The Description of the NTOU RITE System in NTCIR-9

Chuan-Jie Lin and Bo-Yu Hsiao
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, B97570001}@ntou.edu.tw

ABSTRACT
The textual entailment system determines whether one sentence can entail another in a common sense. We proposed several approaches to train textual entailment classifiers, including setting ancestor distance threshold, expanding training corpus, using different sets of features, and tuning classifier settings. The results show that a MC classifier trained by using an expanded training corpus and scoring features performs the best with an accuracy of 64.22% in BC task and 46.11% in MC task.

KEYWORD
NTOU, RITE, Traditional Chinese, BC and MC subtasks, WordNet, Google Translate

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 first time to have RTE tasks focusing on Japanese and Chinese [5]. It is also our first 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 ($t1$, $t2$), the BC subtask is to determine whether $t1$ entails $t2$, 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, $t1 \Rightarrow t2$), "R" (for reverse entailment, $t2 \Rightarrow t1$), "B" (for bidirectional entailment, $t1 \Leftrightarrow t2$), "C" (for contradiction), and "I" (for independence).

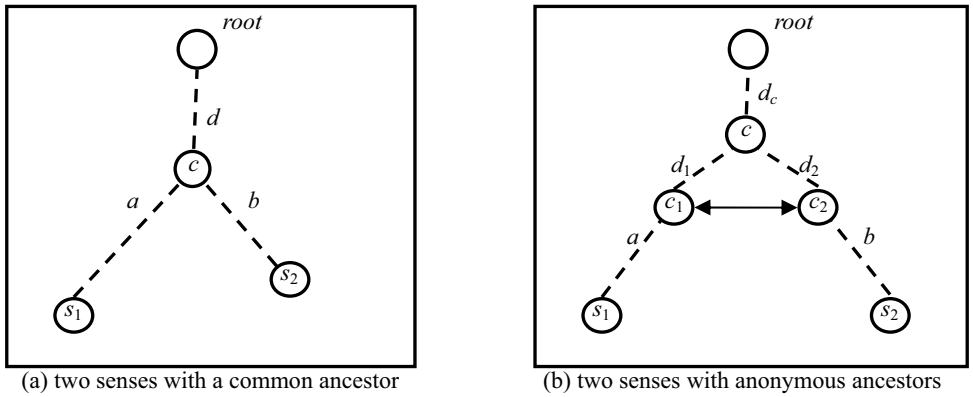
The RITE4QA subtask is also a Y/N binary-class subtask except that the pairs are generated from QA data which can be regarded as an answer validation process.

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
Given two Chinese sentences (*t1*, *t2*), our system first translates them into English, and then finds a one-to-one word alignment between them according to the word sense similarity in WordNet. Feature values are determined by the word alignment and a multi-class classifier trained by machine learning is utilized to predict the entailment type of the sentence pair. This section describes the details in our system.

2.1 Text Translation
In this RITE task, we participated in the Chinese language subtasks. The develop set and the test set are all written in Chinese, but we want to measure the word similarity by using WordNet, which is a thesaurus of English terms. Lack of techniques mapping an English thesaurus into a Chinese one, we simply use Google Translate service to translate all the sentences into English and handle them as usual. The unknown words to the Google Translate service are remained as Chinese strings and are still regarded as content words when doing word alignment.

2.2 WordNet Relatedness Score
The *WordNet relatedness score* we used to measure the relatedness of two words in WordNet is an extension of *WordNet distance*.



(a) two senses with a common ancestor (b) two senses with anonymous ancestors

Figure 1. Illustrations of WordNet relatedness score

Given two words w_1 and w_2, all their senses are paired and looked up in WordNet. The WordNet relatedness score of a pair of senses is defined as follows.

As shown in Figure 1(a), given two senses s_1 and s_2, let c be their nearest common ancestor, a and b the lengths of paths from the two senses to their common ancestor c, and d the length of the path from the root to c. If more than one possible path is found, the shortest one is selected. The ***WordNet relatedness score*** $WNrel(s_1, s_2)$ of a pair of senses is defined as:

$$WNrel(s_1, s_2) = (c + depth_{MAX} - (a + b)/2) / 2depth_{MAX}$$

where $depth_{MAX}$ is the length of the longest path started from the root to a leaf node. As we can see in the definition of the equation, the WordNet relatedness score of two words is higher when their WordNet distance (*i.e.* $a + b$) is shorter and their common ancestor has more specific sense. The addition of $depth_{MAX}$ is to make sure that the score will not become negative. The score is normalized into 0 ~ 1 by divided by $2 \times depth_{MAX}$.

