IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-10 RITE-2

Min-Yuh Day [1,*], Chun Tu[1], Shih-Jhen Huang[1], Hou-Cheng Vong[1], Sih-Wei Wu[1]

[1] Department of Information Management, Tamkang University, New Taipei City, Taiwan

myday@mail.tku.edu.tw, {david761113, kevincncod2, warriorgiroro}@gmail.com, wsw_ya@hotmail.com

ABSTRACT

In this paper, we describe the IMTKU (Information Management at TamKang University) textual entailment system for recognizing inference in text at NTCIR-10 RITE-2 (Recognizing Inference in Text). We proposed a textual entailment system using a hybrid approach that integrate semantic features and machine learning techniques for recognizing inference in text at NTCIR-10 RITE-2 task. We submitted 3 official runs for BC, MC and RITE4QA subtask. In NTCIR-10 RITE-2 task, IMTKU team achieved 0.509 in the CT-MC subtask, 0.663 in the CT-BC subtask; 0.402 in the CS-MC subtask, 0.627 in the CS-BC subtask; In MRR index, 0.257 in the CT-RITE4QA subtask, 0.338 in the CS-RITE4QA subtask.

Categories and Subject Descriptors

H.3.3 **[Information Search and Retrieval]**: Retrieval models and Search process

General Terms

Algorithms, Documentation, Experimentation

Team Name

IMTKU

Subtasks

RITE(CT_BC, CT_MC, CT_RITE4QA, CS_BC, CS_MC, CS_RITE4QA)

Keywords

IMTKU, Textual Entailment, Recognizing Textual Entailment in Chinese, Recognizing Inference in TExt (RITE), NTCIR, Hybrid Approach, Machine Learning, Dependency Parser, WordNet

1. INTRODUCTION

IMTKU participated in NTCIR-10 RITE-2 Binary-class (BC) subtask and Multi-class (MC) subtask in Traditional Chinese (CT). We submitted 3 official runs for BC and MC subtask. In addition, we also participate in RITE4QA subtask in both Traditional Chinese (CT) and Simplified Chinese (CS). We also submitted 3 official runs for RITE4QA subtask in both CT and CS language. In this paper, we described the algorithms, tools and resources used in IMTKU RITE system.

Recognizing Textual Entailment (RTE) is a PASCAL/TAC task of deciding given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text which is mainly focused on English [4, 5] RITE (Recognizing Inference in TExt), however, is a generic benchmark task organized by NTCIR-10 that addresses major text understanding needs in various NLP/Information Access research areas which is mainly focused on Japanese and Chinese [8, 9].

RITE is a benchmark task for evaluating systems which automatically detect entailment, paraphrase, and contradiction in texts written in Japanese, Simplified Chinese, or Traditional Chinese. There are three task settings, namely, Binary-class (BC) subtask, Multi-class (MC) subtask, and RITE4QA subtask in RITE. In all subtasks, a system input is two texts and an output is one of two or five labels [8].

For instance, in the BC subtask, an input text appears as follows:

T1: 香港的主權和領土是在 1997 由英國歸還給中國的。

(Hong Kong's sovereignty and territories were returned to China by the United Kingdom in 1997)

T2: 1997 年香港回歸中國。

(Hong Kong was returned to China in 1997)

The system output for the BC subtask is "YES" for the above T1, T2 pair.

For the Multi-class Classification (MC) in NTCIR-10 RITE-2, given a text pair (t1, t2), a system detects entailment in more detail. The class would be yes (forward entailment, backward entailment, paraphrase), no (contradiction, independence). However, backward-entailment can be detected by checking whether the flipped pair holds forward-entailment (i.e. t can be inferred from h) or not [9]. So backward-entailment relation was excluded from the set of semantic relation used in the MC subtask. It's also an intrinsic evaluation with more challenging setting than the BC subtask. The length of t1 and t2 is about the same [8].

