The WHUTE System in NTCIR-10 RITE Task

Han Ren, Hongmiao Wu
School of Foreign Languages and Literature, Wuhan University
Wuhan 430072, China
{hanren, hmwu}@whu.edu.cn

Chen Lv, Donghong Ji
School of Computer, Wuhan University
Wuhan 430072, China
{chenlv, dhji}@whu.edu.cn

Jing Wan
Center for Study of Language and Information, Wuhan University
Wuhan 30072, China
jennifer_wanj@yahoo.com.cn

ABSTRACT
This paper describes our system of recognizing textual entailment for RITE Traditional and Simplified Chinese subtasks at NTCIR-10. We build a textual entailment recognition framework and implement a system that employs features of three categories, including string, structure and linguistic features, for the recognition. In addition, an entailment transformation approach is leveraged to align text fragments in each pair. We also utilize a cascaded recognition strategy, which first judge entailment or no entailment, and then forward, bidirectional, contradiction or independence relation of each text pair in turn. Official results show that our system achieves a 65.55% MacroF1 performance in Traditional BC subtask, a 45.50% in Traditional MC subtask, a 61.65% in Simplified BC subtask and a 46.79% in Simplified MC subtask. In IR4QA subtasks, our system achieves a 27.33% WorseRanking Top1 accuracy in Traditional subtask and a 18.67% in Simplified subtask.

Keywords
Recognizing Textual Entailment, Binary-Class Subtask, Multi-Class Subtask, Entailment Transformation, Cascaded Entailment Classification

1. INTRODUCTION
Recognizing Textual Entailment(RTE) is a generic framework, by which text inference is able to be viewed as a binary judgment whether one text can be inferred from another. RTE is such a notable research field leveraged in many applications, that well-known evaluation workshops such as TREC and NTCIR hold RTE challenges for exploring and estimating current entailment recognition technologies.

This year, RITE-2 challenge[16] is re-organized by NTCIR-10 evaluation workshop that is the second challenge of the series RITE evaluation. Different with RITE-1, the RITE-2 challenge defines six subtasks: Binary-Class(BC), Multi-Class(MC), ExtraBC, ExtraMC, RITE4QA and OptionalRITE4QA, and types of entailment relations are cut down to four: forward, bidirectional, contradiction and independence relation. We participate in all subtasks and submit two runs for simplified BC and MC subtasks respectively, one run for simplified RITE4QA, Optional RITE4QA, traditional BC, traditional MC, traditional RITE4QA and traditional OptionalRITE4QA subtask respectively.

Since the task definition of RITE-2 is similar with that of RITE-1, the system we implemented in the previous challenge can be easily updated for RITE-2 subtasks. In our updated system, polarity recognition is improved by using antonyms from dictionaries. Being a new part, entailment transformation performs to transform directional and undirectional text fragments in each pair. In addition, a cascaded entailment classification approach including three classifiers is utilized to recognize four types of entailment relations.

Since background knowledge is proved to greatly impact the performance of RTE by many researchers[3], our system employs more knowledge bases such as online dictionaries, lexicons, Wikipedia, PropBank, for a better performance. In addition, those effective features in the pervious system are also employed for the current one. The ablation test estimates performances of these algorithms, features and resources.

The rest of this paper is organized as follows. In section 2, the architecture and workflow of the system are described. Section 3 gives a more detailed explanation for each part of the system, including preprocessing, transformation, all employed features and the cascaded entailment recognition approach. Section 4 gives the experimental results and section 5 gives some discussions about factors that impact performances of our system as well as error analysis. Finally, some conclusions are given in section 6.

2. SYSTEM ARCHITECTURE
The overall architecture of system is shown in Figure 1, which contains a preprocessing model, a transformation model, a feature extraction model and three classifiers. Procedures of the system are described as follows:

1) For each text fragment and hypothesis, a preprocessing procedure is performed, including word segmentation, part-of-speech tagging, named entity recognition, syntactic dependency parsing and semantic role labeling;
2) Texts after preprocessing are aligned through transformation approach, including directional and undirectional terms;
3) In feature extraction, string, structure and linguistic feature vectors are computed according to text pairs;
4) All features are employed to judge entailment or no entailment, and then forward, bidirectional, contradiction or independence through a cascaded classifier.

