ZSWSL Text Entailment Recognizing System at NTCIR-9 RITE Task

Ranxu Su, Sheng Shang, Pan Wang, Haixu Liu, Yan Zheng
School of Computer Science
Beijing University of Posts and Telecommunications
100876, Beijing, China
suranxu.bupt@gmail.com

ABSTRACT
This paper describes our system on simplified Chinese textual entailment recognizing RITE task at NTCIR-9. Both lexical and semantic features are extracted using NLP methods. Three classification models are used and compared for the classification task, Rule-based algorithms, SVM and C4.5. C4.5 gives the best result on testing data set. Evaluation at NTCIR-9 RITE shows 72% accuracy on BC subtask and 61.9% accuracy on MC subtask.

General Terms
Algorithms, Performance, Design, Experimentation

Keywords
Textual Entailment, Feature Extraction, C4.5, NTCIR, RITE

1. Introduction
Given two text fragments, Recognizing Textual Entailment (RTE) is a task of deciding whether one text can be inferred (entailed) from the other [1]. Textual entailment captures a broad range of semantic oriented inferences needed for many Natural Language Processing (NLP) applications, like Information Retrieval, Text Summarization, Information Extraction, Question Answering and Machine Translation. As one of the fundamental problems in those NLP applications RTE has attracted increasing attention in recent years.

In this paper, we discuss the use of our system in the NTCIR-9 RITE task [2]. RITE is a text entailment reorganizing evaluation task which focuses on Asia languages. We participate in BC and MC subtasks on simplified Chinese text, where BC is a binary classification subtask of "entailment" or "no entailment", MC is a multi-class classification subtask of "contradiction", "independent", "forward entailment", "reverse entailment" or "bidirectional entailment". Figure 1 shows an example of how the training data looks like.

```
<pair id="103" label="B">
        <t1>安南来自非洲加纳</t1>
       (<t1>Annan comes from Africa Ghana</t1>)
        <t2>安南出身于非洲加纳</t2>
      (<t2> Annan was born in Africa Ghana</t2>)
</pair>
```

Figure 1. RITE Training Data Set

According to the requirements of the task, we developed a textual entailment system that could handle multi-class entailment recognition. We concentrate on feature extraction using natural language processing method, then use and compare a rule-based and two machine learning algorithms for classification based on these features. In the MC subtask, we find to distinguish "contradiction/independent" from entailment, only using features based on mutual information is not enough; features that represent the differences between two sentences are also important indicators. Furthermore, semantic information is crucial in MC subtask. Semantic dictionary and semantic analysis is applied in feature extraction to generate semantic features.

The rest of the paper is organized as follows: Section 2 provides related works. Section 3 introduces the system in detail from preprocessing to classification labeling. Section 4 shows the evaluation result in RITE task and discussions. Section 5 draws conclusion and future work.

2. Related Works
Jin et al. [3] proposed a feature match method based on exploiting the relation in the WordNet glosses, and reached 52.4% accuracy on RET1, 58.9% accuracy on RET2; Later Jin et al. [4] proposed another new method based on lexical and shallow syntactic analysis combined with fuzzy set theory, and reached 56% accuracy on RET1; Georgiana et al. [5] explored a way of improving an inference rule collection and its application to the task of recognizing textual entailment using refined method and a hand-crafted lexical resource. The method automatically found phrase patterns representing the same meaning, for example "X wrote Y" ≈ "X is author of Y". The method reached 60.00% precision on covered RTE2 data set. Although the precision on full data set was 57.75% due to the low coverage, this method did a nice try in looking for better features other than lexical bag-of-words features. Partha Pakray Sivaji Bandyopadhyay et al. [6] proposed a rule-based syntactic feature extraction method together with a multi-gram lexical feature extraction method. Subject-subject, Subject-verb, object-verb and cross subject-verb comparison were selected as syntactic features. They chose SVM as classification model and reached 55.6% precision on RTE4 data set. Yongping et al. [7] gave additional attention to part-of-speech and named entity in RET task, and got good results.

