ICRC_HITSZ at RITE: Leveraging Multiple Classifiers Voting for Textual Entailment Recognition

Yaoyun Zhang, Jun Xu,* Chenlong Liu, Xiaolong Wang, Ruifeng Xu,
Qingcai Chen, Xuan Wang, Yongshuai Hou and Buzhou Tang
Key Laboratory of Network Oriented Intelligent Computation
Department of Computer Science and Technology
Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, 518055, P.R. China
zhangyy@cs.hitsz.edu.cn hit.xujun@gamil.com

ABSTRACT

The NTCIR-9 RITE challenge is a generic benchmark task that evaluates systems' ability to automatically detect textual entailment, paraphrase and contradiction. This paper describes the ICRC_HITSZ system for RITE. We participate in the binary-class (BC), the multi-class (MC) and the RITE4QA subtask. More specifically, we build textual entailment recognition models for the MC subtask. The predicted multiple class labels are then mapped into Yes/No labels for the BC and RITE4QA subtasks. Different linguistic level features are extracted by using hybrid NLP resources and tools. Based on the hierarchical relations between the labels of the MC subtask, three different classification strategies are designed. Multiple machine learning methods are employed for each strategy. On the assumption that classifiers built from different classification strategies are complementary to each other, so are the different machine learning methods. The final classifier is built with a cascade voting. Evaluation results show that the voting strategies are effective, with the highest performance ranked at the fourth place in terms of accuracy, and at the second place in terms of participant groups in both tasks.

Keywords

textual entailment, textual entailment recognition, multi-classifiers voting

1. INTRODUCTION

In recent years, research on textual entailment (TE) has drawn increasingly attention, since it has many applications such as question answering, multi-document summarization, text generation, and machine translation etc. Generally, TE researches can be categorized into three groups: extraction, generation and recognition [1].The main target of textual entailment recognition (TER) is to determine the directional relationship between two text fragments/expressions, entailment or no. Commonly used methods for TER include machine learning(ML-) based, similarity-based, decoding-based and, logic-based, etc [1]. The ML-based approach is most popular, since it has the ability to combine various features, such as multiple similarity measures, and even the predictions from other TER methods.

Different from previous text entailment tracks [2], the NTCIR-9 RITE challenge [6] proposes a new task which recognize the directional relationship between two sentences. The entailing and entailed texts are termed as text (t_1) and hypothesis (t_2), respectively. The relationship of a text pair $< t_1, t_2 >$ is either entailment in three types, namely forward, reverse and bi-direction; or not entailment in two types, namely contradiction and independence. RITE requires participant systems to predict whether there is an entailment (i.e., the BC subtask) and what type it is (i.e., the MC subtask).

This paper presents the ICRC_HITSZ system in the NTCIR-9 RITE challenge. We participate in the binary-class (BC), the multi-class (MC) and the RITE4QA subtask on both simplified Chinese (CS) and traditional Chinese (CT) sides. More specifically, we build textual entailment recognition models for the MC subtask. The predicted multiple class labels are then mapped into Y/N classes for the BC and RITE4QA subtasks. Different linguistic level features are extracted by using hybrid NLP resources and tools, including EDIT-based features (edit-distance features and similarity-based features), directional entailment features and contradiction features. Based on the hierarchical relations between the labels of the MC subtask, we propose three different problem representation strategies for classification, namely, a five-class recognition problem, a five binary-class recognition problems and a two-dimensional hierarchical recognition problem. Multiple machine learning methods are then employed for each strategy. On the assumption that classifiers built from different classification strategies are complementary to each other, so are the different machine learning methods. The final classifier is built with a cascade voting. Evaluation results show that the voting strategies are effective, with the highest performance ranked at the fourth place in terms of accuracy, and at the second place in terms of participant groups in both tasks.

The next sections are arranged as follows: section 2 describes the features and algorithms employed in our system

*Corresponding Author

in detail; section 3 presents the experimental results and discussion and section 4 concludes the paper.

2. SYSTEM DESCRIPTION

Previous experiments show that for the textual entailment recognition task, using the models originally designed for the multiple-class recognition to solve the binary-class recognition may achieve better performance than models designed specifically for the binary-class recognition [2]. Therefore, in this study, we focus on the MC subtask, and apply the built model directly on the BC and RITE4QA subtasks.