3. SYSTEM DESCRIPTION
3.1 Preprocessing
The preprocessing procedure includes word segmentation, Part-Of-Speech(POS) tagging, named entity recognition, syntactic parsing and shallow semantic paring.

Initially, the text and the hypothesis for each pair are segmented by Stanford Chinese Word Segmenter and tagged by Stanford

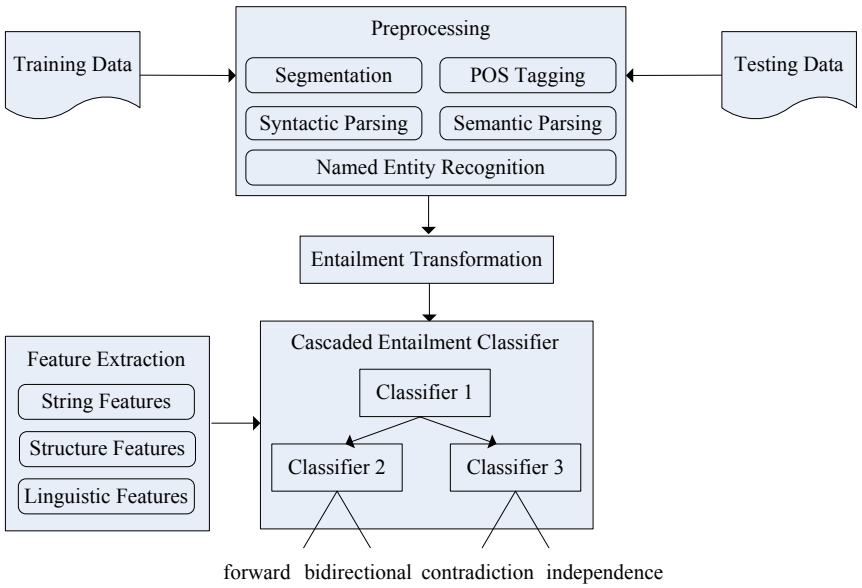

Figure 1. System architecture

POS Tagger[1]. Both the tools are implemented by Java so that they are easily invoked by our system. For named entities, we only extract personal names, locations, organizations and temporal expressions by using ICTCLAS[2], a free Chinese POS tagger and NE recognizer, for the recognition. In addition, we utilize a numeral normalization tool implemented in RITE-1[11], transforming the temporal and Chinese numeral expressions to Arabic numerals.

The syntactic and semantic parsing model follows our system in CoNLL2009[12], which labels syntactic and semantic dependency relations of words, since shallow syntactic and semantic relations are more flexible and precise. The annotation standard is identical with the definition in CoNLL2009, with 30 tags for the syntactic dependents and 25 tags for the semantic roles.

3.2 Entailment Transformation

Transformation is another major strategy for entailment recognition[4, 7, 16] in comparison with classification. and frequently adopted in alignment and syntactic matching. Essentially, transformation is to search for a sequence of rule sets or other resources, i.e., synonyms or hypernym-hyponym knowledge bases, that turns one of the input expressions, i.e., lexical or syntactic representation, to the other.

In our system, transformation proceeds before classification. More specifically, after preprocessing, text fragments in t_1 in each pair are estimated whether or not a corresponding transformable part exists in t_2. If so, text fragments in t_1 are replaced by the corresponding part in t_2. After that, the transformed pair are trained and predicted by entailment classifier.

3.2.1 Directional Transformation

Directional transformation is a synonymous meaning alternation, which includes transformation of synonymous words and some named entities. For word level transformation, we utilize an online resource, CIBA HANYU [3], to search and acquire transformable words. This resource is an online dictionary including most common Chinese words and their synonymous or antonymous words. The process is simple: we search synonymous words for a word w_1 in t_1 in a pair, and then we search if any synonymous word is also in t_2. If such a word w_2 is in t_2, we use w_2 to replace w_1 in t_1. The process is an iterative one until every word in t_1 is visited. The transformation process for antonymous words is similar, except that the negative modifier should be replaced together. Take the pair 243 in simplified subtask test data as an example, t_1 includes "没有接受" not accept that includes a word "接受" accept and a negative word "没有" not, while t2 includes "拒绝" refuse which is one of the antonymous words of "接受" accept. Thus "没有接受" not accept is transformed to "拒绝" refuse so that this two text fragments are identical.

