NCCU-MIG at NTCIR-10: Using Lexical, Syntactic, and Semantic Features for the RITE Tasks

Wei-Jie Huang Chao-Lin Liu
Department of Computer Science, National Chengchi University
64 Chih-Nan Road Section 2, Wen-Shan, Taipei 11605, Taiwan
{100753014, chaolin}@nccu.edu.tw

Abstract
We computed linguistic information at the lexical, syntactic, and semantic levels for the RITE (Recognizing Inference in TExt) tasks for both traditional and simplified Chinese in NTCIR-10. Techniques for syntactic parsing, named-entity recognition, and near synonym recognition were employed, and features like counts of common words, sentence lengths, negation words, and antonyms were considered to judge the logical relationships of two sentences, while we explored both heuristics-based functions and machine-learning approaches. We focused on the BC (binary classification) task at the preparatory stage, but participated in both BC and MC (multiple classes) evaluations. Three settings were submitted for the formal runs for each task. The best performing settings achieved the second best performance in BC tasks, and were listed in the top five performers in MC tasks for both traditional and simplified Chinese.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *language models, language parsing and understanding, text analysis*; I.2.6 [**Artificial Intelligence**] Learning – *knowledge acquisition, parameter learning*; H.2.8 [**Database Management**]: Database Applications – *data mining*.

General Terms
Algorithms, Experimentation, Languages

Keywords
Entailment Recognition, Named-Entity Recognition, Near Synonym Recognition, Heuristic Functions, Support-Vector Machines, Negation and Antonyms

Team Name
MIG

Subtasks
Traditional Chinese BC, Simplified Chinese BC, Traditional Chinese MC, Simplified Chinese MC

1. Introduction
Recent advances in natural language processing have led to the research work for deep understanding of text material. Among the many developments, recognizing entailment between sentences has drawn attentions from many researchers these years, and is one of the most important techniques to capture the meanings of texts.

Recognizing the entailments between sentences has many potential applications. It can improve the ability of information access in QA systems, in which query expansions for high recall in retrieval systems. RITE can also be used in intelligent tutoring systems to evaluate students' understanding of abstract concepts [7].

Since the First Recognizing Textual Entailment Challenge (RTE-1) was held in 2005 [4], recognizing entailments between sentences has become a popular research topic. For Asian languages, NTCIR-9 first provides evaluation standards on recognizing entailment systems [3]. Moreover, NTCIR-10 holds the second Recognizing Inference in Text (RITE-2) competition in 2013 [10]. It helps more researchers join the research field.

RITE task asks participating systems to classify the entailment relation between two sentence t_1 and t_2. There are two subtasks, including binary classes (BC) and multiple classes (MC). Our systems are developed to classify Yes (Y) or No (N) in BC. There are four classes in MC, bidirectional (B), Forward (F), Contradiction (C) and Independence (I). We extended our BC methods to infer MC classes.

We computed lexical, syntactic, and semantics information about the sentence pairs to judge their entailment relationships. The information was computed by public tools and machine-readable dictionaries [2]. Preprocessing steps of the original sentences included the conversion between simplified and traditional Chinese, Chinese segmentation, and converting formats of Chinese numbers. Major linguistic information used in the recognition of entailment included words that were shared by both sentences, synonyms, antonyms, negation words, the similarity of the parse trees, and information about the named entities of the sentence pair.

We explored both heuristic functions and support-vector machine models for the classification task. The best performing systems of our submissions were ranked the second position in the BC tasks for both traditional and simplified Chinese, the third for the MC task for traditional Chinese, and the fifth for the MC task for the simplified Chinese.

We describe the preprocessing steps and named-entity recognition in Section 2, discuss both the heuristic function and the machine-learning based classification models in Section 3, present the evaluation results in Section 4, and summarize and discuss our experience in Section 5.

2. System Components
In this section, we briefly describe components of the running system, including preprocessing units and semantic lexicon search.

2.1 Preprocessing
In this subsection, we explain the preprocessing functions: simplified Chinese conversion, numeric format conversion, Chinese segmentation and named-entity recognition.

