The Description of the NTOU RITE System in NTCIR-10

Chuan-Jie Lin and Yu-Cheng Tu

Department of Computer Science and Engineering
National Taiwan Ocean University
No 2, Pei-Ning Road, Keelung 202, Taiwan R.O.C.
+886-2-24622192 ext. 6610

{cjlin, yctu.cse}@ntou.edu.tw

ABSTRACT

The textual entailment system determines whether one sentence entails another in common sense. This is the second time of a RITE task in NTCIR projects. Three different subtasks, BC, MC, and RITE4QA, were held this time. We proposed several new features, and tried to construct RITE systems by using binary- or multi-class classifiers. After correcting errors in our submitted runs, our best (unofficial) system in the BC subtask achieves 65.12% in macro F-measure and 66.52% in accuracy. The performance of our MC classifiers is around 44.8% in macro F-measure and 56.64% in accuracy. Our best (unofficial) system in the RITE4QA subtask achieves 32.67% in Top1 accuracy, 41.74% in MRR, and 56% in Top 5 accuracy regarding to the WorseRanking.

KEYWORD

NTOU, RITE, Traditional Chinese, BC and MC subtasks, RITE4QA, WordNet, Sinica BOW, Wikipedia

1. INTRODUCTION

Recognizing Textual Entailment is a task to determine whether one sentence can entail another sentence in a common sense. The RTE techniques are useful in many research areas, such as answer validation in Question Answering [1] and text extraction in summarization [2].

Recognizing Textual Entailment has been studied for several years, such as in the TAC RTE tracks [3] and EVALITA IRTE task [4]. It is the second time to have RTE tasks focusing on Japanese and Chinese [5]. It is also our second attempt to develop a Chinese RTE system.

We participated in three subtasks: Binary-Class (BC), Multi-Class (MC), and RITE4QA subtasks. Given a pair of sentences (*t*1, *t*2), the BC subtask is to determine whether *t*1 entails *t*2, while MC subtask is to determine the entailment direction or contradiction. The labels used in BC subtask are “Y” and “N”. The labels defined in MC subtask are “F” (for forward entailment, *t*1 ⇒ *t*2), “B” (for bidirectional entailment, *t*1 ⇔ *t*2), “C” (for contradiction), and “I” (for independence).

The RITE4QA subtask is also a Y/N binary-class subtask except that the pairs are generated from QA data which can be regarded as an answer validation process.

Our RITE system is mainly a SVM classifier trained by using several features concerning surface and sense similarities. We submitted three formal runs in each subtask by using the same three approaches to see the applicability of the proposed strategies.

2. SYSTEM DESCRIPTION

2.1 Classifiers

We built five classifiers by SVM during this task. Three of them take one sentence pair (*t*1, *t*2) as input, while two of them take the prediction results from other classifiers as input.

The BC classifier determines whether *t*1 entails *t*2 and gives a “Y” or “N” label. The MC classifier determines the entailment relationship (F, B, C, or I, as defined in the MC subtask) between *t*1 and *t*2.

The third classifier, Contra, determines whether *t*1 and *t*2 are contradiction. This classifier helps the BC classifier to determine multi-class entailment relationships.

The fourth classifier, BCbyMC, determines whether *t*1 entails *t*2 (and gives a Y or N label) by the prediction of the MC classifier. When a pair (*t*1, *t*2) is predicted, the probabilities of 4 classes (F, B, C, and I) are also generated. These probabilities serve as features for the BCbyMC classifier. The output of the classifier is Y or N.

The fifth classifier, MCbyBC, determines the entailment relationship between *t*1 and *t*2 (and gives a label of F, B, C, or I) by the predictions of the BC classifier and the Contra classifier. For a pair (*t*1, *t*2), the BC classifier determines whether *t*1 entails *t*2 and *t*2 entails *t*1 (by using (*t*2, *t*1) as input), and the Contra classifier determines whether *t*1 contradicts *t*2 and *t*2 contradicts *t*1 (in both direction although contradiction should be symmetric). The probabilities of Y and N classes for these 4 predictions serve as features for the MCbyBC classifier. The output of the classifier is F, B, C, or I.

2.2 Training Sets

The formal development set of NTCIR-10 RITE CT-MC subtask is created by merging the development set and the formal test set of NTCIR-9 RITE CT-MC subtask. We will refer to the development set in RITE1 as CT-MC-dev1, and the set in RITE2 as CT-MC-dev2 later in this paper. They are used to train MC classifiers.