Acquiring those named entities with the same meaning, such as "哈利法塔" Burj Khalifa Tower and "杜拜塔" Dubai Tower, is also important for entailment recognition. Here we utilize a Wikipedia based method to extract synonymous named entities. In fact, some synonymous named entities are easy to find through two ways: one is Wikipedia redirection[13], the other is some expressions such as "also known as" or brackets after named entities. For Wikipedia redirection, we search named entities in each pair to find redirect terms. On the other hand, heuristic rules are built to extract terms in brackets after named entities appeared in the texts in each pair.

3.2.2 Undirectional Transformation

Undirectional transformation is an asymmetric meaning alternation from t_1 to t_2. We consider hypernym, hyponym and geographic information in our system. As to hypernym and hyponum relations, a lexicon TongYiCi CiLin is utilized. The procedure is described as follows: first we search every word in t_1 from the lexicon; if it is found, the further research is proceeded that whether its hypernymous words are also appeared in t_2; if so,

[1] http://nlp.stanford.edu/software/
[2] http://ictclas.org/
[3] http://hanyu.iciba.com/

the word in t_1 will be replace by the hypernymous word in t_2. On the other hand, geographic information includes geographic hypernyms and hyponyms that also impacts the performance of entailment recognition. An example is the pair 237 in simplified subtask, in which t_1 is "上海获得世博会主办权" Shanghai is awarded the right to host the World Expo, and t_2 is "中国获得世博会主办权" Shanghai is awarded the right to host the World Expo. Apparently, "上海" Shanghai belongs to "中国" China. To acquire geographic entailment relation, we utilize a geographic knowledge built before and extract rules according to geographic hypernyms and hyponyms. If t_1 contains a geographic term and t_2 contains its hypernym one, the former term is replaced by the latter one.

3.3 Cascaded Entailment Classifier

The entailment type forward is supposed to be a directional relation. Unfortunately, few features in our system are undirectional, which makes little impact in classifying forward relation. For recognizing forward relation by a single classifier, directed features such as word overlap and sub tree overlap feature are duplicated, considering t_1 is the text and t_2 is the hypothesis, and then t_2 the hypothesis and t_1 the text. Intuitively, if a feature gets a high score under the condition that t_1 is the text and t_2 is the hypothesis, whereas it gets a low one under the condition that t_2 is the text and t_1 is the hypothesis, it probably indicates that t_1 entails t_2 and not vice versa.

Our first run in simplified MC subtasks employs directed and undirected features for classification. However, the performance of the experiment is no satisfying, since the classifier judges not only entailment class but also entailment direction at the same time. Alternately, a cascaded entailment recognition strategy is utilized, inspired by the approach in our RITE-1 system[11], that is, a text pair is first judged entailment or no entailment, and then forward, bidirectional, contradiction or independence. More specifically, for each pair(t_1, t_2), a bi-categorization classifier is employed to judge whether t_1 entails t_2. Thus the problem is equivalent with that of the 2-way judgment in BC subtask. After that, a logical decision is made, where (t_1, t_2) has a forward entailment relation or not. If t_1 entails t_2, the second classifier is employed to judge whether t_2 entails t_1 or not. If t_1 does not entail t_2, the third classifier is employed to judge whether there is a contradiction or independence relation between t_1 and t_2. Finally, the output is given, that the relation of the pair(t_1, t_2) is forward if t_1 entails t_2 but t_2 does not entail t_1, bidirectional if t_1 entails t_2 and t_2 entails t_1, or contradiction or independence according to the output of the third classifier.

In the cascaded recognition approach, three classifiers are trained, and the features employed in the prior single stage system are still utilized except for those duplicated ones. For the purpose of undirectional entailment recognition, each pair in training data, which at least has one entailment relation from one text to another, are split into two entailment pairs. More specifically, if t_1 and t_2 in a pair have the relation of bidirection, then two entailment pairs $t_1 \rightarrow t_2$ and $t_2 \rightarrow t_1$ are generated automatically. Using this method, training data are pseudo-expanded that benefits the improvement of system performance.

3.4 Features

There are three types of features employed in our system: string, structure and linguistic features, among which most of them are similar with those employed in our prior system in RITE-1.