2.1.1 Simplified Chinese Conversion
We relied on public tools to do Chinese segmentation and named-entity recognition, and those tools were designed to perform better for simplified Chinese. Hence, we had to convert traditional Chinese into simplified Chinese. We converted words between their traditional and simplified forms of Chinese with an automatic

procedure which relied on a tool in the packages of Microsoft Word. We did not design or invent a conversion dictionary of our own, and the quality of conversion depended solely on Microsoft Word.

2.1.2 Numeric Format Conversion

There are several formats of numbers in Chinese. The differences between numeric formats may lead to ambiguous interpretations of sentences. In order to solve the problem, regular expressions were used to capture specific strings and converted Chinese numerals into Arabic numerals. Figure 1 shows an example of the conversion.

Original: 蘇哈托政權在一九九八年五月廿一日結束
(The Suharto regime ended in May 21, 1998)
Conversion: 蘇哈托政權在 1998 年 5 月 21 日結束

Figure 1. Numeric Format Conversion Example

2.1.3 Chinese Segmentation

Words are separated by spaces in English sentences, but it is quite different in Chinese. It is much harder for computers to separate individual words in Chinese strings. Figure 2 shows an example of Chinese segmentation. In addition, the meanings of Chinese sentences may change because of different segmentation results. Stanford Word Segmenter was used to generate Chinese segmentations in our system [9].

Original: 若望保祿二世是教廷國家領導人
(John Paul II is the national leader of the Holy City)
Segmented: 若望保祿 二 世 是 教廷 國家 領導人

Figure 2. Chinese Segmentation Example

2.1.4 Named-entity Recognition

We expect that proper nouns provide important information for entailment recognition. S-MSRSeg is an NER tool developed by Microsoft Research in 2005 [5]. We used it to identify names of persons, locations and organizations. In Section 3, we will describe how to use information about named entities to recognize entailments.

Word: 溶解 (dissolve)
Expression: {StateChange|態變:StateFin={StateLiquid|液態}}

Figure 3. E-HowNet Lexical Sense Expression

2.2 Lexical Semantics

We are also in the belief that synonyms, antonyms and negation words in sentences are important in judging the meanings of sentences. E-HowNet employs an entity-relation model to represent lexical senses. It contains 88079 traditional Chinese words in its 2012 version. Word senses are defined by primitives, concepts and conceptual relations. Figure 3 shows a lexical sense expression in E-HowNet.

Some words are often used to express negation meanings. Hence, we want to capture these words in sentences. We retrieved words whose E-HowNet expressions contain function word "not", and they were filtered manually to construct a negation words dictionary. The dictionary was applied to check whether the sentences have opposite meanings.

Furthermore, antonyms may also be used to represent opposite meanings in two sentences. Hence, our system should be capable of capturing antonyms in sentences. We detect antonyms by an antonym dictionary that was provided by the Ministry of Education in Taiwan [1].

In our approach, we tried to retrieve synonyms to improve our ability in identifying overlapping words in sentence pairs. Based on E-HowNet expressions, we proposed a method to compute the similarity between words and output a confidence score from 0 to 1. We would then set a threshold to tell whether two words were synonyms by experiments.

To compare whether two words are similar, we first parse their E-HowNet expressions into tree structures as shown in Figure 4. Each node represents one primitive, function word, or a semantic role. From the tree structures, we can realize the relations between nodes. We think the relations show how two words share their semantics in more details. We compute the similarity between two words by how their tree structures match, and output a similarity score.