However, few experiments were implemented as multi-classification tasks. And as far as we known, no textual entailment experiments on Chinese text have been implemented yet.

3. System Description

3.1 System Architecture

As shown in figure 2, the system is composed of two major process, feature extraction and classification. Multiple NLP methods are applied, including word segmentation, POS, syntactic analyze and semantic analyze. TongYiCi CILin[8] and an antonym dictionary are used to construct semantic features. Using features generated above, we implemented rule-based algorithms, SVM and C4.5 for classification. Evaluation and comparison were conducted after the experiments.

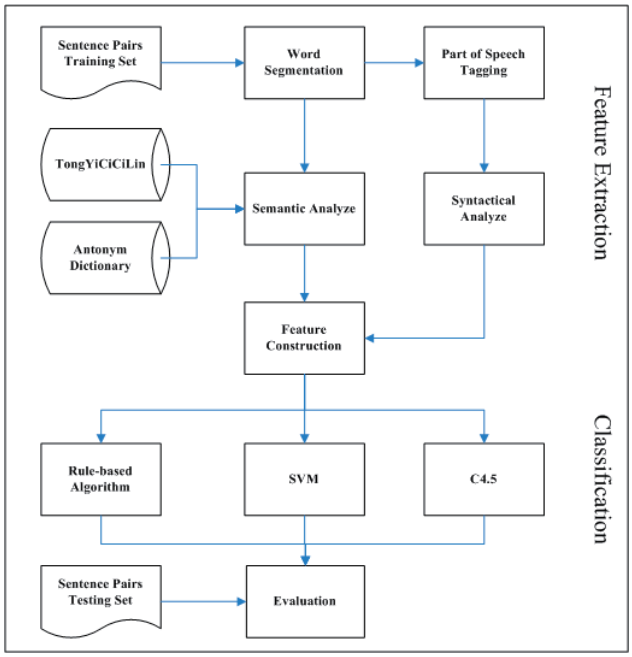

Figure 2. System Architecture

3.2 Word Segmentation and POS

Different from English text, Chinese characters are written adjacent to each other with no space between each word. The reason is that ancient Chinese word is composed of one single character in most cases, so no space is needed to separate them. Modern Chinese term however, is always composed of two characters or three characters, so it's important to do word segmentation before applying other common NLP technologies. For example, in English, "Knowledge is power" is naturally segmented into three words: "knowledge", "is" and "power". But in Chinese, "知识就是力量" can not be separated into "知识", "就是", "力量" likewise because there are no space between them. Here we use an open source program ictclas4j[9] for Chinese word segmentation. Ictclas4j is an open source Chinese lexical analysis program based on Hierarchical Hidden Markov Model.

POS (part-of-speech) is to label each word with noun/verb/adjective etc. Figure 3 gives an example. POS provides fundamental information for feature extraction. On one hand, it provides labels that could directly be used for lexical feature construction. On the other hand, POS labels can be used as features for syntactic analysis. We also use it to adjust the syntactic analysis result from Stanford Parser[10] which would be introduced later section in detail. In this task, Ictclas4j is also used to conduct POS tagging.

Text: 知识就是力量
(Knowledge is power.)
POS: 知识/n 就是/vshi 力量/n
(Knowledge/n is/vshi power/n)

Figure 3 POS Example

3.3 Syntactical Analyze

Syntactical analyze gives the structure of a sentence. Figure 4 shows an example. Syntactical analyze is crucial in the task because our inputs are two sentences, and much information is embedded in the sentence structures. As we have seen in related works section, many researchers have used syntactic features for their RTE task on English text. However, a major difference in Chinese syntactic analyze is that the accuracy of the state-of-art of Chinese syntactic parsing is unsatisfactory, which has been a bottle neck of Chinese NLP in recent years [11].