Here is another instance of the MC subtask:

T1: 尼泊爾毛派叛亂份子攻擊安全警衛哨站。

(Nepal's Maoist insurgents assaulted a security guard outpost)

T2: 尼泊爾毛派游擊隊攻擊民航機。

(Nepal's Maoist guerrillas assaulted civil aviation aircraft)

The system output for the BC subtask is "NO" for the above

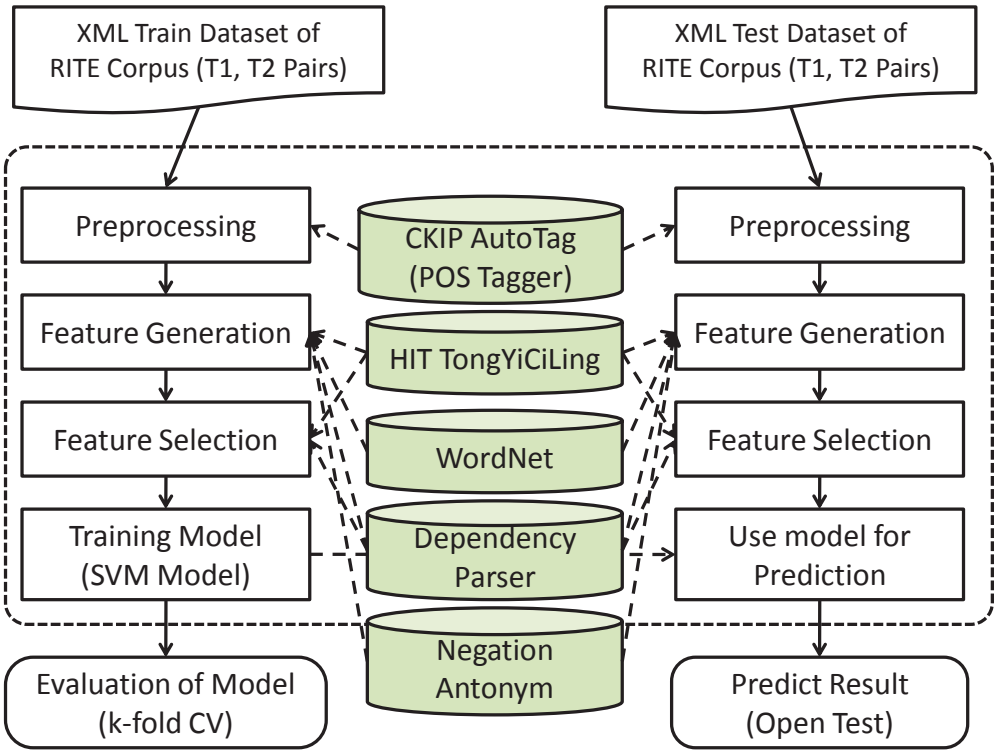

Figure 1. System Architecture of IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-10 RITE-2

T1, T2 pair. Further, the system output for the MC subtask is "Contradiction" where either T1 cannot be inferred from T2 or T2 cannot be inferred from T1.

Generally, features used for dealing with TE can be roughly divided into two categories, syntactic features and semantic features Semantic features include synonyms, antonyms, and negation. Most studies emphasize semantic features in text fragments. For example:

T1: 車諾比病毒在 1999 年 4 月總共造成超過 200 萬台電腦無法開機

(CIH caused severe boot problems in more than 200 million computers in April, 1999)

T2: 1999 年 4 月車諾比病毒總共造成逾 200 萬台電腦無法開機

(CIH caused severe boot problems in over 200 million computers in April, 1999)

If we consider only syntactic features, the output would be "Forward". However, if we consider both syntactic features and semantic features "超過(more than)" and "逾(over)" are synonyms. Therefore, the output would be "Bidirection" which is the correct answer.