2.1 System Architecture

Fig.1 shows the architecture of our entailment system. The main modules in this system are described as follows, respectively:

2.1.1 Preprocessing Module

This module uses hybrid NLP resources and tools for sentence paring and NE recognition. Each <t_1, t_2> pairs are first preprocessed by the LTP tool [3], for word segmentation, POS tagging, dependency syntactic parsing and named entity (NE) recognition. The recognized NEs include people, institutions, times, locations and numbers. However, our observation shows that many terminologies and person names are not well recognized by LTP, which are split into several words in the segmentation procedure. Since NE plays a critical role in entailment recognition [2], we extract the page titles of Chinese Wikipedia [1] to expand the terminology lexicon of LTP, to improve the coverage the NE recognition. Additionally, a number/time normalization module is deployed to unify various formations of numbers/times. It can also conduct comparisons between numbers/times and judge the entailment relations between them.

2.1.2 Resource Pools

We introduce synonym, antonym and hyponym relations and positive and negative lexicons to produce lexical features for similarity computation and contradiction detection. Firstly, a synonym list is generated consisting of word pairs with similarity score greater than 0.8(empirical thershold). The similarity score is computed based on Hownet [2] API. Next, an antonym list is also extracted from Hownet. This list is further expanded by merging with about 11,000 manually collected antonym lists [3]. Finally, a module for judging hyponym-relation between two words is built according to the sememe hierarchy in Hownet.

Moreover, two lexicons consisting of positive words and negative words, respectively, are applied to judge the polarity of the statement. These lexicons are originally collected for sentiment judgments [7].

2.1.3 Feature Sets

In this study, three types of features are designed. The most obvious difference from previous works is that we extract features for recognizing the directional entailment relations.

EDIT-based Features: The open source package EDIT [4] is employed to generate similarity related features. EDIT is a general-purpose tool, which implements a collection of algorithms and provides a configurable framework to quickly set up a working environment for recognizing textual entailment. The two edit distance based algorithms, (i.e., the token edit distance and syntactic tree edit distance), and the five similarity algorithms, (i.e., word overlap, Jaro-Winkler distance, cosine similarity, longest common subsequence and Jaccard coefficient) provided by EDIT are used to compute the similarity score between t_1 and t_2, respectively. Similar to the paper [5], three types of presentations of t_1 and t_2, namely word tokens, POS tagging and Subject-Verb-Object structures (Sub_V_Obj) extracted from the dependency parsing results are used as the input of EDIT, respectively. We also use the windowing function in EDIT to accommodate the length difference between t_1 and t_2. Since EDIT can only differentiate between two classes: entailment/non entailment, the labels of the input instances are converted

Table 1: Description list of directional entailment features

Description
Proportion of t_1 to t_2/t_2 to t_1
Sentence length in terms of word numbers Number of equal NEs Number of equal content words Number of equal nouns Number of equal numbers Number of equal times Number of equal locations Numbers of equal Sub_Verb_Obj Structures
NE' s Existence in t_1 or t_2
Numbers exist in t_1/t_2, but not in t_2/t_1 Times exist in t_1/t_2, but not in t_2/t_1 Locations exist in t_1/t_2, but not in t_2/t_1
Entailment of different linguistic granularity
Words in t_1/t_2 are hyponyms of words in t_2/t_1 If two words, A and B, are the same, whether A/B' s modifier is the substring of the other' Whether a number from t_1/t_2 can be entailed by a number from t_2/t_1 Whether a time from t_1/t_2 can be entailed by a time from t_2/t_1 Whether a location from t_1/t_2 can be entailed by a location from t_2/t_1 Whether a person from t_1/t_2 can be entailed by a person from t_2/t_1 Whether a Sub_V_Obj Structure from t_1/t_2 can be entailed by a Sub_V_Obj Structure from t_2/t_1 Whether a Sub_V_Obj Structure which has dependency relations with a NE from t_1/t_2 can be entailed by a Sub_V_Obj Structure which has dependency relations with the same NE from t_2/t_1
Definitional Feature
In t_1/t_2, A is the attribute modifier of B, and t_2/t_1 can be considered as representing a B is_a A relation. A and B can be a word or a phrase. The is_a relations are recognized by matching several simple syntactic patterns. Both t_1 and t_2 can be considered as representing an A is_a B relation.