Word: 漲價 (Increase in price)
Expression: {BecomeMore| 增 多 :domain={economy| 經濟},theme={price({object|物體})}}
Tree:

Figure 4. E-HowNet Expression and Tree Structure

3. Recognition Methods

We applied different linguistic information to recognize entailment between two sentences. Several NLP tools, dictionaries and semantic resources were used to construct the heuristic functions and to extract features from sentences. The features were trained as classification models by machine learning algorithms. Initially, we focused on recognizing binary relations when we designed these two approaches. A while later, we determined to participate in MC tasks and chose to generate our decisions based on BC models. Figure 5 shows the inferences from BC to MC. Notice that we did not do the Contradiction decisions with the right model,

if t1 →t2 =' Y' and t2 → t1 = 'N' then output 'F'
if t1 →t2 = 'Y' and t2 → t1 = 'Y' then output 'B'
if t1 →t2 =' N' and t2 → t1 = 'N' then output 'I'
if t1 →t2 =' N' and t2 → t1 = 'Y' then output 'C'

Figure 5. Inferences from BC to MC

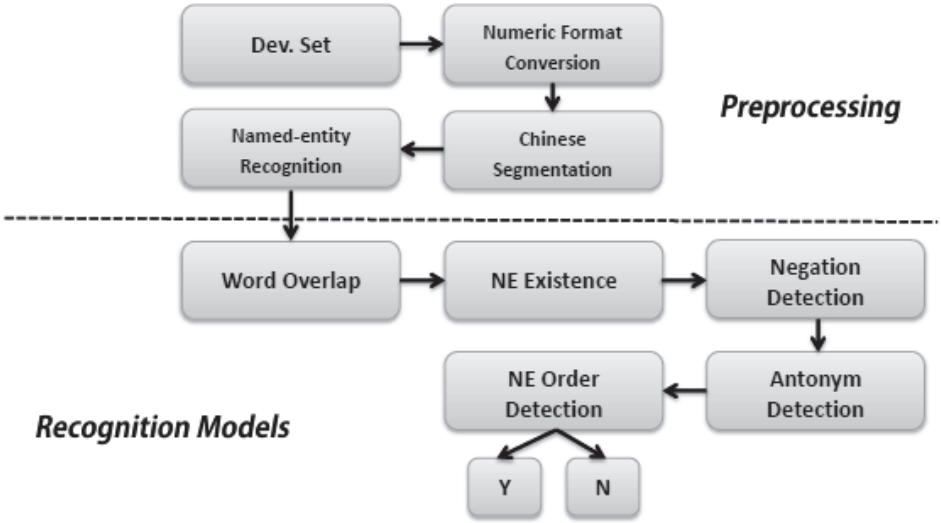

Figure 6. System Architecture

and our performance in this category in the formal runs reflected this flaw.

3.1 Heuristic Functions

Figure 6 shows the system architecture with which we learned the threshold from training data for entailment decisions and used to compute scores of test data. First, preprocessing steps mentioned in Section 2.1 are done. After that, we computed the ingredients for the final scores step by step. The process will output a score of each sentence pair from 0 to 1. The scores could be used for training purposes, and could be used to determine entailments of test data.

Intuitively, a sentence t_1 is much easier to imply another sentence t_2 if they share more common words. Hence, we determined entailments based on the proportion of shared words in the sentence pair. Obviously, the same concept may not be expressed in exactly the same terms. Hence, we search synonyms between t_1 and t_2 to obtain a better estimate of the ratio. The ratio of words overlapping is defined as follows:

$$\text{Score} = f_{Overlap}(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_2|},$$

where T_1 and T_2 are the set of words after t_1 and t_2 are segmented.

As we mentioned earlier, NEs are expected to be important information in sentences. Based on the ratio of overlapping words, we further checked whether the NEs retrieved from t_2 appear in t_1 simultaneously. If there are NEs from t_2 which are not shown in t_1, we believe that it is much harder to say t_1 entails t_2. The function becomes:

$$\text{Score} = f_{Overlap}(T_1, T_2) - f_{NEExistence}(t_1, t_2),$$

$$f_{NEExistence}(t_1, t_2) = \alpha \times Count(t_1, t_2),$$

where $Count(t_1,t_2)$ returns the numbers of NEs in t_2 that do not appear in t_1 and α is the penalty manually chosen between 0 and 1 based on some experiment results.

