IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-11 RITE-VAL

Min-Yuh Day [1, *], Ya-Jung Wang[1], Che-Wei Hsu[1], En-Chun Tu[1], Shang-Yu Wu[1], Huai-Wen Hsu[1], Yu-An Lin[1], Yu-Hsuan Tai[1], Cheng-Chia Tsai[1]

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

myday@mail.tku.edu.tw, {lucy19890924, kuma0928002898, spoonbill12, a7875815, k120861032002, yuan9090, skystar1020, qooqoo988}@gmail.com

ABSTRACT

In this paper, we describe the IMTKU (Information Management at TamKang University) textual entailment system for recognizing inference in text at NTCIR-11 RITE-VAL (Recognizing Inference in Text). We proposed a textual entailment system using statistics approach that integrate semantic features and machine learning techniques for recognizing inference in text at NTCIR-11 RITE-VAL task. We submitted 3 official runs for BC, MC subtask. In NTCIR-11 RITE-VAL task, IMTKU team achieved 0.2911 in the CT-MC subtask, 0.5275 in the CT-BC subtask; 0.2917 in the CS-MC subtask, 0.5325 in the CS-BC 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, CS_BC, CS_MC)

Keywords

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

1. INTRODUCTION

IMTKU participated in NTCIR-11 RITE-VAL Binary-class (BC) subtask and Multi-class (MC) subtask in Traditional Chinese (CT). We submitted 3 official runs for BC and MC subtask. 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-11 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 in RITE. In all subtasks, a system input is two texts and an output is one of two or four labels [8].

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

T1: 吉力馬札羅山位於坦尚尼亞東北，臨近肯亞邊界，是非洲的最高山。

(Mount Kilimanjaro is located in northeast Tanzania, near the Kenyan border, is the highest mountain in Africa.)

T2: 吉力馬札羅山位於坦尚尼亞

(Mount Kilimanjaro is located in Tanzania.)

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

For the Multi-class Classification (MC) in NTCIR-11 RITE-VAL, given a text pair (t1, t2), a system detects entailment in more detail. The class would be yes (forward entailment, backward entailment), 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: 水蘊草適合生長在營養及光線充足的環境中。

(Yun grass growing in the water for nutrition and well-lit environment.)

T2: 水蘊草適合生長在營養及缺乏光線的環境中。

(Water Yun grass growing in nutrition and a lack of suitable light environment.)

The system output for the BC subtask is "NO" for the above T1, T2 pair. Further, the system output for the MC subtask is "Contradiction" t1 and t2 contradict each other.