[1] http://download.wikimedia.org/zhwiki
[2] http://www.keenage.com
[3] http://fyc.5156edu.com/

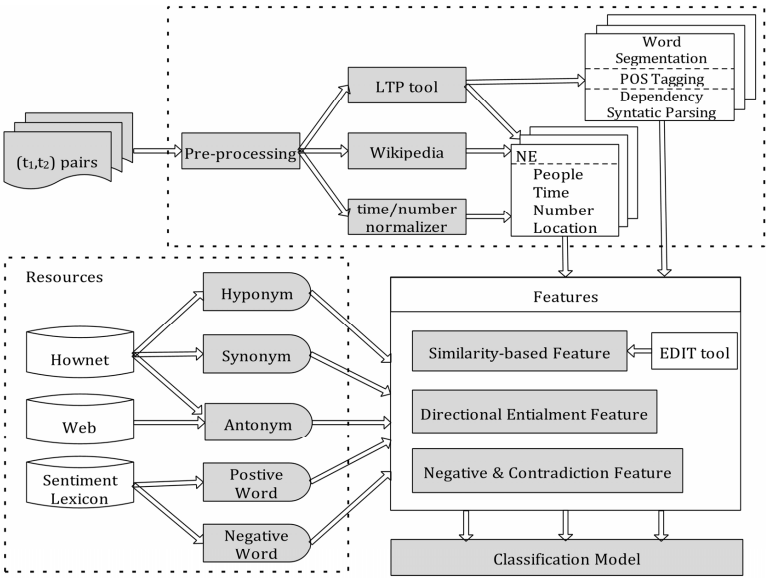

Figure 1: Entailment System Architecture of ICRC_HITSZ.

into these two classes. Totally 7(algorithms)*3(presentations)*2(with or without window) models are built. We randomly split the development set into training set and test set with ratio 4:1. Similarity scores of each t_1&t_2 pairs from the top 50% best-performance models are used as features.

Directional Entailment Features: The directional entailment features are designed to indicate the entailment direction between t_1 and t_2. Features in this category are symmetric, which can be further classified into four types: (1) the number proportions of equal linguistic units of t_1 to t_2/t_2 to t_1; (2) whether a specific type of NE only appears in t_1 or t_2; (3) entailment between t_1 and t_2 in different linguistic granularities, ranging from word to syntactic structures; (4) Whether there is any equivalent definitional (mostly is-a) relation between contents of t_1 and t_2. Detailed features as listed in Table 1.

Contradiction Feature: Contradiction features are extracted to recognize the contradiction relations between t_1 and t_2. A large part of the features for contradiction recognition are already included in the directional entailment features, especially in the "entailment in different linguistic granularities" and "definitional" features. Table 2 lists the remainder, which are lexical features including antonym, sentiment word.

2.2 Classification Module

The MC subtask is represented in three different ways, namely, as a five-class recognition problem, as five binary-class recognition problems and as a two-dimensional hierarchical recognition problem, respectively. For each problem, different ML methods are applied to build automatic recognition models. We assume that the classifiers built from different problem representations are complementary to each other, so are the different ML methods. Therefore, voting of multiple classifiers is leveraged to improve the recognition

Table 2: Description list of directional entailment features

Feature	Description
Antonym_Rel	Whether two words form t_1 and t_2 are antonyms
Negative	Whether t_1/t_2 has negative words, two negative words in one statement are counted as a positive word.
Positive	Whether t_1/t_2 has positive words

accuracy. The classification schemes are illustrated in Fig. 2. The three runs submitted are described as follows:

Run01: Build a five-class classifier by using one ML algorithm. Three machine learning algorithms are employed, namely Decision tree (DT), Support Vector Machine (SVM) and Logistic Regression (LR), respectively. Since DT achieves the highest accuracy on the development set,the classification results of DT are submitted.

Run02: Voting among three five-class classifiers built from different ML algorithms. The class taxonomy is the same as Run 01. Nevertheless, a voting among the classification results of DT, SVM and LR is conducted. If the result from at least two classifiers is the same, then it is selected; while if the three results are different from each other, the DT result is used as default.