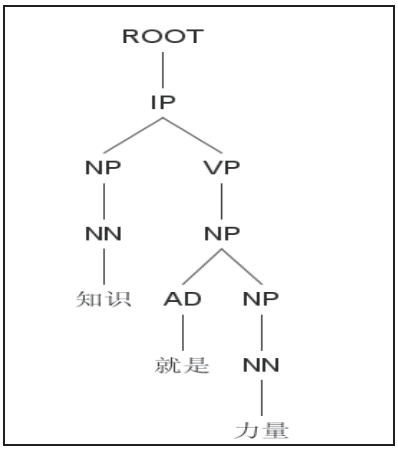

Figure 4. Syntactical Analysis Example

We have compared several different syntactic parsers and decided to use Stanford Parser in the end. In our evaluation, Stanford Parser's average accuracy is about 70%. Since we have a pair of sentences for each judgment, the accuracy for each pair goes to about 50%. To improve the accuracy, we use lexical analyze results from POS, named-entity identification and manually designed rules to re-check Stanford Parser's result. If we detected that Stanford Parser has failed, we set all syntactic feature values to null to decrease the bad influence. For example if the parsed syntactic tree has no VP node or NP node, this parse must have failed. We also developed several post process rules to try to fix the result of the syntactic tree if the error is small. For example, if a person's name has been spited into several nodes, they will be merged as an NR node.

3.4 Semantic Analyze

Both lexical features and syntactical features only rely on literal match. A major problem with it is that synonyms will not be considered as the same and antonyms won't be identified. In RITE task, we find it important to deal with both synonyms and antonyms. To do that we have tried different resources, and come

down to two dictionaries, TongYiCi CiLin and an antonym word list summarized from the Web.

TongYiCi CiLin is a 5-layer Chinese synonym word dictionary. Figure 5 shows a snapshot of CiLin. Words with similar meanings are organized in lines. A line of Chinese words share the same id and they have the same meaning. Ids are organized in a 5 layers manner, so that words' similarity can be compared simply by their ids. Generally speaking, the longer common prefix two ids share the more similar they are. We use CiLin to expand the literal match on important nouns and verbs, words are considered matched if they're synonyms.

Hi15C02@ 医嘱
Hi16A01= 介绍 引见 牵线 穿针引线
Hi16A02= 说合 撮合 说 说说
Hi16A03= 说媒 做媒 保媒 说亲 提亲

Figure 5. TongYiCi CiLin

Antonym on the other hand is also important in RITE task because we have to identify the "Contradiction" sentence pair, and many contradict sentences have antonyms. We couldn't find any effective open source Chinese antonym dictionary, so we use resources from the internet to summarize the antonym dictionary ourselves.

Other than synonym and antonym, named entities are also important. Ictclas4j is used to for named entity identification. We also used a person name identification program developed by our own lab to improve named identification accuracy.

3.5 Feature construction

Since the object is to recognize entailment between two sentences, the features that represent two sentences' relationships are the features we are looking for. So we combine lexical, syntactical and semantic analyze results together, and use comparison results between two sentences as features. We've constructed 35 features totally including lexical and semantic features. In this section, we'll list a few of the most effective ones and explain them in detail.

3.5.1 Synonym based Word Match

The ratio of common words is always a strong feature to indicate whether two sentences could be entailed from each other. Though we do have seen exceptions, in most of the scenario, the more two sentences share words in common, the more likely they have entailment relationships. Here we consider synonyms equal to the same words, and use the common word count divided by the shorter sentence's word count as common words' ratio. This feature is simple but proved to be effective in our experiment. On the other hand, we do use other features to cover those exceptions, which are more challenging and will be introduced later.

3.5.2 Length Compare

Apart from determine if entailment exist between two sentences, it's also important to identify the direction of entailment, as there're three directions: forward, reverse and bidirectional. In cope with word match feature, the length compare ratio can be a strong feature to indicate the direction linguistically. The more one sentence is longer than the other, the more likely the direction is from the former to the latter. The smaller the length difference is, the more likely it is a bidirectional direction. The feature is simple and effective. But again, there're exceptions. We use more complicated features which will be introduced later to cover those exceptions.