For RITE4QA in NTCIR-10 RITE-2 is same as the BC subtask in terms of input and output, but as an embedded answer validation component in Question Answering system. In RITE-2 RITE4QA subtask, data creation and evaluation are different from NTCIR-9 RITE. In NTCIR-9 RITE, RITE4QA pairs were created by selecting answer passages similar to the question sentence, but such an approach may not be able to show the real performance of a RITE system in a factoid QA setting because answer-bearing sentences are not always similar to the question sentence.[9] NTCIR-10 RITE4QA pairs include all the possible answer-bearing sentences.

According to the task description of NTCIR-10 RITE-2 task [8], BC Subtask is defined as "Given a text pair *(t1, t2)* identify where *t1* entails (infers) a hypothesis *t2* or not", the expected system output label of RITE BC subtask is "{Y, N}". In addition, MC Subtask is defined as "A 4-way labeling subtask to detect (forward / bi-directional) entailment or no entailment (contradiction / independence) in a text pair", the expected system output label of RITE MC subtask is "{F,B,C,I}", where F means "forward entailment (t1 entails t2 AND t2 does not entails t1)"; B means "bidirectional entailment (t1 entails t2 AND t2 entails t1)"; C means "contradiction (t1 and t2 contradicts, or cannot be true at the same time)"; I means "independence (otherwise)". The evaluation of RITE system is the accuracy of labels predicted [8].

Section 2 describes the system architecture. In Section 3 describes the experimental results and analysis. Finally, we present the system performance in Section 4 and conclude our work in Section 5.

2. SYSTEM ARCHITECTURE

Figure 1 shows the proposed system architecture of IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-9 RITE.

2.1 Preprocessing

We extracted text fragments from NTCIR-10 RITE-2 raw datasets and use CKIP Autotag[3] for producing available datasets.

2.1.1 XML dataset extraction

We extracted IDs and text pairs from raw XML datasets of the RITE corpus for analysis.

2.1.2 Data format unification

A word may be expressed in different ways. For example, 1990 may be written "1990 年" or "一九九零年". It is thus necessary to unify the data format.

2.1.3 CKIP Autotag

We adopt the Chinese Knowledge and Information Processing (CKIP) System to process text pairs for analysis.

2.2 Feature Generation

We designated 14 semantic and syntactic features:

Word Similarity, String Length, String Length Difference, String Length Ratio, Longest Common Substring (LCS), Char-Based Edit Distance, Word Length, Word Length Difference, Word Length Ratio, Word-Based Edit Distance.

(1) String Length/Length Difference/Ratio

Basic syntactic approach we adopted as a feature. We use string length difference as a feature to reduce bias on a length basis. We can use string length ratio to confine the range between 0 and 1 to reduce bias and enhance accuracy.

(2) Longest Common Substring

We use Longest Common Substring [10] to find similarity in text pairs. The formula is:

$$\mathrm{LCS}(X_{1\ldots i}, Y_{1\ldots j}) = \begin{cases} \emptyset & \text{If i=0 or j=0} \\ \mathrm{LCS}(X_{1\ldots i-1}, Y_{1\ldots j-1}) + x_i & \text{If } x_i = y_i \\ \max\left(\mathrm{LCS}(X_{1\ldots i}, Y_{1\ldots j-1}), \mathrm{LCS}(X_{1\ldots i-1}, Y_{1\ldots j})\right) & \text{else} \end{cases}$$

Find the longest string (or strings) that is a substring (or are substrings) of two or more strings. We calculate the number of same characters appear in text pair without To the formula finds the longest string (or strings) that is a substring (or are substrings) of two or more strings. We first find the longest subsequences common to X_i and Y_j and then compare the elements x_i and y_j. If they are equal, then the sequence LCS (X_{i-1}, Y_{j-1}) is extended by that element, x_i. If they are not equal, then the longer of the two sequences, LCS (X_i, Y_{j-1}), and LCS (X_{i-1}, Y_j), is retained (if they are both the same length, but not identical, then both are retained.) Notice that the subscripts are reduced by 1 in these formulas, which can result in a subscript of 0. Since the sequence elements are defined to start at 1, it was necessary to add the requirement that the LCS is empty when a subscript is zero.