By changing F and B labels into Y, and C and I labels into N, each MC development set can be converted into a BC development set. They are called CT-BC-dev1 and CT-BC-dev2 in this paper and are used to train BC classifiers.

To create Contra training sets, we changed C labels in CT-MC-dev2 into Y labels, and all the other labels into N labels. However, the numbers of C and non-C labels were unbalanced. So we duplicated Y-type pairs for several times to make the two labels balanced. The training set is referred as CT-Contra-dev2 in this paper.

To create a training set for the BCbyMC classifier, pairs in CT-BC-dev2 were predicted by the MC classifier (trained by using CT-MC-dev2). The correct binary label and the probabilities of each pair belonging to the 4 labels were collected as the training data.

To create a training set for the MCbyBC classifier, each pair ($t1$, $t2$) in CT-MC-dev2, together with its flipped pair ($t2$, $t1$), was predicted by the BC classifier (trained by using CT-BC-dev2) and the Contra classifier (trained by using CT-Contra-dev2). Its correct multi-class label and the probabilities of being Y and N labels (totally 8 features) were collected as the training data.

2.3 Text Processing

The text processing on sentences in the training sets and RITE2 testset includes Chinese word segmentation, POS tagging, named entity recognition, temporal resolution, and encyclopedia lookup. All systems were built in our lab.

Based on the characteristics of Chinese POS, only normal nouns, proper nouns, and verbs were considered as content words in our experiment.

The information of person, location, and time is important when describing an event. Therefore, person names and location names were identified by our NER system.

Date expressions were extracted by patterns. Moreover, the (year, month, day) information in a date expression was resolved if possible.

In order to catch more contemporary terms, we also considered Wikipedia titles as a feature. The titles of Wikipedia entries appearing in the sentences were extracted by the longest matching strategy.

For each sentence t_i, $i \in (1, 2)$, the following sets were created for similarity scoring and feature extraction:

N_i	the set of distinct nouns in t_i
V_i	the set of distinct verbs in t_i
W_i	the set of distinct content words in t_i ($= N_i \cup V_i$)
P_i	the set of distinct person names in t_i
L_i	the set of distinct location names in t_i
D_i	the set of distinct date expressions in t_i
K_i	the set of distinct Wikipedia titles in t_i

2.4 Function Definition

Some functions needed for feature extraction are defined as follows.

$len(t)$ = the length of a sentence t (in bytes)

$|S|$ = the number of elements in a set S

$isOverlap(A, B) = 1$ if $A \cap B \neq \varnothing$; 0 otherwise.

$isDiff(A, B) = 1$ if $A \neq \varnothing$, $B \neq \varnothing$, and $A \neq B$; 0 otherwise.

$dateWeight(d)$: weight of a date expression d
$= 0.6 \times \delta(d, year) + 0.3 \times \delta(d, month) + 0.1 \times \delta(d, day)$,
where $\delta(d, x) = 1$ when the x field in a date expression d is not empty, and 0 if empty.

$dateWeight(D)$: total weight of a set of date expressions D
$= \sum_{d \in D} dateWeight(d)$

$dateSim(d_i, d_j)$: similarity of two date expressions d_i and d_j
$= 0.6 \times \mu(d_i, d_j, year) + 0.3 \times \mu(d_i, d_j, mon) + 0.1 \times \mu(d_i, d_j, day)$,
where $\mu(d_i, d_j, x) = 1$ when the x fields in date expressions d_i and d_j are identical but not empty, and 0 otherwise.

$dateSim(D_i, D_j)$: similarity of sets of date expressions D_i and D_j
$= \sum_{d \in D_i} \max_{y \in D_j} dateSim(d, y)$

$depth(s)$: depth of a sense s in WordNet

$nca(s_1, s_2)$: nearest common ancestor of s_1 and s_2 where the sum of distance from the ancestor to s_1 and s_2 is smallest

$WNsim(s_1, s_2)$: similarity of two senses s_1 and s_2 measured in WordNet by an equation proposed by Wu and Palmer [6]
$= 2 \times depth(nca(s_1, s_2)) / (depth(s_1) + depth(s_2))$
We used Chinese WordNet (Sinica BOW) and English WordNet 2.1 to measure the similarity scores.

$$WNsim(w_1, w_2) = \max_{\substack{s_1 \in Sense(w_1),\\ s_2 \in Sense(w_2)}} WNsim(s_1, s_2)$$

$WNsim(W_1, W_2)$: the sum of WordNet similarities of the aligned words in W_1 and W_2. The alignment algorithm is described in our RITE1 paper [7].