3.5.3 Named Entity Match

Named entities are the key components of a sentence, including time, location, person name, organization name etc. Many researchers use named entity match just like word match, which is to calculate the common ratio of the named entity between two sentences. But we believe there's more information embedded in named entities, which can't be revealed by common ratio feature. Firstly, named entities should be compared category-wisely. There's no point comparing a person's name with a location even if they're the same; secondly, named entities should be compared with more dimensions, direction and confliction should also be considered. If one sentence has a time entity but the other sentence doesn't, this may be crucial to determine a forward direction when they have entailment relation because the first sentence has more key information. Things are the same when one of the entities is more specific than the other, for example "October 30th in 2011" is more specific than "October 30th". In another case, if two sentences share a lot of common words, but have different time entity, they may have a contraction relationship, because they're probably describing the same event with different event time. Thus we constructed the named entity feature into a 5 dimensional vector with each dimension represent a type of relation—same, forward entailment, reverse entailment, different, independent. For each vector only one dimension will be set as 1 and the others all as 0. Fore example, forward entailment may be represented as <0, 1, 0, 0, 0>, and contradiction may be represented as <0, 0, 0, 1, 0>. This way, not only contradiction has a strong feature resource, but entailment direction can also benefit. This feature covered some exceptions mentioned earlier if they have different named entities.

3.5.4 Different Verb Number

Apart from named entities, verbs are always associated with events. If two sentences have a lot of different verbs, they're most likely describing two different events, although two sentences have few different verbs doesn't necessarily mean they are describing the same event. The different verb number is a good feature to identify "independent" scenario, and may assist to identify "contradiction".

3.5.5 Antonym Number

Two entailed sentences are consistent in opinion expression and are not likely to have antonym words with each other. So antonym is a strong indicator in identifying "contradiction" scenario, especially when it is used together with syntactic information and the antonyms are judged between words of the same syntactic roles. But since we have a Chinese syntactic analyze bottle neck problem, here we simply use the number of antonyms two sentences have as a raw feature. Experiments show that although the method is raw and simple, it is effective in identifying contradiction sentence pairs. We'll later introduce the syntactic-wise antonym comparison feature that we also used.

But since the syntactic results are always nulls, that feature actually contributed less though it is better in term of accuracy.

3.5.6 Syntactic Match

After analyzing cases in RITE task, we find the syntactic-wise comparison will be beneficial and will cover most of the exceptions in lexical match. Specially, we focus on the role of subject, predicate, object and the attributes of these three roles. To extract these roles from a sentence, we developed a few common patterns in Chinese language. Every pattern represents a type of Chinese sentence structure, and each sentence may fit into one or more patterns. Figure 6 shows the pattern matching logic. Briefly speaking, we consider the backbone of a sentence with 3 types of patterns: a) passive pattern, where the sentence has a structure of <object, passive verb, subject>. e.g. "John was praised by his teacher"; b) linked-verb pattern, where subject is defined by attributes, with a structure <subject, linked-verb, attribute>. e.g. "John is a good student"; c) standard pattern, where the sentence is organized as <subject, verb, object>, sub-sentence is also considered in standard pattern. e.g. "Lucy said John ate the apple". Besides the backbone structure, we also consider other sub structures like alias, reason, attribute block and parataxis structure, to make sure no attribute information is lost or filtered after the syntactic analyze.