(3) Char-based Edit Distance

Edit Distance is a distance in which insertions and deletions have equal cost and replacements have twice the cost of an insertion. It is thus the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. For instance:

T1: 我喜歡打籃球 (I like to play basketball)

T2: 我討厭打籃球 (I hate to play basketball)

In the text pair, the edit distance is 2 since the character "喜" undergoes one replacement, becoming into "討", while "歡" undergoes one replacement to become into "厭"

(4) Word Length/Difference/Ratio

We use CKIP Autotag to tokenize sentences into every word and calculate the total words. We use string word length difference as a feature to reduce bias on a word length basis. We can use word length ratio to confine a range between 0 and 1. In other words, the word length ratio is used to reduce bias and enhance accuracy.

(5) Word-based Edit Distance

Edit Distance is to measure distance as the number of operations required to transform a string into another where this feature is token-based. For instance:

T1: 我(I)(N) 喜歡(Like)(Vt) 打(to play)(Vt) 球(basketball)(N)

T2: 我(I)(N) 討厭(hate)(Vt) 打(to play)(Vt) 球(basketball)(N)

In this text pair, the edit distance is 1 where the word "喜歡 "(like) transforms into "討厭"(hate).

(6) Noun/Verb Number

We incorporated a feature which calculates noun/verb numbers in a sentence, so we could do a simple comparison in advance.

(7) Word Semantic (Synonym) Similarity

We proposed a semantic feature that uses HIT TYCCL where each word in the TYCCL is assigned an ID and words with same ID are considered synonyms. For example:

Di01A01=世界, 世, 世上, 大地, 天下, 天底下, 全世界, 環球, 全球, 舉世, 中外, 寰宇, 五洲, 海內, 海內外, 五湖四海, 大千世界, 大世界, 普天之下

However, using the original TYCCL for recognizing texts may be too complicated because each synonym has its own ID number, meaning that the more synonyms a word has, the more complicated the queries are. Thus, data may be hard to maintain and update because those synonyms are correlated. Therefore, we do a format conversion to the TYCCL and also added a similarity value for querying.

Formula: TYCCL Scoring Function: $((\tau-\rho) + 1) / \tau$

τ:synonym number ρ: word ranking in synonym list

For example, 世界(World) has 19 synonyms. The synonym list shows that the word 世界 (World) has the highest ranking in the 世界(World) synonym list, so we calculate its similarity score as

$((19-1)+1)/19 = 19/19 = 1$

Thus, the word 世界 (World) has a similarity of 1 in the 世界 (World) synonym list, meaning that it is 100% similar. After calculating word similarity, the results are shown as follows:

世界 Di01A01=| 世 界 :1.0000, Di14C04=| 世風:0.5000, Dd05B03=|領域:0.3333

The results showed the list of synonyms of the word 世界. Each synonym has its ID and its similarity value to 世界.

The results show that if we compare 世界 and 世風 on a syntactic basis, they as appear to be two independent words, but on a semantic basis, 世風 is 50% similar to 世界, which could decrease the experimental bias.

We can also evaluate text fragments via word similarity. We use CKIP Autotag on each text fragments in order to calculate their similarities on a word basis, not on a char basis, and reduce experimental bias. For example:

T1: 車諾比病毒在 1999 年 4 月總共造成超過 200 萬台電腦無法開機

(CIH caused severe boot problems in more than 200 million computers in April, 1999)

T2: 1999 年 4 月車諾比病毒總共造成逾 200 萬台電腦無法開機

(CIH caused severe boot problems over in 200 million computers in April, 1999)

Results show that if we consider only syntactic features, the output would be "Forward" because the T1 String Length is longer than the T2 String Length. However, if we consider semantic features, the output would be "Binary" because the word 超過 (more than) and 逾(over) are synonyms.

(8) WordNet Similarity

We first searched each CKIP token in the WordNet corpus. Once found, we got its Synset. Synonym words share same Synset ID. If two sentences have more Synset ID in common, the more similar these two sentences are. In other words, these two sentences have a higher similarity.