Run03: Voting between three classifiers using different class taxonomies and ML algorithms. Besides the five-class taxonomy, the other two problem representations are employed: (1) binary-class: five binary-class classifiers (e.g., for 'F', build a classifier in the form of t_2: (F, F)) are built using DT, SVM and LR for each of the five classes, respectively. When more than one class is recognized, the class with the highest confidence is considered as the final class. One the contrary, if none of the classes are labeled as YES, the class with the

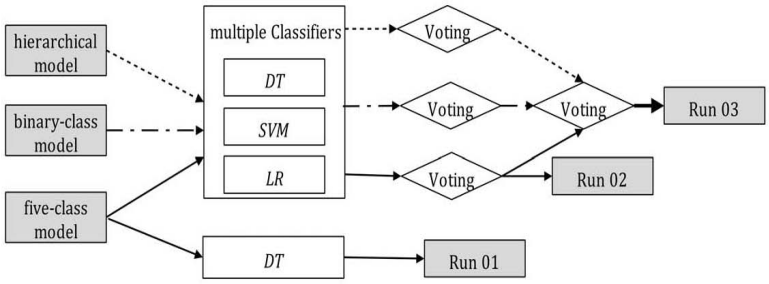

Figure 2: Classification model for each submitted run.

least confidence is considered as the final class. (2) two-dimensional hierarchical-class: firstly, we reverse the order of t_1-t_2 pairs labeled as 'R' in the development set, then as the first dimension in the hierarchy, a binary-class model is built to recognized whether t_1 entails t_2 or not. Another two models are built in the secondary dimension of the hierarchy, for 'F', 'B', 'R' and 'C',' I' recognition, respectively.

Finally, a voting among the three ML algorithm of the same class taxonomy are first conducted, based on which, another voting of the results of the five-class, binary-class and hierarchical-class is conducted to decide the final class. If the three results are different from each other, the result output by 5-class model is used as default.

3. EVALUATION AND DISCUSSION

The open source tool Weka is employed for classification in this system. Our system mainly focus on CS. The results for CT are derived from using the models built for CS directly. Run01 and run02 of BC in CT use the same configurations as run01 and run02 described in section 2. Only one run is submitted for MC in CT, which uses the same configuration as run02 described in section 2.

As for the RITE4QA subtask, the recognition model is built using the same feature set for subtasks BC and MC, except that the development set is expanded by the test sets of previous subtasks. Besides, the test corpus contains a large amount of traditional Chinese specific words. Since LTP has difficulty in word-segmentation and NE recognition for them, a maximum-common-string matching is conducted between t_1 and t_2. The matched strings are dynamically added into our lexicon as proper nouns. In this way, we hope to enhance the recall of NE recognition, and the RITE4QA performance ultimately. Furthermore, the voting mechanism between different models is N-class-biased, i.e., if one model outputs 'N', the final class is 'N', in hope of enhancing the precision of the answer. Run01 and run03 uses the same configuration as run02 described in section 2, with the N-class biased strategy. Run01 uses the original built lexicon; while run03 uses the dynamically updated lexicon. Run02 is the evaluation result using SVM as the classifier, with the dynamically updated lexicon.

3.1 Official Results

Table 3 and table 4 display the evaluation results of BC and MC subtasks on CS and CT side, respectively. As can be seen, accuracy of the three runs increases incrementally for CS, which shows the effectiveness of the voting strategies. Especially, run02 enhances the accuracy for 6.92% and 8.52% from run01, while run03 further enhances the accuracy for 2.51% and 2.72% from run02. This result indicates that the accuracy enhancement of employing the voting of different problem representations is not as high as the voting of different ML methods.

Table 3: Evaluation results of BC and MC subtask in CS

CS	BC	MC
run01	0.708	0.575
run02	0.757	0.624
run03	0.776	0.641

Table 4: Evaluation results of BC and MC subtask in CT

CT	BC	MC
run01	0.613	0.497
run02	0.597	

We directly applied the model developed for CS to CT. It is observed that the performance of CT drops sharply as compared with CS. A deep analysis show that many errors are caused by the word segmentation and pos-tagging, because the LTP tool has the difficulty to process CT, especially CT NE recognition. Furthermore, voting of multiple ML methods decreases the accuracy slightly for 2.61% from using the DT algorithm alone. One possible reason is that errors are accumulated and strengthened by the voting, instead of reduced. More detailed examination should be conducted.