3.4.1 String Features

The ideal of string feature is simple: *T* entails *H* if most of most of the surface strings in *T* is identical with that of in *H*. In fact, string features are the dispensable part for most classification-based systems in the series RTE[1, 4, 5, 10, 15] and RITE[2, 6, 8, 9, 11] challenges.

N-gram Overlap This character-based feature computes how similar the hypothesis is to the text by comparing how many of the same n-grams appear in *H* of each pair. In our system, bigram and trigram are taken into account.

Word Overlap This feature is similar with the N-gram Overlap, except that character is replaced by word. Recall that words in *T* are replaced according to those synonymous words in *H* by transformation process.

Matching Coefficient Different with Word Overlap, this feature considers $|words(T) \cap words(H)|$, namely how many of the same words appear in both *T* and *H*, where $words(T)$ and $words(H)$ are the word sets of the text and the hypothesis in each pair.

LCS Similarity This feature in our system estimates the similarity between the longest common substring of *T* and *H* in each pair, and the shorter one in two of them. It is computed as below:

$$LCS\ Similarity = \frac{LCS(T,H)}{min\{words(T), words(H)\}} \quad (1)$$

Cosine Similarity This feature builds the word vectors of *T* and *H* in each pair, and computes its cosine similarity.

Levenshtein Distance Also known as edit distance, this distance considers the minimum number of transform operations from one string to another, where an operation refers to an insertion, deletion or substitution of a single unit, which in our system is a Chinese character or a word.

Length Ratio This feature considers the length ratio of the text snippets in each pair. The length is the total number of the unigrams in each text snippet.

Numeral Coverage This feature gives a boolean value; it is true if all the numerals in the hypothesis(if have) also appear in the text, or false otherwise.

Common String Overlap This feature considers the ratio of common substrings between *T* and *H*:

$$CSOverlap = \frac{\{\text{length of all common strings in } T \text{ and } H\}}{\{\text{length of } H\}} \quad (2)$$

3.4.2 Structure features

Four syntactic and semantic features are employed in our system, aiming at estimating similarity of the dependency structures between the text and the hypothesis in each pair.

Unlabeled Sub Tree Overlap This features computes the ratio of the same sub trees in the text and the hypothesis, as described in the following formula. Each sub tree has a head and one of its dependents derived from the syntactic dependency tree. Two sub

trees are viewed identical if they have the same heads and the dependents.

$$UST\ Overlap = \frac{subtree(T) \cap subtree(H)}{subtree(H)} \quad (3)$$

Labeled Sub Tree Overlap Similar with Unlabeled Sub Tree Overlap, this feature computes how similar the hypothesis is to the text by comparing the ratio of the same sub trees appear in *H*, except that the dependency relations(or classes) are also taken into account in sub trees.

Partial Sub Tree Overlap In comparison with the above features, this feature is more relax, taking partial matching of the sub trees into account. That is, two sub trees are viewed partially identical if they have the same heads or the dependents. In order to differ full matching and partial matching of sub trees, we set a weighting value, which equals 1 if sub trees are full matched, 0.5 if partially matched and 0 if no matched, following with the experiments in RITE-1[11].

Predicate Argument Overlap This features computes the ratio of the same predicate-argument pairs in the text and the hypothesis. Each predicate-argument pair has a predicate and one of its arguments(if have) derived from the semantic parsing result. Two predicate-argument pairs are viewed identical if they have the same heads and the corresponding dependents. The feature is computed as below:

$$PA\ Overlap = \frac{pred\text{-}arg(T) \cap pred\text{-}arg(H)}{pred\text{-}arg(H)} \quad (4)$$

3.4.3 Linguistic Features

Linguistic features are employed to estimate the relevance between *T* and *H* from a linguistic view.

Named Entity Coverage This feature gives a boolean value; it is true if all the named entities in the hypothesis(if have) also appear in the text, or false otherwise. Note that some synonymous entities in *H* which are also in *T* are replaced by entailment transformation.

Polarity This feature gives a boolean value; it is true if the overall polarity of *T* and *H* are identical, or false otherwise. The overall polarity of a text is a multiplicative value, which is updated by multiple -1 for every negative word in the text.