t_1: 若望保祿二世是四百五十多年來第一位**非**義大利籍的教宗 (John Paul II is the first non-Italian Pope in the past four hundred and fifty years)

t_2：若望保祿二世是四百五十多年來第一位義大利籍的教宗 (John Paul II is the first Italian Pope in the past four hundred and fifty years)

Figure 7. Negation Word Leads to Opposite Meaning

Sometimes the ratio of overlapping words is extremely high, but t_1 does not really entail t_2. Figure 7 illustrates this problem. Negation words change the meanings of sentences; they often make totally opposite meanings from one to another. Therefore, it is desirable to be capable of identifying negation words. In addition, we add another step to examine whether two sentences share the same number of negation words. Hence, the function is further refined to:

$$\text{Score} = f_{Overlap}(T_1, T_2) - f_{NEExistence}(t_1, t_2) - f_{NegDet}(t_1, t_2),$$

where f_{NegDet} is the penalty that t_1 and t_2 do not have the same number of negation words. The penalty was defined between 0 and 1.

Antonyms also can lead to opposite meanings between sentences. Unlike negation words, we believed that antonyms are much easier to show there were possible opposite meaning and none entailment between two sentences. Hence, we want to decrease the entailment score heavily to indicate this problem. We then modified our function as below:

$$\text{Score} = \frac{f_{Overlap}(T_1,T_2) - f_{NEExistence}(t_1,t_2) - f_{NegDet}(t_1,t_2)}{f_{AntonymDet}(T_1,T_2)},$$

where $f_{\text{AntonymDet}}$ is the penalty, in the range [1, 2], when any some words in t_1 and t_2 are antonyms.

NEs carry important information as we have mentioned. When we began to consider the syntactic structures, we found that NEs' orders played a significant role in recognizing entailments. When the score computed from previous steps is high, two sentences may share many words, suggesting they are about the same events. However, differences in NEs' orders possibly change the meanings of sentences because the subjects and objects may be switched. We tried to retrieve the differences of NEs' orders and decreased entailment score by specified penalty. Moreover, syntactic structures should be considered to avoid the usage of passive sentences in the future. Our function with NEs' orders is improved by:

$$\text{Score} = \frac{f_{Overlap}(T_1, T_2) - f_{NEExistence}(t_1, t_2) - f_{NegDet}(t_1, t_2)}{f_{AntonymDet}(T_1, T_2) \times f_{NEOrder}(t_1, t_2)}$$

where f_{NEOrder} is the penalty whenever the NEs' orders in t_1 are different in t_2 and its range is between 1 and 2.

We have been done many experiments to set up the entailment threshold and penalties. In the next section, we explain another approach of our system.

3.2 Classification Models

To compare with the approach of heuristic functions, we used different combinations of lexical and syntactic features to train classification models to recognize entailments between sentences. Classification algorithms were used to automatically analyze and to learn the generality from data. The trained models were further used to predict the classifications of new instances. In this section, we list the features used to train classification models as shown in Figure 8.

For quick implementation, we extracted features as Section 3.1 described: ratio of overlapping words, counts of named entities, counts of negation words, and synonyms etc. These features were considered to tell how the two sentences shared linguistic information at lexical level. Also, the lengths of sentences were thought as an important feature. Many of the sentences pairs were classified as entailment when the length of t_1 was longer than the length of t_2. Hence, we extracted the lengths of sentences and generated Boolean features to show if the length of t_1 is longer than the length of t_2.

In addition to the lexical level features, syntactic features are considered to show if the two sentences use similar syntactic structures. A sentence structure is similar to another when they have more common patterns in parse trees. Stanford Parser was used to generate parse trees of t_1 and t_2 [8]. We proposed a simple method to compute the syntactic similarity between t_1 and t_2 and to output similarity scores from 0 to 1. Subtrees were retrieved from each parse tree. We calculated the ratio of matching subtrees as follows:

$$\text{ParseTreeRatio}(Tr_1, Tr_2) = \frac{\left|Subtree_{Tr_1} \cap Subtree_{Tr_2}\right|}{\left|Subtree_{Tr_2}\right|},$$

where $Subtree_{Trl}$ is the set of subtrees retrieved from the parse tree of t_1.