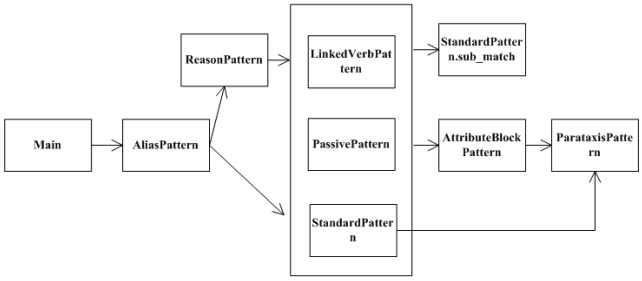

Figure 6. Pattern Matching Logic

To identify these patterns, we use a variety of features from lexical word to POS, named entity etc. But most importantly, we use the syntactic tree parsed by Stanford Parser. Take the standard pattern for example, it always fit into a [DNP]<NP>[ADVP]<VV>[DNP]<NP> tree structure, where NP, VP, NP represent subject, predicate, object, DNP represent attributes to subject or object, ADVP represent adverbs to predicate. Figure 7 shows a sentence that fit the standard pattern, where DNP: "中国队的"(from China), NP:"刘翔"(Xiang Liu), ADVP: "再次" (again), VV: "获得" (win), DNP: "一百一十米栏" (110 meter hurdle), NP: "冠军" (champion). This is just a standard example, in real data informal expressions exist and the pattern doesn't always fit perfectly. Some components may be missing and Stanford Parser may have an incorrect parsed result. We use the POS result to validate the syntactic tree, and use several rules to adapt to informal expressions.

After we identify different roles of the word, we construct syntactic-wise match features that aggregate with former lexical and semantic features, especially with semantic features. In antonym comparison, we not only consider the antonyms themselves but also the syntactic context they are in. Antonym only has accurate contradiction indication when the antonyms are attributes of the same entity, E.g. "The horse runs fast" contradicts with "The horse runs slowly", but is independent with "The tortoise runs slowly". Also, compared with different verb, we consider different predicate as an even stronger feature. Furthermore, length compare has its syntactic version of role number compare, where only the number of subject, predicate, object and attribute are taken into account rather than all the words. This feature covers sentences which is short but contains a lot of key information. These features are organized as compare ratio or vectors just as lexical and semantic features. The experiments do have proved they have better accuracy. But as we've mentioned, to overcome the syntactic analyzer's low precision problem, syntactic features would be set as null if the result is not confident enough. Thus though these features cover some of the exceptions lexical features couldn't handle, overall they have less coverage and contribute less than the lexical and semantic features.

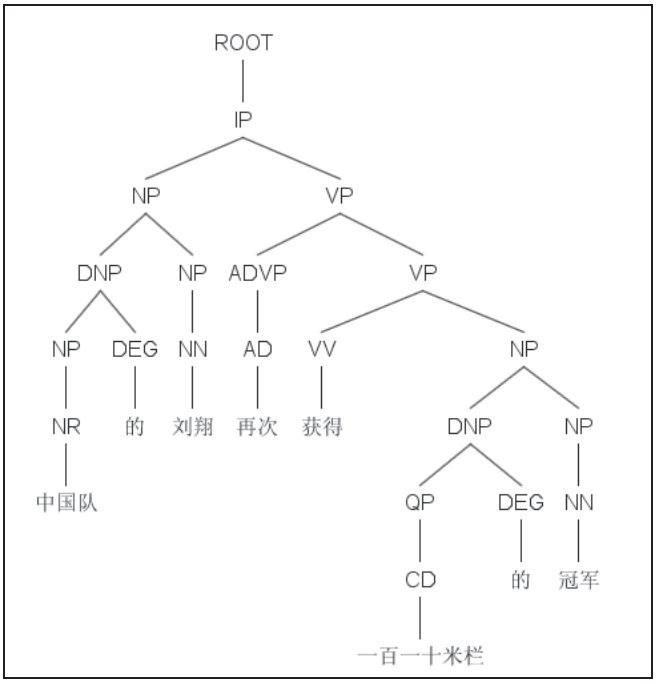

Figure 7. Standard Pattern Example

3.6 Classification Models

We use a simple rule-based algorithm as baseline. We also used and compared the state-of-art SVM model and C4.5 decision tree model as classification models.