(9) Negation

We proposed a feature which integrated negation words from prior researches into a 52 negation words list. We first detected the negation words number of each text pair. By comparing negation words number to determine whether each text pair is opposite or similar.

(10) Antonym

We proposed a feature which integrated antonym words from prior researches into a 568-antonym-pair list. By first detecting antonym word in each text pair, we could determine if words appeared in the text pair is antonym words or not.

(11) Dependency Parser

We proposed a feature which adopted Stanford Parser to do sentence dependency parsing. In prior research, we found that tree edit distance was common in most dependency parser features. Tree Edit Distance is which the minimum number of edits needed to transform one sentence tree structure into the other , with the allowable edit operations being insertion, deletion, or substitution of a single character.

2.3 Machine Learning

We used LibSVM as the machine learning module. [1] LibSVM provides two tools for enhancing model accuracy: grid.py and fselect.py. These two tools select the best parameters and best features for the model.

3. EXPERIMENTAL RESULTS AND ANALYSIS

We conduct several experiments using various datasets (sample data and develop data) to train and test models, as well as different combinations of features.

3.1 Official RITE-2 Runs

In this section, we describe the algorithms and resources we used for generating the official runs. We also present the official results and discussions.

Table 1.Summary of IMTKU Official Runs

IMTKU BC Subtask Official Runs	Resources	Features
RITE-2-IMTKU-CT-BC-01 RITE-2-IMTKU-CS-BC-01	Bilingual Wordnet, HIT TongYiCiLing, Stanford Parser	Antonym, Negation, Word Based Similarity, Token Based Similarity, Lexical overlap, Text Pair Length, Token Length, WorkNet Similarity, Tree Edit Distance
RITE-2-IMTKU-CT-BC-02 RITE-2-IMTKU-CS-BC-02	Bilingual Wordnet, HIT TongYiCiLing	Antonym, Negation, Word Based Similarity, Token Based Similarity, Lexical overlap, Text Pair Length, Token Length, WorkNet Similarity
RITE-2-IMTKU-CT-BC-03 RITE-2-IMTKU-CS-BC-03	Stanford Parser	All syntactic and semantic features (except Stanford Parser)
RITE-2-IMTKU-CT-MC-01 RITE-2-IMTKU-CS-MC-01	Stanford Parser	Longest Common Substring, Word Length Ratio, Text Length, Similarity between t1 and t2, Tree Edit Distance
RITE-2-IMTKU-CT-MC-02 RITE-2-IMTKU-CS-MC-02	Bilingual Wordnet, HIT TongYiCiLing, Stanford Parser	Integrated Semantic features and Machine Learning Approach
RITE-2-IMTKU-CT-MC-03 RITE-2-IMTKU-CS-MC-03	Bilingual Wordnet, HIT TongYiCiLing	Longest Common Substring, Word Length Ratio, Text Length, Similarity between t1 and t2, Tree Edit Distance
RITE-2-IMTKU-CT-RITE4QA-01 RITE-2-IMTKU-CS-RITE4QA-01	Stanford Parser	Antonym, Negation, Word Based Similarity, Token Based Similarity, Lexical overlap, Text Pair Length, Token Length, WorkNet Similarity
RITE-2-IMTKU-CT-RITE4QA-02 RITE-2-IMTKU-CS-	Bilingual Wordnet, HIT TongYiCiLing	Antonym, Negation, Word Based Siilarity, Token Based Similarity,

RITE4QA-02		Lexical overlap, Text Pair Length, Token Length
RITE-2-IMTKU-CT-RITE4QA-03 RITE-2-IMTKU-CS-RITE4QA-03	HIT TongYiCiLing	Longest Common Substring, Text Length, Text Length Ratio, Antonym, Negation

In the BC and MC subtask, systems were evaluated by macro-F1 Score. In RITE4QA subtask, two kinds of "source factoid QA answer ranking"(SrcRank) are used for evaluation: BetterRanking and WorseRanking. In addition, three factoid QA evaluation metrics are used in RITE4QA subtask: Top1, MRR and Top5.