Without external data sets, we used development sets to train the classification models. The development sets of traditional and simplified Chinese consist 1321 and 814 sentence pairs individually. C50 and LibSVM were selected to train the classification models. We compared the two algorithms on their 10-fold cross-validation performance with several combinations of features. The models trained by LibSVM provided better performance than the models trained by C50. Hence, we used LibSVM to train the classification models [6]. The models were used to classify BC results. To recognize MC results, we swapped t_1 and t_2 to extract features and classified their BC results. MC results were generated from the two BC results as the inferences shown in Figure 5.

(1) Words' Overlapping Ratio
(2) NE Quantities ($|NE_{t1}|$, $|NE_{t2}|$)
(3) NE_{t2} Match Ratio in t_1
(4) NE_{t2} Not Match Quantities in t_1
(5) Sentence Length (Len_{t1}, Len_{t2})
(6) Sentence Length Comparison
(7) Parse Tree Matching Ratio
(8) Negation Words Quantities ($|Neg_{t1}|$, $|Neg_{t2}|$)
(9) Neg_{t2} Match Ratio in t_1
(10) Synonym Quantities (Synset)
(11) Synonym Ratio

Figure 8. Features

(1) Words' Overlapping Ratio*
(2) NE Quantities ($|NE_{t1}|$, $|NE_{t2}|$)*
(3) NE_{t2} Match Ratio in t_1*
(4) NE_{t2} Not Match Quantities in t_1*
(5) Sentence Length (Len_{t1}, Len_{t2})*
(6) Sentence Length Comparison*
(7) Parse Tree Matching Ratio*
(8) Negation Words Quantities ($|Neg_{t1}|$, $|Neg_{t2}|$)
(9) Neg_{t2} Match Ratio in t_1
(10) Synonym Quantities (Synset)
(11) Synonym Ratio

* (1) – (7) were used to train simplified Chinese model

Figure 9. Features to Train Models

4. Evaluation

In this section, we describe the settings of our systems for formal runs. We also discuss systems performance based on the evaluation results.

We participated in both BC and MC subtasks for traditional and simplified Chinese. There are 881 and 781 sentence pairs without labeling answers in traditional and simplified Chinese testing sets. Our systems are able to output BC results. MC results were inferred by BC results afterwards.

4.1 Settings

For each subtask, we submitted three runs for the testing sets. We set up three different settings for this purpose.

Run 01: For the approach of heuristic functions, we had conducted many experiments to find the threshold and penalties to optimize our performance for the development sets. We did not consider synonyms in this setting.

Run 02: We added the consideration for synonyms to achieve the setting for Run 01. Based on results that we observed in explorative experiments, we would consider two words synonymous if their confidence score is greater than 0.88 (cf. Section 2.2).

Run 03: We trained two separate SVM models for traditional and simplified Chinese. The features to train models were searched for

the highest 10-fold cross-validation accuracy of the development sets. Figure 9 shows the features for training the two models. The features marked with star symbols were used to train simplified Chinese model.

4.2 Formal Run Results

Table 1 and Table 2 show the results of the formal runs of BC and MC subtasks. Because our system development focused on the BC subtasks, we discuss the results of only BC subtasks.

Table 1 shows our performance in the BC subtasks. We achieved better results with the setting of Run 02 than with the setting of Run 01. We used our synonym retrieval method to figure out whether two words are synonyms in the setting of Run 02. The results indicate that synonyms are useful to improve the system performance on recognizing binary relations. Moreover, Run 03 results show the capability of the two models trained by different features. The model of traditional Chinese still had great ability to recognize the entailments between sentences, but the model of simplified Chinese did not receive similar results. As figure 9 shows, negation words and synonyms were not used to train the model of simplified Chinese, because the combination of features did not receive higher cross-validation performance on the development set. Without negation words and synonyms, we think some linguistic information was lost to train recognition models. Hence, the result of Run 03 in simplified Chinese was dropped down sharply.

