natural Science Foundation of Hubei Province (No. 2009CDB311) and Social Science Grant of Department of Education of Hubei Province (No. 2011jyte126).

REFERENCES

[1] Tian Jiule, Zhao Wei. Words Similarity Algorithm Based on Tongyici Cilin in Semantic. Journal of Jilin University (Information Science Edition),2010,28(6):602-608

[2] Qun Liu , Sujian Li. Word Similarity Computing Based on How-net.Computational Linguistics and Chinese Language Processing,2002,7(2):59-76

[3] P.Schlenker. Meaning II: Entailments. Linguistics1: Introduction to the Study of Language, 2006.

[4] Ido Dagan, Dan Roth, Fabio Massimo. Textual Entailment. ACL, 2007.

[5] Jonathan Berant, Ido Dagan, Jacob Goldberger. Global Leaning of Focused Entailment Graphs. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010:1220-1229.

[6] Roni Ben Aharon, Idan Szpektor, Ido Dagan. Generating Entailment Rules from FrameNet. Proceedings of the ACL 2010 Conference Short Papers, 2010:241-246.

[7] Ion Androutsopoulos, Prodromos Malakasiotis. A Survey of Paraphrasing and Textual Entailment Methods. Journal of Artificial Intelligence Research, 2010: 135-187.