4. EXPERIMENTAL RESULTS

There are six subtasks, including Binary Class(BC), Multi Class(MC), Extra Binary Class(ExtraBC), Extra Multi Class(ExtraMC), RITE4QA and Optional RITE4QA, for both traditional Chinese and simplified Chinese subtasks in RITE-2. We participated in subtasks of BC, MC, ExtraBC, ExtraMC and RITE4QA of each language. Since only the results of BC, MC and RITE4QA of each language are released, this section reports the official RITE-2 results of these four subtasks. In addition, an ablation test is also reported.

4.1 BC Subtask

For the simplified BC subtask, we submit two runs: RITE2-WHUTE-CS-BC-01 and RITE2-WHUTE-CS-BC-02. Since our aim in this subtask is to estimate the impact of the structure information to the entailment recognition, the experiments are set up as follows: the first run employs all the features in section 3.4 except for the structure ones for the entailment classifier, while the second run appends the structure features based on the first run. Table 1 shows the official results of these two runs, where Y denotes entailment relation, N non entailment relation, Prec. precision and Rec. recall.

Table 1. Official results of simplified BC subtask

	WHUTE-CS-BC-01	WHUTE-CS-BC-02
MacroF1	0.5820	0.6165
Accuracy	0.6479	0.6658
Y-F1	0.7479	0.7540
Y-Prec.	0.6099	0.6260
Y-Rec.	0.9668	0.9479
N-F1	0.4161	0.4790
N-Prec.	0.8750	0.8451
N-Rec.	0.2730	0.3343

The second run achieves a better performance in most cases, except for Y-Rec and N-Prec, as shown in Table 1. More specifically, for entailment relation, run2 achieves a 1.61% performance increase of precision and a 1.89% decrease of recall in comparison with run1; for non entailment relation, the results of run2 show a 2.99% decrease of precision and a 6.13% increase of recall, in comparison with run1. In spite of this, our system achieves an increasing performance 1.79% of accuracy and 3.45% of MacroF1 metric.

Table 2. Official results of traditional BC subtask

	WHUTE-CT-BC-01
MacroF1	0.6555
Accuracy	0.6617[4]
Y-F1	0.7020
Y-Prec.	0.6737
Y-Rec.	0.7328
N-F1	0.6089
N-Prec.	0.6444
N-Rec.	0.5771

For the traditional BC subtask, we submit one run: WHUTE-CT-BC-01, which utilizes the same approach with the second run of the simplified BC subtask. Table 2 shows the official results.

4.2 MC Subtask

For the simplified MC subtask, we submit two runs: RITE2-WHUTE-CS-MC-01 and RITE2-WHUTE-CS-MC-02. The first run utilizes a unitary recognition approach, namely judges the entailment class directly by using a single classifier. The second run utilizes the cascaded recognition approach introduced in section 3.3, where three classifiers are trained for two stage recognition. Table 3 shows the official results, where F denotes forward entailment relation, B bidirectional relation, C contradiction relation and I independence relation.

The second run achieves better performances in most cases, while the first run gains better recall performances in most entailment categories. For forward relation, there are an increasing 1.34% precision and a decreasing 1.44% recall of run2 in comparison with run1; for bidirectional one, the performance has a 2.54%

[4] The accuracy value of the traditional BC subtask is not provided by the official evaluation.

drop of precision and a 4.14% raise of recall; for contradiction one, the performance has a 6.06% drop of precision and a 7.54% drop of recall; for independence one, the precision raise a 2.51% of precision and a 7.9% of recall.

Table 3. Official results of simplified MC subtask

	WHUTE-CS-MC-01	WHUTE-CS-MC-02
MacroF1	0.4679	0.4653
Accuracy	0.5480	0.5659
F-F1	0.6436	0.6509
F-Prec.	0.4990	0.5124
F-Rec.	0.9061	0.8917
B-F1	0.6154	0.6225
B-Prec.	0.6241	0.5987
B-Rec.	0.6069	0.6483
C-F1	0.1871	0.0826
C-Prec.	0.3939	0.3333
C-Rec.	0.1226	0.0472
I-F1	0.4258	0.5053
I-Prec.	0.7308	0.7559
I-Rec.	0.3004	0.3794