3.6.1 Rule-based algorithm

Rule-based algorithm considers features with different priorities, and uses a combination of feature values to make a decision. An overview of this algorithm can be seen in Figure 8. The process is simple and easy to understand. The problem is that both the logic and the thresholds are set arbitrarily. And since some logic paths are hard to reach, like bidirectional entailment, they may end up with low recall.

```
If (Syntactic Features are not null and have a conclusion)
        Return Conclusion;
Else
    If (Synonym based Word Match < 0.5)
        Return Independence;
    Foreach Named Entity feature
        If (contradiction found)
            Return Contradiction;
        If (forward entailment found)
            Return Forward Entailment;
        If (reverse entailment found)
            Return Reverse Entailment
    If (Length Compare > 0.7)
        Return Forward Entailment;
    Else if (Length Compare < 0.3)
        Return Reverse Entailment;
    Else
        Return Bidirectional Entailment;
```

Figure 8. Rule-based Algorithm

3.6.2 SVM Model

Support Vector Machine (SVM) is considered as the state-of-art classification model academically and has been applied to many applications [12]. SVM can deal with non-linear features and does not have the over-fitting problem. We use Radial Basis Function (RBF) as the kernel function shown in equation 1. RBF function is generally a good kernel function for SVM, especially when the feature number is small.

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0 \qquad (1)$$

However our features in RITE task are different than traditional text classification features. In our feature set, some features only make sense when they're combined together. The problem with using SVM is that the combination rules are various and hard to be described by just one single kernel function. Furthermore, SVM is a binary classification model, though it could adapt to multi-classification tasks, this adaptation may decrease its accuracy.

3.6.3 C4.5 Model

C4.5 [13] is a decision tree model. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets with minimum entropy, i.e. the largest information gain. The attribute and the split value with the highest normalized information gain are chosen to make the decision.

In RITE task, C4.5 use combination of features to make decisions just like rule-based algorithm. The difference is that it chooses the most effective feature with the most effective threshold, which is not arbitrary. Furthermore, C4.5 training process could identify the syntactic feature's confidence using potential information in other features, which further reduces influence from incorrect syntactic parsing result. The problem with C4.5 is that it could have over-fitting problem, and weaker adaptation ability to the new data.

4. Experiments and Evaluation

4.1 Experiment Setting

We use 2/3 RITE develop data set as training set, 1/3 develop data set as testing set to compare our three different models. We also use RITE evaluation data set to evaluate our final system.

We use LibSVM[14] to perform SVM classification, RBF is chosen as the kernel function, grid search is applied for parameter estimation with a 10 folder cross validation. We use Weka[15] to perform C4.5 decision tree experiment, 3 folder cross validation is used for training.

4.2 Experiment Result

Firstly, we test three models independently on our testing set. Then we used C4.5 which performed the best for evaluation.

Table1 Experiment result on rule-based algorithm

Rule-based Algorithm	Precision
Independent	43.7%
Contradiction	15.2%
Bidirectional Entailment	58.5%
Forward Entailment	57.1%
Reverse Entailment	48.4%
Average	46.1%

Table 1 shows the test result of role-based algorithm. Contradiction shows the lowest precision of 15.2%, indicating the contradiction rule is too arbitrary.

Table 2 Experiment result on SVM model

SVM Model	Precision
Independent	47.1%
Contradiction	24.0%
Bidirectional Entailment	45.9%
Forward Entailment	57.6%
Reverse Entailment	52.0%
Average	46.0%

Table 2 shows the test result of SVM. Though it gives less arbitrary results, the average precision is almost the same with rule-based algorithm.

Table 3 Experiment result on C4.5 model

C4.5 Decision Tree	Precision
Independent	38.1%
Contradiction	25.0%
Bidirectional Entailment	60.8%
Forward Entailment	56.7%
Reverse Entailment	69.2%
Average	55.9%

Table 3 shows the test result of C4.5 model. Compare to rule-based algorithm, C4.5 has a 10% improvement, which also outperforms SVM. Finally we use C4.5 in the evaluation task.