We list the summary of IMTKU Official Runs for RITE CT BC, CS BC, CT MC, CS MC, CT RITE4QA, CS RITE4QA subtasks in Table 2, 3, 4, 5, 6, 7. Table 2 shows that the best performance of our submitted official runs for RITE CT BC Subtask is 0.659, which is "RITE-2-IMTKU-CT-BC-01". Table 3 shows that the best performance of our submitted official runs for RITE CS BC Subtask is 0.543, which is "RITE-2-IMTKU-CT-BC-03".Table 4 shows that the best performance of our submitted official runs for RITE CT MC Subtask is 0.358, which is "RITE-2-IMTKU-CT-MC-01". Table 5 shows that the best performance of our submitted official runs for RITE CS MC Subtask is 0.273, which is "RITE-2-IMTKU-CS-MC-03".Table 6 shows that the best performance of our submitted official runs for RITE CT RITE4QA Subtask is 0.2570 in MRR, which is "RITE-2-IMTKU-CT-RITE4QA-03". Table 7 shows that the best performance of our submitted official runs for RITE CS RITE4QA Subtask is 0.3377 in MRR, which is "RITE-2-IMTKU-CS-RITE4QA-03".

Table 2. Macro-F1 and Accuracy of IMTKU CT BC Subtask Official Runs

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITE-2-IMTKU-CT-BC-01	**0.659**	**0.663**
RITE-2-IMTKU-CT-BC-02	0.486	0.515
RITE-2-IMTKU-CT-BC-03	0.638	0.643

Table 3. Macro-F1 and Accuracy of IMTKU CS BC Subtask Official Runs

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITE-2-IMTKU-CS-BC-01	0.508	0.540
RITE-2-IMTKU-CS-BC-02	0.501	0.603
RITE-2-IMTKU-CS-BC-03	**0.543**	**0.627**

Table 4. Macro-F1 and Accuracy of IMTKU CT MC Subtask Official Runs

IMTKU MC Subtasks Official Runs	Macro-F1	Accuracy
RITE-2-IMTKU-CT-MC-01	**0.358**	**0.509**
RITE-2-IMTKU-CT-MC-02	0.324	0.366
RITE-2-IMTKU-CT-MC-03	0.194	0.501

Table 5. Macro-F1 and Accuracy of IMTKU CS MC Subtask Official Runs

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITE-2-IMTKU-CS-MC-01	0.239	0.376
RITE-2-IMTKU-CS-MC-02	0.197	0.361
RITE-2-IMTKU-CS-MC-03	**0.273**	**0.402**

Table 6. MRR of IMTKU CT RITE4QA Subtask in WorseRanking Official Runs

IMTKU CT RITE4QA Subtask Official Runs	TOP1	MRR	TOP5
RITE-2-IMTKU-CT-RITE4QA-01	0.1467	0.2258	0.3733
RITE-2-IMTKU-CT-RITE4QA-02	0.1200	0.1984	0.3267
RITE-2-IMTKU-CT-RITE4QA-03	**0.1733**	**0.2603**	**0.4067**

Table 7. MRR of IMTKU CS RITE4QA Subtask in WorseRanking Official Runs

IMTKU CS RITE4QA Subtask Official Runs	TOP1	MRR	TOP5
RITE-2-IMTKU-CS-RITE4QA-01	0.1067	0.1991	0.3867
RITE-2-IMTKU-CS-RITE4QA-02	0.1467	0.2144	0.3600
RITE-2-IMTKU-CS-RITE4QA-03	**0.2800**	**0.3377**	**0.4267**

The confusion matrices of RITE-2 IMKTU CT BC subtask official runs are shown in Table 8, 9, 10. CS BC subtask official runs are shown in Table 11, 12, 13; CT MC subtask official runs are shown in Table 14, 15, 16. CS MC subtask official runs are shown in Table 17, 18, 19, respectively.