When two senses do not have a common ancestor in WordNet, their relatedness score is defined as 0.

The WordNet relatedness score in antonymy relationship is defined in the similar way except that, as depicted in Figure 1(b), a pair of ancestors (c_1, c_2) who are antonyms to each other is identified instead of a common ancestor. Now a and b are defined as the lengths of paths from s_1 and s_2 to their respective ancestors, and d is the average of the depths of c_1 and c_2 via a common ancestor c, i.e. $d = d_c + (d_1 + d_2) / 2$, where d_1 and d_2 are the lengths of paths from c_1 and c_2 to their common ancestor c, and d_c is the length of the path from c to the root.

The WordNet relatedness score of two words is defined as the highest relatedness score measured among their sense pairs, i.e.

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

2.3 Word Alignment

Given a pair of sentences ($t1$, $t2$), the system first collects the content words in them, denoted as $\boldsymbol{P} = \{p_1, p_2, \ldots p_m\}$ and $\boldsymbol{H} = \{h_1, h_2, \ldots h_n\}$. Redundant words and stop words are removed in advanced. The second step is to do one-to-one word alignment between $\boldsymbol{P}$ and $\boldsymbol{H}$. Word alignment is found by two strategies, exact match and WordNet similarity. The words in the intersection of $\boldsymbol{P}$ and $\boldsymbol{H}$ are first aligned and removed from $\boldsymbol{P}$ and $\boldsymbol{H}$. All unaligned words left in $\boldsymbol{P}$ and $\boldsymbol{H}$ are paired and searched in WordNet. These pairs of words are sorted according to their WordNet relatedness scores. The similar or antonymous pair with the highest score is chosen as alignment and removed from $\boldsymbol{P}$ and $\boldsymbol{H}$ repeatedly until no more pairs with a common ancestor can be found. The word alignment algorithm is shown in Figure 2.

Given $\boldsymbol{P} = \{p_1, p_2, \ldots p_m\}$ and $\boldsymbol{H} = \{h_1, h_2, \ldots h_n\}$:

Let $\boldsymbol{A}$ be the chosen alignment. Set $\boldsymbol{A}$ as ∅ initially.
For each pair (p_i, h_j) where $p_i \in \boldsymbol{P}$, $h_j \in \boldsymbol{H}$, and $p_i = h_j$,
  $\boldsymbol{P} \leftarrow \boldsymbol{P} - \{p_i\}$; $\boldsymbol{H} \leftarrow \boldsymbol{H} - \{h_j\}$
  $\boldsymbol{A} \leftarrow \boldsymbol{A} + \{(p_i, h_j, \text{identical})\}$
For each pair remains in $\boldsymbol{P} \times \boldsymbol{H}$
  Measure its WordNet relatedness score.
Repeat
  Select a pair (p_i, h_j) with the highest relatedness score
  Let *type* be their relationship (similar or antonymous)
  $\boldsymbol{P} \leftarrow \boldsymbol{P} - \{p_i\}$; $\boldsymbol{H} \leftarrow \boldsymbol{H} - \{h_j\}$
  $\boldsymbol{A} \leftarrow \boldsymbol{A} + \{(p_i, h_j, type)\}$
Until $\boldsymbol{P}$ or $\boldsymbol{H}$ is empty.

Figure 2. Algorithm of word alignment for RITE pairs

2.4 Features

20 features are used to train our SVM classifier. Feature values can be determined by the result of word alignment. The definitions of the features are given as follows.

$f_{\|P\|}$:	number of distinct words in $t1$ (= m)
$f_{\|H\|}$:	number of distinct words in $t2$ (= n)
$f_{\|P\|\cap\|H\|}$:	number of overlapped words in $t1$ and $t2$
$ovr_{\|P\|}$:	ratio of overlapped words in $t1$ (= $f_{\|P\|\cap\|H\|} / f_{\|P\|}$)
$ovr_{\|H\|}$:	ratio of overlapped words in $t2$ (= $f_{\|P\|\cap\|H\|} / f_{\|H\|}$)
f_{capP}:	number of capitalized words in $t1$
f_{capH}:	number of capitalized words in $t2$
f_{capPH}:	number of overlapped capitalized words
ovr_{capP}:	ratio of overlapped capitalized words in $t1$ (= f_{capPH} / f_{capP})
ovr_{capH}:	ratio of overlapped capitalized words in $t2$ (= f_{capPH} / f_{capH})
f_{Nsim}:	number of aligned similar nouns
f_{Nant}:	number of aligned antonymous nouns
f_N:	total number of aligned nouns
f_{Vsim}:	number of aligned similar verbs
f_{Vant}:	number of aligned antonymous verbs
f_V:	total number of aligned verbs
d_{Nsim}:	average of the scores of aligned similar nouns
d_{Nant}:	average of the scores of aligned antonymous nouns
d_N:	average of the scores of aligned nouns
d_{Vsim}:	average of the scores of aligned similar verbs
d_{Vant}:	average of the scores of aligned antonymous verbs
d_V:	average of the scores of aligned verbs