Table 4. Official results of traditional MC subtask

	WHUTE-CT-MC-01
MacroF1	0.4550
Accuracy	0.5516[5]
F-F1	0.6706
F-Prec.	0.5436
F-Rec.	0.8750
B-F1	0.5886
B-Prec.	0.5636
B-Rec.	0.6159
C-F1	0.1208
C-Prec.	0.2571
C-Rec.	0.0789
I-F1	0.4399
I-Prec.	0.6340
I-Rec.	0.3368

Table 5. Official results of simplified RITE4QA subtask

			WHUTE-CS-RITE4QA-01
Worse Ranking	R	Top1	0.1867
		MRR	0.2759
		Top5	0.4333
	R+U	Top1	0.2200
		MRR	0.3367
		Top5	0.5400
Better Ranking	R	Top1	0.1867
		MRR	0.2764
		Top5	0.4333
	R+U	Top1	0.2267
		MRR	0.3406
		Top5	0.5400

[5] The accuracy value of the traditional MC subtask is not provided by the official evaluation.

For the traditional MC subtask, we submit one run: WHUTE-CT-MC-01, which utilizes the same approach with the second run of the simplified MC subtask. Table 4 shows the official results.

4.3 RITE4QA Subtask

For the simplified RITE4QA subtask, we submit one run: RITE2-WHUTE-CS-RITE4QA-01. Table 5 shows the official results, where BetterRanking is produced from a good QA system, and WorseRanking is the reverse ranking of BetterRanking.

Table 6. Official results of traditional RITE4QA subtask

			WHUTE-CT-RITE4QA-01
Worse Ranking	R	Top1	0.2733
		MRR	0.3457
		Top5	0.4667
	R+U	Top1	0.3067
		MRR	0.3876
		Top5	0.5267
Better Ranking	R	Top1	0.2667
		MRR	0.3429
		Top5	0.4667
	R+U	Top1	0.3000
		MRR	0.3848
		Top5	0.5267

For the traditional RITE4QA subtask, we submit one run: RITE2-WHUTE-CT-RITE4QA-01. Table 6 shows the official results.

4.4 Ablation Test

In RITE-1, ablation test is suggested by the organizer[14], aiming at estimating the contribution of each resource(or feature) to participants' system performances. In RITE-2, we make the experiments by removing one feature at one time in the run2 for BC subtask. Due to the time limitation, some features that greatly impact the performance in RITE-1 are selected[11]. Table 7 shows the results of the ablation test.

Table 7. Results of ablation test

System Description	Accuracy
Baseline	0.6658
Without Bigram Overlap	0.6645
Without Cosine Similarity	0.6645
Without LCS Similarity	0.6569
Without Character Levenshtein	0.6581
Word Levenshtein Distance	0.6645
Without Length Ratio	0.6633
Without Trigram Overlap	0.6645
Without Unlabeled Sub Tree Overlap	0.6529
Partial Sub Tree Overlap	0.6619
Without Named Entity Coverage	0.6415
Without Numeral Coverage	0.6517
Without Common String Overlap	0.6364
Without Transformation	0.6517
Without Polarity Judgment	0.6619

Each of the former 12 results shows the accuracy when removing only one feature at one time from the entailment classifier. The latter two results show the accuracy when removing the process of transformation and the polarity judgment respectively.

5. DISCUSSION

In this section, we analyze the performance of our system in every subtask and some typical cases that are judged incorrectly by our system. Also, some directions for further improvement are given.

5.1 System Performance

As to the simplified BC subtask, the usage of structure features improves the performance of our system, especially for non entailment recall. As a matter of fact, structure matching leads an increasing performance of precision as well as a decreasing performance of recall for the judgment of entailment category, mainly because the structure features makes the judgment of entailment category more rigid than the string ones. On the other hand, as a general comparison, the increasing rate of the overall performance is less than that of the performance in RITE-1 subtask[11], partly because the syntactic and semantic relations in RITE-2 dataset are more complex; in other words, two text fragments having the same meaning are less identical or similar with their syntactic or semantic structures.

In the experiments of simplified MC subtask, contradiction relation judgment of run2 outperforms that of run1; alternately, the performance of independence relation judgment greatly increases. As a matter of fact, those false contradiction judgments of pairs are derived from the false bi-categorization of entailment against non entailment relation, since the single classifier in run1 utilizes quite a few bidirectional features, i.e., string overlap and structure overlap, whereas most features employed in the first stage of the cascaded entailment classification are directional ones. In other words, less features result in lower performance. Despite this, the system still achieves better F1 performances for forward, bidirectional and independence relations. It indicates that: 1)the cascaded classifier is helpful in recognizing most entailment relations; 2)contradiction relation is more suitable to be judged in the first stage, since pairs of entailment and contradiction are more similar except for some polarity words or phrase; 3)more features should be employed by the first classifier of the cascaded entailment classification for a better performance of contradiction judgment.