In terms of the ranked results of the formal runs for both BC and MC and for both traditional and simplified Chinese, we performed relatively good compared with most other systems.

Table 1. Formal Run Results of BC Subtask

	CT	CS
Run 01	65.42	65.71
Run 02	67.07	68.09
Run 03	66.99	57.19

Table 2. Formal Run Results of MC Subtask

	CT	CS
Run 01	42.16	41.82
Run 02	45.15	44.74
Run 03	44.21	34.42

5. Discussions

We explored two approaches to recognize entailments, employing public tools for Chinese segmenter, syntactic parsing, named-entity recognition, and additional semantic resources. Heuristic functions and SVM models which considered different combinations of the linguistic features were proposed and applied to the RITE tasks. Our systems performed relatively well in the formal runs. We achieved the second best scores for the BC subtasks for both traditional and simplified Chinese, and ranked among the top five teams for the MC subtasks. The results suggest the general applicability of the considered features and explored methods.

The proportion of words shared by the sentence pairs formed an important basis in our heuristic functions. It is certainly risky just to rely on words that were literally the same in computing the shared words. Results of some internal evaluations showed that our computing near synonyms with the help of E-HowNet provided instrumental help to the overall performance. The score components that considered the named entities enhanced the performance of our systems, as indicated by the performance of the setting of Run 02. We have not achieved good results using the SVM models, which remains as an item for future work.

Further examination of the judgments of our systems showed the weakness of the current systems. There is room for improving the contributions of negation words and antonyms. Our ability to identify temporal relationships remained to be enhanced. Some forms of automatic or semi-automatic detection and identification of negation expression, antonymous relationships, and time stamps are greatly desirable.

For the formal runs, we adopted SVM models for the machine-learning approach, but did not achieve results that we expected. In some follow-up experiments, we switched to weighted linearly score functions and learned the weights for the selected features. Reasonable and encouraging results were observed, and will be reported in future reports.

Acknowledgements

This work was supported in part by the grants NSC-100-2221-E-004-014- and NSC-101-2221-E-004-018- from the National Science Council, Taiwan.

References

[1] Chinese Dictionary from Ministry of Education. http://dict.revised.moe.edu.tw/ [2013/02/01]

[2] Extended-HowNet. http://ehownet.iis.sinica.edu.tw/ [2013/02/01]

[3] Hideki Shima, Hiroshi Kanayama, C.-W. Lee, C.-J. Lin, Teruko Mitamura, Yusuke Miyao, Shuming Shi and Koichi Takeda. 2011. Overview of NTCIR-9 RITE: Recognizing Inference in TExt, *Proceedings of NTCIR-9 Workshop Meeting*, 291-301.

[4] Ido Dagon, Oren Glickman and Bernardo Magnini, The PASCAL Recognising Textual Entailment Challenge, In Quiñonero-Candela,J.; Dagan, I.; Magnini, B.; d'Alché-Buc, F.(Eds.), *Machine Learning Challenges*. Lecture Notes in Computer Science, Vol. 3944, 177-190, Springer, 2006.

[5] Jianfeng Gao, Mu Li, Andi Wu and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach, *Computational Linguistics,* 31(4).

[6] LibSVM – A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/ [2013/02/01]

[7] Rodney D. Nielsen, Wayne Ward, and James H. Martin. 2009. Recognizing entailment in intelligent tutoring systems. *JOurnal of Natural Language Engineering.* 15(4)*:* 479-501.

[8] Stanford Parser. http://nlp.stanford.edu/software/lex-parser.shtml [2013/02/01]

[9] Stanford Word Segmenter. http://nlp.stanford.edu/software/segmenter.shtml [2013/02/01]

[10] Yotaro Watanabe, Yusuke Miyao, Junta Mizuno, Tomohide Shibata, Hiroshi Kanayama, C.-W. Lee, C.-J. Lin, Shuming Shi, Teruko Mitamura, Noriko Kando, Hideki Shima and Kohichi Takeda. 2013. Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Conference, *Proceedings of NTCIR-10 Conference.*