2.5 Features

20 features were used to train our first three SVM classifiers. The definitions of the features are given as follows.

len_1:	length difference ratio $(len(t1) - len(t2)) / len(t1)$
len_2:	length difference ratio $(len(t1) - len(t2)) / len(t2)$
wc_1:	word number difference ratio $(\|W_1\| - \|W_2\|) / \|W_1\|$
wc_2:	word number difference ratio $(\|W_1\| - \|W_2\|) / \|W_2\|$
ovr_1:	ratio of overlapped words $\|W_1 \cap W_2\| / \|W_1\|$
ovr_2:	ratio of overlapped words $\|W_1 \cap W_2\| / \|W_2\|$
wk_1:	ratio of overlapped Wiki titles $\|K_1 \cap K_2\| / \|K_1\|$
wk_2:	ratio of overlapped Wiki titles $\|K_1 \cap K_2\| / \|K_2\|$
pn_{same}:	having same person names $isOverlap(P_1, P_2)$
pn_{diff}:	having different person names $isDiff(P_1, P_2)$
lc_{same}:	having same location names $isOverlap(L_1, L_2)$
lc_{diff}:	having different location names $isDiff(L_1, L_2)$
wk_1:	weight ratio of dates $dateSim(D_1, D_2)$/dateWeight(D_1)
wk_2:	weight ratio of dates $dateSim(D_1, D_2)$/dateWeight(D_2)
wn_{W1}:	weight ratio of senses $WNsim(W_1, W_2) / \|W_1\|$
wn_{W2}:	weight ratio of senses $WNsim(W_1, W_2) / \|W_2\|$
wn_{N1}:	weight ratio of noun senses $WNsim(N_1, N_2) / \|N_1\|$
wn_{N2}:	weight ratio of noun senses $WNsim(N_1, N_2) / \|N_2\|$
wn_{V1}:	weight ratio of verb senses $WNsim(V_1, V_2) / \|V_1\|$
wn_{V2}:	weight ratio of verb senses $WNsim(V_1, V_2) / \|V_2\|$

Note that a feature value is defined as 0 if its denominator is 0.

3. EXPERIMENTS

3.1 Formal Run Description

We submitted 3 runs for the BC subtask, 3 runs for MC subtask, and 3 runs for RITE4QA subtask this year. The run settings were described as follows.

NTOUA-CT-BC-01: a BC classifier trained with CT-BC-dev2
NTOUA-CT-BC-02: a BCbyMC classifier
NTOUA-CT-BC-03: a BC classifier trained with CT-BC-dev1
NTOUA-CT-MC-01: a MC classifier trained with CT-MC-dev2
NTOUA-CT-MC-02: a MCbyBC classifier
NTOUA-CT-MC-03: a MC classifier trained with CT-MC-dev1
NTOUA-CT-RITE4QA-01: same as NTOUA-CT-BC-01
NTOUA-CT-RITE4QA-02: same as NTOUA-CT-BC-02
NTOUA-CT-RITE4QA-03: same as NTOUA-CT-BC-03

Table 1. Performance of CT-BC formal and unofficial runs

RunID	macroF	Acc	Y-F	Y-P	Y-R	N-F	N-P	N-R
NTOUA-CT-BC-01	**32.63**	33.48	**25.06**	32.34	20.46	40.20	34.08	49.00
NTOUA-CT-BC-02	30.70	**34.17**	15.20	25.37	10.86	**46.20**	36.83	61.94
NTOUA-CT-BC-03	31.71	33.94	19.39	28.81	14.61	44.04	35.89	56.97
NTOUA-CT-BC-01-u	**65.12**	**66.52**	72.09	65.92	79.54	**58.16**	67.66	51.00
NTOUA-CT-BC-02-u	62.18	65.83	73.94	63.17	89.14	50.41	74.63	38.06
NTOUA-CT-BC-03-u	63.44	66.06	73.23	64.11	85.39	53.64	71.19	43.03
NTOUA-CT-BC-04-u	61.41	66.06	**74.81**	62.71	92.69	48.00	79.77	34.33