Table 8. Confusion Matrix of RITE-2-IMTKU-CT-BC-01 (Accuracy = 0.663)

	Y	N	
Y	**333**	151	484
N	146	**251**	397
	479	402	

Table 9. Confusion Matrix of RITE-2-IMTKU-CT-BC-02 (Accuracy = 0.515)

	Y	N	
Y	**122**	70	192
N	357	**332**	689
	479	402	

Table 10. Confusion Matrix of RITE-2-IMTKU-CT-BC-03 (Accuracy = 0.643)

	Y	N	
Y	**331**	167	498
N	148	**235**	383
	479	402	

Table 11. Confusion Matrix of RITE-2-IMTKU-CS-BC-01 (Accuracy = 0.603)

	Y	N	
Y	**407**	295	702
N	15	**64**	79
	422	359	

Table 12. Confusion Matrix of RITE-2-IMTKU-CS-BC-02 (Accuracy = 0.603)

	Y	N	
Y	**412**	300	712
N	10	**59**	69
	422	359	

Table 13. Confusion Matrix of RITE1-IMTKU-CS-BC-03 (Accuracy = 0.627)

	Y	N	
Y	**413**	282	695
N	9	**77**	86
	422	359	

Table 14. Confusion Matrix of RITE-2-IMTKU-CT-MC-01 (Accuracy = 0.509)

	F	B	C	I	
F	**232**	38	21	37	328
R	0	12	1	30	43
B	14	**109**	4	12	139
C	36	35	**11**	31	113
I	121	27	14	**96**	258
	403	221	51	206	

Table 15. Confusion Matrix of RITE-2-IMTKU-CT-MC-02 (Accuracy = 0.366)

	F	B	C	I	
F	**66**	3	16	236	321
R	7	29	4	6	46
B	0	**0**	6	116	122
C	12	1	**13**	84	110
I	25	0	14	**243**	282
	110	33	53	685	

Table 16. Confusion Matrix of RITE-2-IMTKU-CT-MC-03 (Accuracy = 0.501)

	F	B	C	I	
F	**242**	47	3	33	325
R	3	6	2	42	53
B	15	**98**	2	30	145
C	35	48	**1**	28	112
I	118	27	1	**100**	246
	413	226	9	233	

Table 17. Confusion Matrix of RITE1-IMTKU-CS-MC-01 (Accuracy = 0.376)

	F	B	C	I	
F	**249**	3	18	7	277
B	37	**6**	99	3	145
C	64	5	**34**	3	106
I	161	46	41	**5**	253
	511	60	192	18	

Table 18. Confusion Matrix of RITE-2-IMTKU-CS-MC-02 (Accuracy = 0.361)

	F	B	C	I	
F	**260**	3	8	6	277
B	83	**8**	53	1	145
C	87	5	**8**	6	106
I	193	46	8	**6**	253
	623	62	77	19	

Table 19. Confusion Matrix of RITE-2-IMTKU-CS-MC-03 (Accuracy = 0.402)

	F	B	C	I	
F	**256**	8	13	0	277
B	44	**13**	88	0	145
C	53	7	**45**	1	106
I	133	92	28	**0**	253
	486	120	174	1	

3.2 Discussions

In order to test the consistence of difference dataset, we conduct experiments of cross validation focused on difference datasets. We used the gold standard dataset of NTCIR9 CT BC subtask from organizers to train our machine learning model. Table 20 shows the experimental result of 10 fold cross validation (CV) of development and test datasets, which is development dataset with 421 pairs. In addition, we randomly selected 1000 dataset pairs from BC development dataset and test dataset with the total of 1321 data pairs. The results show that the best performance of cross validation in BC subtask is 73.83%. However, we obtain 72.29% cross validation on the BC test dataset with 1321 pairs by using the same features with same configuration in the machine learning model. In terms of consistence of dataset, we consider that the quality of development dataset is better than test dataset in BC subtask.