Table 5 displays the evaluation results of RITE4QA in both CS and CT. As can be seen, run01 achieves the best performance among the three runs. Performance of run03 drops slightly from that of run01, indicating that the dynamic lexicon updating strategy does not help to recognize more NEs. On the contrary, this strategy introduces more noises and affects the performance.

Table 5: Evaluation results of RITE4QA subtask in CS and CT

CS&CT	Top1	MRR5
run01	0.2479	0.3520
run02	0.2234	0.2705
run03	0.2262	0.3398

3.2 Discussion

There is much room left to further improve the ICRC_HITSZ system. Several directions for further improvement are summarized as below:

Adding world knowledge from Wikipedia or Baidu baike: some entailments need to be inferenced from the world knowledge. For example, a mother should be female (#21 in CS test set), and 大威廉姆斯/ Venus Williams is the elder sister of 小威廉姆斯/Serena Williams (#33 in CS development set), and so on. Anyhow, how to represent the world knowledge, and how to design the inference mechanism for the entailment task remains a problem.

Number/time normalization improvement: At present, the number/time normalization module used in our system is only able to compare two numbers or times directly. The relation between a moment and a time interval cannot be recognized. Besides, the literal comparison of numbers (such as 大于/larger than，小于/lower than) in the entailment pairs cannot be recognized either.

Co-reference resolution: adding windowing in EDIT can alleviate the influence of length difference between t_1 and t_2. However, for long sentences, some co-references should be resolved, to make the similarity calculation more precise.

Conclusion phenomenon: one type of entailment not solved in our system is the "conclusion" or "cause-result" phenomenon. For example, in #13 of CS test set,

<pair id="13">
<t_1>
不吃早餐易导致高胆固醇而增加心脏病风险
(Not eating breakfast easily *leads to* high cholesterol and increased risk of heart disease)
</t_1>
<t_2>
心脏病与胆固醇有关
(Heart disease *is related to* cholesterol)
</t_2>
</pair>

the is-related-to relation in t_2 can be concluded from the cause-result relation in t_1. To resolve this type of phenomenon, cues indicating relations between event / entities should be detected first. Inference mechanisms to deduce the two relations in t_1 and t_2 should also be designed.

4. CONCLUSIONS

This paper presents the ICRC_HITSZ system in the NTCIR-9 RITE challenge. We participate in the BC, MC and RITE4QA tasks on CS and CT side, respectively. Different linguistic level features and voting of multiple classifiers using multiple problem representations are leveraged to improve the recognition accuracy. Evaluation results demonstrate that the voting strategies are effective, with the highest performance ranked at the fourth place in terms of accuracy, and at the second place in terms of groups in both tasks.

Our future work includes adding world knowledge and inference mechanisms into the entailment module. Besides, ways of generating large-scale corpus consisting of enriched entailment phenomena will also be examined.

5. ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (No. 61173075 and 60973076) and HIT.NSFIR.2010124.

6. REFERENCES

[1] I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research.*, 38(1):135–187, May 2010.

[2] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. The fifth pascal recognizing textual entailment challenge. In *Proceedings of TAC'2009*, 2009.

[3] W. Che, Z. Li, and T. Liu. Ltp: A chinese language technology platform. In *COLING (Demos)*, pages 13–16, 2010.

[4] M. Kouylekov and M. Negri. An open-source package for recognizing textual entailment. In *ACL (System Demonstrations)*, pages 42–47, 2010.

[5] P. Malakasiotis and I. Androutsopoulos. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 42–47, 2007.

[6] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, S. S. Y. Miyao, and K. Takeda. Overview of ntcir-9 rite: Recognizing inference in text. In *NTCIR-9 Proceedings, to appear*, 2011.

[7] R. Xu and C. Kit. Incorporating feature-based and similarity-based opinion mining - ctl in ntcir-8 moat. In *Proceedings of NTCIR-8 Workshop Meeting*, pages 13–16, 2010.