Some features are explained in more details here. The values of f_{N*} and f_{V*} are obtained by counting the aligned pairs according to their *type* attributes in each POS. Since each aligned pair has a WordNet relatedness score, the values of d_{N*} and d_{V*} are obtained by averaging the WordNet relatedness score of the aligned pairs in each POS.

2.5 Classification

Our classifier is trained to label 5 classes defined in the Multi-Class (MC) subtasks which classes are forward-entailment (F), reverse-entailment (R), bidirectional-entailment (B), contradiction (C), and irrelevant (I). The classifier is used directly in a MC subtask.

When producing a run in Binary-Class (BC) subtasks, the multi-class labels are mapping into binary-class labels in the following way: labels F and B are mapped into label Y (entailment) while labels R, C, and I are mapped into label N (non-entailment).

3. LEARNING APPROACHES

We have experimented on several learning approaches including threshold setting of distance to the common ancestor, training corpus expansion, and feature selection. There approaches are explained in this section.

3.1 Ancestor Distance Threshold

When finding a common ancestor of a sense pair in WordNet, if the common ancestor is too far away from any of the sense, we

tend to regard them as a dissimilar pair and discard it by assigning its score as 0. Therefore, an appropriate threshold of the longest distance from the common ancestor to one sense (*i.e.* max(*a*, *b*) as shown in Figure 1) should be learned from the development set.

3.2 Training Corpus Expansion

Because our classifier is a multi-class one, we need a larger corpus in order to train a more reliable classifier. We expand the development set by swapping the pairs in the set. I.e. for each pair (*t*1, *t*2) with a label provided in the development set, a new pair (*t*2, *t*1) with a new label is expanded into the development set. The F-pairs (pairs labeled as 'F') become new R-pairs, so as R-pairs into F-pairs, and the B-, C-, and I-pairs remain their labels. By doing so, we can obtain a double-sized training corpus.

3.3 Feature Selection

The features can be divided into three groups, numbers of terms, ratios, and WordNet relatedness score averages. The number of terms is not as reliable as the ratio because it tends to be larger when a sentence is longer. We have experimented on feature combinations with or without number features f_*.

3.4 Binary Classes vs. Multi-Classes

Although our system is designed as a multi-class classifier in the first place, we have also used the same learning approaches to train a binary-class classifier and investigate its performance in a BC subtask.

In order to apply to a multi-class task, given a sentence pair (*t*1, *t*2), a BC classifier will predict in both direction, i.e. for (*t*1, *t*2) and (*t*2, *t*1). The label of "F", "R", and "B" can be easily derived from the predictions in the two directions. However, there is no way to predict a contradictory pair. We borrow the results from a MC classifier just for investigation convenience: if a BC classifier predicts "N" in both direction and a MC classifier (trained in the same setting as the BC classifier) predicts "C", the final label is "C"; otherwise it is labeled as "I".

4. EXPERIMENTS AND FORMAL RUNS

Four experiments were conducted to decide the best setting of our system, including the ancestor distance threshold setting, training corpus size, feature selection, and number of prediction classes. These settings were tuned by 10-fold cross-validation using the CT-MC development set, which contains 421 pairs. The pairs in each class were randomly yet equally separated into 10 folds, so that each of the 10 development subsets contains the same number of labels as the others.

The formal evaluation metric is the accuracy score (Acc) defined as the portion of correctly predicted pairs in a set. 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 settings and evaluation results of our formal runs submitted to the RITE CT-BC and CT-MC subtasks are also given here, so are the results of the runs submitted to the RITE4QA CT subtask.

4.1 Ancestor Distance Threshold Settings

We set the threshold as 1 to 18, the maximum depths in the WordNet noun dictionary. Table 1 shows some results in the CT-MC development set by different learning approaches.