It should be noted that there are significant differences in MC subtask labels between NTCIR-9 and NTCIR-10. The main cause of low accuracy in MC subtask is that R is excluded from 5-way labeling subtask in NTCIR-10, made it into 4-way labeling subtask. However, when training the data pairs from NTCIR-9, reverse entailment is included in MC subtask labels. In RITE-2-IMTKU-CT-MC runs, we made serious mistakes that R(Reverse

entailment) was still considered as one of the output labels in MC subtask which should be removed from MC subtask in NTCIR-10 RITE-2, resulting in label inconsistency and low accuracy.

Table 20. Cross Validation of Development and Test datasets of CT BC Subtask

Datasets	10 Fold CV Accuracy
RITE1_CT_dev_bc_g.txt (gold standard) (BC Development Dataset: 421 pairs)	72.21**%**
RITE1_CT_test_bc_g.txt (Random select 1000 pairs from BC Dev+Test Dataset)	**73.83**%
RITE1_CT_dev_test_bc_g.txt (BC Dev+Test Dataset: 421+900 =1321 pairs)	72.29%

4. CONCLUSIONS

In this paper, we proposed a textual entailment system using a hybrid approach that integrate semantic features and machine learning techniques for recognizing inference in text at NTCIR-10 RITE-2 task. We submitted 3 official runs for BC, MC and RITE4QA subtask. In NTCIR-10 RITE-2 task, IMTKU team achieved 0.2570 in MRR evaluation in the CT-RITE4QA subtask and 0.3377 in the CS-RITE4QA subtask.

The contributions of our study are as follows: (1) we proposed an RITE system by integrating semantic features and machine learning approach; (2) the machine learning approach used lexical and semantic features that measure the similarity of text pair to determine whether the text pair entails each other.

5. ACKNOWLEDGMENTS

This research was supported in part by the National Science Council of Taiwan under Grants NSC101-3113-P-032-001 and TKU research grant

6. REFERENCES

[1] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, 2011, pp. 27:1--27:27.

[2] R. Levy and C. D. Manning. 2003. "Is it harder to parse Chinese, or the Chinese Treebank?" *ACL 2003*, pp. 439-446.

[3] CKIP. "CKIP AutoTag," http://ckipsvr.iis.sinica.edu.tw/.

[4] I. Dagan, B. Dolan, B. Magnini, and D. Roth, “Recognizing textual entailment: Rational, evaluation and approaches (vol 15, pg 1, 2009),” *Natural Language Engineering,* vol. 16, 2010, pp. 105-+.

[5] I. Dagan, O. Glickman, and B. Magnini, “The PASCAL recognising textual entailment challenge,” *Machine Learning Challenges,* vol. 3944, 2006, pp. 177-190.

[6] C.-R. Huang, “Sinica BOW: integrating bilingual WordNet and SUMO ontology,” in International Conference on Natural Language Processing and Knowledge Engineering (NLPKE 2003), 2003, pp. 825-826.

[7] J.-J. Mei, Y.-M. Zhu, Y.-Q. Gao, and H.-X. Yin, *TongYiCi CiLin (Chinese Synonym Forest): Shanghai Press of Lexicon and Books*, 1983.

[8] H. Shima. "NTCIR10 RITE-2 Main Page," http://www.cl.ecei.tohoku.ac.jp/RITE-2/doku.php.

[9] Yotaro Watanabe and Yusuke Miyao and Junta Mizuno and Tomohide Shibata and Hiroshi Kanayama and C. -W. Lee and C. -J. Lin and Kohichi Takeda, “Overview of Recognizing Inference in TExt(RITE-2) at the NTCIR-10 Workshop,” in Proceedings of NTCIR-10 Workshop Meeting, Tokyo, Japan, 2013.

[10] D. S. Hirschberg, "Algorithms for the Longest Common Subsequence Problem," *Journal of the Assocrauon for Computing Machinery,* vol. 24:4, pp. 664-675, 1997.