Table 2. Performance of CT-MC formal and unofficial runs

RunID	macroF	Acc	B-F	B-P	B-R	F-F	F-P	F-R	C-F	C-P	C-R	I-F	I-P	I-R
NTOUA-CT-MC-01	44.63	**56.64**	**62.07**	54.82	71.52	**65.79**	54.01	84.15	0.00	0.00	0.00	50.66	69.28	39.93
NTOUA-CT-MC-02	33.49	49.94	1.29	25.00	0.66	62.50	50.96	80.79	**13.98**	18.06	11.40	**56.20**	56.49	55.90
NTOUA-CT-MC-03	**44.80**	55.73	61.10	50.43	77.48	64.21	55.00	77.13	1.50	5.26	0.88	52.40	70.59	41.67
NTOUA-CT-MC-04-u	34.43	47.67	0.00	0.00	0.00	63.66	56.34	73.17	**19.61**	15.63	26.32	54.45	57.03	52.08

Table 3. Performance of CT-RITE4QA formal and unofficial runs

	WorseRanking						BetterRanking					
CT	R			R+U			R			R+U		
Run	Top1	MRR	Top5	Top1	MRR	Top5	Top1	MRR	Top5	Top1	MRR	Top5
NTOUA-CT-RITE4QA-01	**8.00**	9.28	11.33	13.33	17.06	22.67	8.67	9.61	11.33	15.33	18.17	22.67
NTOUA-CT-RITE4QA-02	7.33	8.78	10.67	11.33	14.11	18.00	8.00	9.22	10.67	12.00	14.56	18.00
NTOUA-CT-RITE4QA-03	**8.00**	**9.97**	**13.33**	12.67	17.19	24.00	9.33	10.63	13.33	14.67	18.30	24.00
NTOUA-CT-RITE4QA-01-u	**32.67**	**41.74**	**56.00**	34.67	44.68	60.67	**37.33**	**44.47**	**56.00**	39.33	47.40	60.67
NTOUA-CT-RITE4QA-02-u	29.33	37.40	50.67	33.33	43.62	59.33	32.67	39.66	50.67	37.33	46.21	59.33
NTOUA-CT-RITE4QA-03-u	29.33	39.01	54.00	34.00	44.34	60.00	34.00	41.92	54.67	38.67	47.14	60.67
orgQAsys-CT-RITE4QA-01	7.33	11.54	22.67	10.67	16.99	31.33	40.67	47.60	57.33	44.67	52.32	64.00

The formal evaluation metric of BC and MC subtasks are Macro F-measure (the average of F-measures of every labels) the accuracy score (Acc, the ratio of correctly predicted pairs). The CT-BC test set contains 900 pairs with half as Y-pairs and half as N-pairs. The CT-MC test set also contains 900 pairs, with equal number of pairs in each of the 5 classes.

The formal evaluation metric of RITE4QA are Top1 accuracy (ratio of questions being correctly answered by top-1 answers), MRR (the average of reciprocals of the highest ranks of correct answers), and Top5 accuracy (ratio of questions being correctly answered by top-5 answers.

3.2 Results and Discussion

Table 1, Table 2, and Table 3 show the evaluation results of all the runs in CT-BC, CT-MC, and CT-RITE4QA subtasks, respectively. In Table 1 and Table 2, the rest columns besides macroF and accuracy show F-measure scores of all the labels.

Comparing NTOUA-CT-BC-02 to NTOUA-CT-MC-01, where both runs use the predictions of the MC classifier, it is strange that the performance drops after changing multi-class prediction into binary prediction by MC2BC SVM classifier. Moreover, according to the experience learnt from RITE1, MC classification approach achieves better performance than BC classification approach in the BC subtask. However in this year, NTOUA-CT-BC-02 does not outperform other runs.

It turns out that the output labels in our official runs were all incorrect. We mistakenly mapped all "Y" classes (in numbers) in the SVM predictions into "N" labels and vice versa. By reproducing the 9 official runs and redoing evaluation, their true results are shown in the three tables with suffix "-u" added to the end of the names of the corresponding unofficial runs.

Besides, we also made an unofficial BC run which was created by directly converting MC predictions into binary labels by rules, i.e. mapping F and B into Y, and C and I into N. The run is named as NTOUA-CT-BC-04-u and its evaluation result is listed in Table 1.