Table 1. Experimental results of ancestor distance settings

Threshold:	6	7	8	14	20
all*f*_c1_mc	48.69	**48.93**	48.22	48.69	48.69
score*f*_c1_mc	47.27	**47.74**	47.27	**47.74**	**47.74**
all*f*_c2_mc	52.02	**52.49**	51.54	50.83	50.83
score*f*_c2_mc	51.07	52.02	**52.49**	52.02	52.02

As we can see in Table 1, by setting the threshold as 7, it can achieve the best performance in most cases. The ancestor distance threshold is set to be 7 in the following experiments.

4.2 Learning Approach Evaluation

8 different approaches were proposed to predict the entailment relationships. We illustrate these approaches by the notion of [*featureSet*]_[*corpus*]_[*class*] defined as follows.

Setting	Value	Meaning
featureSet	all*f*	Using all features
	score*f*	Using ratio and score average features
corpus	c1	Original development set
	c2	Expanded development set
class	mc	Multi-class classifier
	bc	Binary-class classifier

In the training phase, as depicted in Table 1, using expanded training corpus (all*f*_c2_mc and score*f*_c2_mc) performs better than using the original corpus. However, the difference between feature selections is not obvious and stable. There is no conclusion which selection is better than the other.

Table 2 and Table 3 show the performances of all the 8 approaches evaluated using the CT-BC and CT-MC test sets, respectively. The internal columns in these two tables list the F-scores (defined as harmonic mean of precision and recall) in each class, where the last columns are the accuracy scores.

Table 2. Evaluation using CT-BC test set

System	RunID	YF	NF	Acc
all*f*_c1_mc	NTOUA-CT-BC-03	61.91	51.35	60.22
all*f*_c2_mc	NTOUA-CT-BC-01	64.34	54.14	61.33
score*f*_c1_mc		**70.11**	66.01	63.33
score*f*_c2_mc	NTOUA-CT-BC-02	68.55	57.90	**64.22**
all*f*_c1_bc		59.28	57.34	58.33
all*f*_c2_bc		60.89	60.89	60.89
score*f*_c1_bc		57.78	66.00	62.33
score*f*_c2_bc		34.87	**68.64**	57.67

Table 3. Evaluation using CT-MC test set

System	RunID	F	R	B	C	I	Acc
all*f*_c1_mc	NTOUA-CT-MC-03	55.70	55.07	35.73	**23.39**	39.07	42.11
all*f*_c2_mc	NTOUA-CT-MC-01	**61.17**	57.00	36.96	21.26	37.80	44.00
score*f*_c1_mc		53.33	54.27	**50.09**	0.00	38.69	44.78
score*f*_c2_mc	NTOUA-CT-MC-02	58.69	55.64	48.46	0.00	**44.22**	**46.11**
all*f*_c1_bc		54.02	51.15	35.73	21.60	35.59	40.33
all*f*_c2_bc		60.33	**58.19**	33.54	19.64	41.43	44.33
score*f*_c1_bc		55.59	52.55	32.61	0.00	43.73	41.78
score*f*_c2_bc		36.65	40.85	12.24	0.00	37.02	31.11

The results are different from what we have seen in the training phase. Expanding training corpus still succeeded in most cases, but using ratio and score average features outperformed the setting of using all features.

Surprisingly, MC classifiers outperformed BC classifiers in most cases, even in a BC task. MC classifiers can do better guessing for Y-pairs and B-pairs thus improve the performance.

4.3 RITE4QA Evaluation

Table 4 shows the results of our systems in the RITE4QA subtask. The outcome is totally different! The system trained by using all features and the original training set performs much better than the other two. We are still finding the reasons.

Table 4. RITE4QA evaluation results

System	RunID	Acc	MRR
all*f*_c1_mc	NTOUA-CT-RITE4QA-03	0.6346	0.3824
all*f*_c2_mc	NTOUA-CT-RITE4QA-01	0.5459	0.3803
score*f*_c2_mc	NTOUA-CT-RITE4QA-02	0.5124	0.3572

5. CONCLUSION AND FUTURE WORK

It is our first attempt to develop a textual entailment system. Several approaches have been proposed to train entailment relationship classifiers, including ancestor distance threshold setting, training corpus expansion, feature selection, and classifier setting. The results show that a MC classifier trained by using an expanded training corpus and scoring features performs the best with an accuracy of 64.22% in BC task and 46.11% in MC task.

Because we use Google Translate to rewrite all the sentences from Chinese into English, the errors of translation might have hurt the performance of our system. In the future, we plan to use a different strategy to abolish the language gap of well-known thesaurus.

6. 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*.

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