As to the traditional and simplified RITE4QA subtask, although only one run for each subtask is submitted and the gold standard is not provided by the organizer, some discussions still can be made: since the testing data of RITE4QA is derived from a real question answering dataset, each text fragment t_2(hypothesis, actually is answer in QA dataset) is more complex than those in other subtasks, which greatly impact the entailment classification in comparison with other subtasks. For a better performance, more precise features or deep semantic relation acquisition should be considered.

As shown in Table 7, there are five factors that impact the system performance more than others in the ablation test: Unlabeled Sub Tree Overlap, Named Entity Coverage, Numeral Coverage, Common String Overlap and Transformation process. It indicates that: 1)the impact of structure and linguistic information for the system performance is more than that of other features; 2)since the Common String Overlap feature depends on the transformation of synonymous phrases, the transformation approach is also important for the entailment recognition. Take the pair 304 in BC test data as an example, "巴拉克·欧巴马" Barack Obama in t_1 is another synonymous expression with "奥巴马" Obama in t_2, but the judgment will be false if the first Chinese name is unable to be transformed or aligned with the second name. Another example is the pair 237, which contains a geographic entailment relation, that is, "上海" Shanghai belongs to "中国" China. Considering that geographic entailment is a directional entailment relation, the directional transformation is supposed to be leveraged. Therefore, as a direction, the transformation of words and phrases are expected to be further improved for a better performance.

5.2 Error Analysis

This subsection shows major error types with examples in the following. For the convenience of case explanation, text snippets are shown instead of full texts for some examples.

Take a close view to the error cases in BC and MC subtasks, most of them are due to false contradiction judgment. For example, the pair 113 in simplified MC test data is a contradiction one:

(1) t_1:目前还没有证实流感疫苗可以预防禽流感. It is not confirmed that influenza vaccine helps prevent avian influenza.
t_2:打流感疫苗根本没有预防禽流感的效用. Injecting influenza vaccine is unhelpful for preventing avian influenza.

Apparently, most of the words in t_1 and t_2 are identical. Although in t_1, there is a negative word "没有" no, it also appears in t_2. Thus the system makes the false judgment of forward relation in this case. As a matter of fact, "证实有效" confirm helpful means something is helpful, whereas "没有证实有效" not confirm helpful does not mean something is unhelpful, but maybe or may not helpful. More specifically, t_1 contains a logical relation(help or not) between the medicine and the effect, while t_2 .only makes the negative statement that the medicine is unhelpful. Essentially, this error belong to false judgment of phrase entailment. Another example of pair 347 is:

(2) t_1:周杰伦是家中的独子. Jay Chou is the only child in his family.
t_2:周杰伦有 2 个哥哥 1 个姐姐. Jay Chou has two brothers and a sister.

This pair is falsely judged as independence, since few words and structures in the two texts are identical. In fact, the word "独子" only child in t_1 entails that Jay Chou does not have any brother or sister, hence the two texts are contradictory. Apparently, numeral analysis fails in this case so that the false judgment is made.

The third type of errors comes from deficient inferable transformation. For example, the pair 451 is a contradition one:

(3) t_1:狼是社会性的猎食动物. Wolfs are social predatory animals.
t_2:狼单独活动. Wolfs act alone.

In t_1, "社会性" social means gregarious, while a gregarious animal acts collectively, which contains a negative meaning against "单独活动" act alone. The true judgment can be made if

there are sufficient background knowledge and effective transformation process.

Another quite a few errors are thanks to false independence judgment. For example, the pair 190 in traditional MC test data is an independence one:

(4) t_1: 國際油價可望回跌至每桶 20 至 25 美元之間. Gas prices are expected to drop to 20 to 25 dollars a barrel. t_2: 國際油價應該在每桶 20 到 25 美元之間. Gas prices are supposed to be 20 to 25 dollars a barrel.