In order to do comparison, another unofficial MC run was also created by directly converting BC predictions into multi-class labels by rules represented in Table 4. This run is named as NTOUA-CT-MC-04-u and its evaluation result is listed in Table 2.

Table 4. MCbyBC prediction rules

(*t*1, *t*2) by BC	-	-	-	Y	Y	N
(*t*2, *t*1) by BC	-	-	-	Y	N	-
(*t*1, *t*2) by Contra	Y	Y	N	N	N	N
(*t*2, *t*1) by Contra	Y	N	Y	N	N	N
Multi-Class Prediction	C	C	C	B	F	I

In the four CT-BC unofficial runs, the first system (a BC classifier trained with CT-BC-dev2) achieves the best performance in both macro F-measure and accuracy. The two cross-model systems (CT-BC-02-u, a BCbyMC classifier, and CT-BC-04-u, a rule-based system using a MC classifier) are worse than the two single-model systems.

In the four CT-MC runs, NTOUA-CT-MC-03 (a MC classifier trained with CT-MC-dev1) achieves the best performance in macro F-measure but NTOUA-CT-MC-01 (a MC classifier trained with CT-MC-dev2) is best in accuracy. Again, the two cross-model systems (CT-MC-02, a MCbyBC classifier, and CT-MC-

04-u, a rule-based system using four BC classifiers) are worse than the two single-model systems.

Interestingly, CT-MC-02 achieves the highest F-measure in the Contradiction relation among all the participating systems. CT-MC-04-u gets an even higher score. The reason is because that these systems deliver more C labels than others. CT-MC-02 outputs 72 contradictory pairs and CT-MC-04-u 192 pairs. But they are very weak in predicting bidirectional entailment pairs.

The three RITE4QA runs become very competitive after the errors are corrected. Regarding to the WorseRanking, NTOUA-CT-RITE4QA-01-u achieves 32.67% in Top1 accuracy and 41.74% in MRR, which is better than the best CT-RITE4QA official run in the overview paper.

4. CONCLUSION AND FUTURE WORK

It is our second time to participate in NTCIR RITE task. Several features have been proposed to learn entailment relationship classifiers. 9 formal runs were submitted. Unfortunately the 3 BC runs and 3 RITE4QA runs contained fatal errors. This paper obverses the performance of corrected unofficial runs instead.

Our best (unofficial) system in the BC subtask achieves 65.12% in macro F-measure and 66.52% in accuracy. The performance of our MC classifiers is around 44.8% in macro F-measure and 56.64% in accuracy. Our best (unofficial) system in the RITE4QA subtask achieves 32.67% in Top1 accuracy, 41.74% in MRR, and 56% in Top 5 accuracy regarding to the WorseRanking.

As our first system in each subtask always achieves the best performance, it can be concluded that larger training set is better, and single-model systems outperforms cross-model systems under current feature settings and training sets.

We adopted a different set of features from the set in our RITE1 system to do machine learning. In the future, we will study the efficiency of each feature and find out the best combination. The advantages and weaknesses of our systems will also be observed under the characteristics of the NTCIR RITE training datasets.

5. REFERENCE

[1] Rodrigo, A., Penas, A., and Verdejo, F. 2008. Overview of the Answer Validation Exercise 2008. In *Working Notes for the CLEF 2008 Workshop*, LNCS 5706, 296-313.

[2] Lloret, E., Ferrández, Ó., Muñoz, R., and Palomar, M. 2008. A text summarization approach under the influence of textual entailment. In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science* (*NLPCS 2008*), 22–31.

[3] Bentivogli, L., Clark, P., Dagan, I., Dang, H.T., and Giampiccolo, D. 2010. The Sixth PASCAL Recognizing Textual Entailment Challenge. In *TAC 2010 Workshop Notebook Papers and Results*.

[4] Bos, J., Zanzotto, F.M., and Pennacchiotti, M. 2009. Textual Entailment at EVALITA 2009, In *Proceedings of EVALITA 2009*.

[5] 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. 2013. Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Workshop. In *NTCIR-10 Proceedings*, to be appeared.

[6] Wu, Z. and Palmer, M. 1994. Verb semantics and lexical selection. In *Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics*, 133-138.

[7] Lin C.J. and Hsiao B.Y. 2011. The Description of the NTOU RITE System in NTCIR-9. In *NTCIR-9 Proceedings*, 353-356.