4.3 Evaluation Result

Table 4 Evaluation Result

Sub-Task	Accuracy	Team Rank
BC	72.0%	7/12
MC	61.9%	3/11

The evaluation result is shown in Table 4. As the system is designed particularly for MC subtask, the result shows that our

system performs better in MC than BC in term of team rank. Our system reached a good accuracy of 61.9% on MC formal run evaluation set, which ranks the 3rd place among all participated teams. On BC sub set, the result isn't quite satisfactory compared to other teams, but still it's quite a good result of 72.0% accuracy.

4.4 Discussions

The testing and evaluation results have confirmed our analysis of different models in section 3.6. For RITE MC subtask, C4.5 is the best model performed on our features, since it handles the combination of features flexibly which is optimal statistically. Also we found semantic and syntactic features are helpful. But due to the low accuracy of Chinese syntactic parsers, we must use syntactic features with cautious.

5. Conclusion and future work

In this paper, we introduce our text entailment system for NTCIR-9 RITE task. We extract lexical features, syntactic features and semantic features from Chinese text. Two semantic dictionaries are used, CiLin and antonym dictionary. Stanford Parser is used for syntactic analyzing. We use and compared three different models based on these features, and C4.5 outperform rule-based algorithm and SVM. Evaluation result shows a good accuracy of 72.0% in BC sub task and 61.9% in MC subtask.

Although many semantic features are extracted, no semantic inference is performed in our system. During our work, we find to develop an inference framework with an inference knowledge base that links up human's common sense with language grammar would be an interesting and very beneficial work.

6. References

[1] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge, Machine Learning Challenges, vol:3944. Lecture Notes in Computer Science, p 177-190, 2006

[2] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In NTCIR-9 Proceedings, to appear, 2011.NTCIR-9.

[3] Jin Feng, Yiming Zhou, Trevor Martin. Recognizing Textual Entailment based on WordNet, Proceedings - 2008 2nd International Symposium on Intelligent Information Technology Application, IITA 2008, v 2, p 27-31, 2008,

[4] Jin Feng, Yiming Zhou, Trevor Martin. Combining Lexical Resources with Fuzzy Set Theory for Recognizing Textual Entailment, 2008 International Seminar on Business and Information Management, ISBIM 2008, v 2, p 54-57, 2008

[5] Georgiana Dinu, Rui Wang. Inference Rules and their Application to Recognizing Textual Entailment, Business and Information Management, 2008. ISBIM '08. V 2, p 54-57 2008

[6] Partha Pakray Sivaji Bandyopadhyay, Alexander Gelbukh. A Hybrid Textual Entailment System using Lexical and Syntactic Features, Proceedings of the 9th IEEE International Conference on Cognitive Informatics, ICCI 2010, p 291-296, 2010

[7] Yongping Ou, Changqing Yao. Recognize Textual Entailment by the Lexical and Semantic Matching, ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings, v 2, p V2500-V2504, 2010

[8] Wanxiang Che, Zhenghua Li, Ting Liu. LTP: A Chinese Language Technology Platform. In Proceedings of the Coling 2010: Demonstrations. 2010.08, p13-16, Beijing, China, 2010.

[9] iAmSin...@gmail.com. ictclas4j open source project, http://code.google.com/p/ictclas4j/, 2007

[10] Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. Discriminative Reordering with Chinese Grammatical Relations Features. In Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation, p51-59, 2009.

[11] Xin WANG, Weiwei SUN, Zhifang SUN. A Lightweight Chinese Semantic Dependency Parsing Model Based On Sentence Compression, Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2010, 2010

[12] Li Meng; Wang Miao; Wang Chunguang. Research on SVM Classification Performance in Rolling Bearing Diagnosis, 2010 International Conference on Intelligent Computation Technology and Automation, Volume: 3 , p132-135, 2010

[13] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011

[15] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. The WEKA Data Mining Software: An Update, SIGKDD Explorations, Volume 11, Issue 1. 2009