In this case, most words including the numbers and structures in the two texts are identical, thus the system makes a false judgment that t_1 entails t_2. Therefore, to improve the performance of independence judgment, more features discriminate entailment against independence should be employed.

6. CONCLUSION

In this paper, we describe our system for RITE-2 subtask at NTCIR-10. Based on the prior system in RITE-1, we improve the polarity recognition by using antonyms from dictionaries. We also build a new part, entailment transformation, to transform directional and undirectional text fragments in each pair. In addition, a modified cascaded entailment classification approach including three classifiers is utilized to recognize four types of entailment relations. For a better performance, more knowledge bases compared to the prior system are employed.

We also notice very low performances in recognizing contradiction, and the main reasons lies in: 1)contradiction relation is more suitable to be judged in the first stage, since pairs of entailment and contradiction are more similar except for some polarity words or phrase; 2)the system fails to acquire complex semantic relations in texts, hence the complex entailment cases are not able to be truly judged. As a direction, more complex semantic entailment relations such as case alternation and disagree acquisition should be further studied.

7. ACKNOWLEDGMENTS

This work is supported by Natural Science Foundation of China(Grant Nos. 61070082, 61173062 and 61133012).

8. REFERENCES

[1] Agichtein, E., Askew, W. and Liu, Y. Combining Lexical, Syntactic, and Semantic Evidence For Textual Entailment Classification. *In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2008.

[2] Akiba, Y., Taira, H. and Fujita, S. NTTCS Textual Entailment Recognition System for NTCIR-9 RITE. *In Proceedings of the 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[3] Androutsopoulos, I. and Malakasiotis, P. 2010. A Survey of Paraphrasing and Textual Entailment Methods. Journal of Artificial Intelligence Research 38: 135-187.

[4] Bar-Haim, R., Berant, J., Dagan, I., Greental, I., Mirkin, s., Shnarch, E. and Szpektor, I. Efficient Semantic Deduction and Approximate Matching over Compact Parse Forests. *In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2008.

[5] Galanis, D. and Malakasiotis, P. AUEB at TAC 2008. *In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2008.

[6] Huang, H.-H., Chang, K.-C., II, J. M. C. H. and Chen, H.-H. NTU Textual Entailment System for NTCIR 9 RITE Task. *In Proceedings of The 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[7] Iftene, A. UAIC Participation at RTE4. *In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2008.

[8] Pakray, P., Neogi, S., Bandyopadhyay, S. and Gelbukh, A. A Textual Entailment System using Web based Machine Translation System. *In Proceedings of The 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[9] Pham, Q. N. M., Nguyen, L. M. and Shimazu, A. A Machine Learning based Textual Entailment Recognition System of JAIST Team for NTCIR9 RITE. *In Proceedings of the 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[10] Ren, H., Ji, D. and Wan, J. WHU at TAC 2009: A Tri-categorization Approach to Textual Entailment Recognition. *In proceedings of the Fifth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2009.

[11] Ren, H., Ji, D. and Wan, J. The WHUTE System in NTCIR-9 RITE Task. *In Proceedings of the 8th NTCIR workshop meeting*. Tokyo, Japan, 2011.

[12] Ren, H., Ji, D., Wan, J. and Zhang, M. Parsing Syntactic and Semantic Dependencies for Multiple Languages with A Pipeline Approach. *In Proceedings of the 13th Conference on Computational Natural Language Learning*. Boulder, Colorado, USA, 2009.

[13] Shima, H., Li, Y., Orii, N. and Mitamura, T. LTI's Textual Entailment Recognizer System at NTCIR-9 RITE. *In Proceedings of the 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[14] Shima, H., Kanayama, H., Lee, C.-W., Lin, C.-J., Mitamura, T., Miyao, Y., Shi, S. and Takeda, K. Overview of NTCIR-9 RITE: Recognizing Inference in TExt. *In Proceedings of The 9th NTCIR Workshop*. Tokyo, Japan, 2011.

[15] Wang, R. and Neumann, G. A Divide-and-Conquer Strategy for Recognizing Textual Entailment. *In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA, 2008.

[16] Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., Lin, C.-J., Shi, S., Mitamura, T., Kando, N., Shima, H. and Takeda, K. Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Workshop. *In Proceedings of the 10th NTCIR Conference*. Tokyo, Japan, 2013.