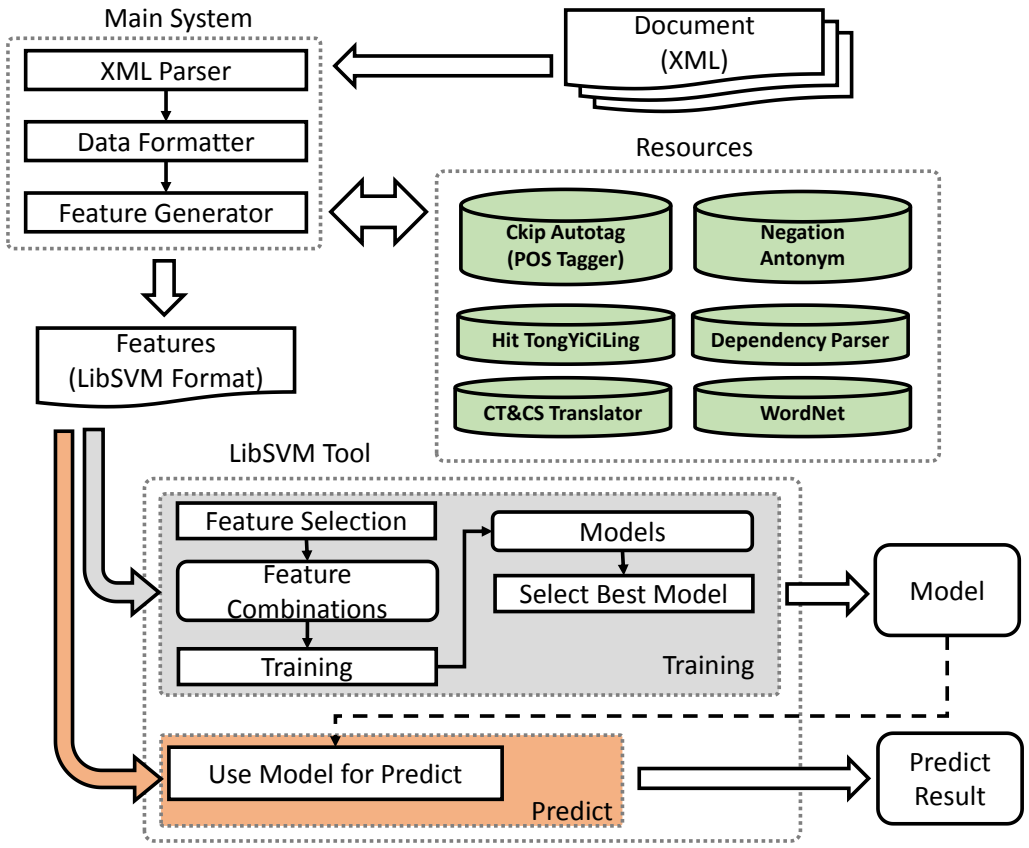

Figure 1. System Architecture

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.

According to the task description of NTCIR-11 RITE-VAL 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 contradict each other)"; I means "independence (t1 can not to inferred t2, and t2 can not to inferred t1)". The evaluation of RITE system is the accuracy of labels predicted [8].

2. SYSTEM ARCHITECTURE

The architecture of IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-11 RITE-VAL is describe in Figure 1.

2.1 Main System Module

Main System Module consists of three sub-module: XML Parser, Data Formatter, Feature Generator.

2.1.1 XML Parser

The given XML datasets has been parsed using XML Parser. The XML Parser parsed IDs and text pairs from XML datasets of the RITE-VAL corpus for analysis.

2.1.2 Data Formatter

We necessary to unify the data format. A word may be expressed in different ways. For example, 2014 may be written "2014 年" or "二零一四年".

2.1.3 Feature Generator

We had designated 14 semantic and syntactic features:

String Length, String Length Difference, String Length Ratio,

Longest Common Substring Sequence, Char-Based Edit Distance, Word Length, Word Length Difference, Word Length Ratio, Word-Based Edit Distance.

(1) T1/T2 String Length/Length Difference/Ratio

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. Basic syntactic approach we adopted as a feature.

(2) Longest Common Substring Sequence(LCSS)

We use Longest Common Substring Sequence [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: 我喜歡去購物 (I like to go shopping)

T2: 我討厭去購物 (I hate to go shopping)

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) T1/T2 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) 去(to go)(Vt) 購物(shopping)(N)

T2: 我(I)(N) 討厭(hate)(Vt) 去(to go)(Vt) 購物(shopping)(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:

Hj19B01=參加, 入, 入夥, 加入, 加盟, 在, 投入, 出席, 進入

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, 參加(Correct) has 22 synonyms. The synonym list shows that the word 參加 (Correct) has the highest ranking in the 參加(Correct) synonym list, so we calculate its similarity score as

((9-1)+1)/9 = 9/9 = 1

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

參加 Ed12A01=| 參 加 :1.0000, Di14C04=| 出席:0.8667

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

The results show that if we compare 參加 and 出席 on a syntactic basis, they as appear to be two independent words, but on a semantic basis, 出席 is 86% similar to 參加, 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 16115-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 LibSVM Tool

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-VAL 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
RITEVAL-IMTKU-CS-SVBC-01 RITEVAL-IMTKU-CT-SVBC-01	Bilingual Wordnet	Antonym, Negation, String Length and Length Ratio and Length Difference, LCSS, Char-Based Edit Distance, Noun/Verb Number, Wordnet Similarity and Ratio and Minimum
RITEVAL-IMTKU-CS-SVBC-02 RITEVAL-IMTKU-CT-SVBC-02	Bilingual Wordnet, HIT TongYiCiLing	Antonym, Negation, Word Based Similarity, LCSS , Word Length and Length Ratio and Length Difference, Word-Based Edit Distance, Noun/Verb Number, Wordnet Similarity and Ratio and Minimum
RITEVAL-IMTKU-CS-SVBC-03 RITEVAL-IMTKU-CT-SVBC-03	HIT TongYiCiLing	Antonym, Negation, String Length and Length Ratio and Length Difference, LCSS, Char-Based Edit Distance, Noun/Verb Number, Word Length and Length Ratio and Length Difference, Word-Based Edit Distance
RITEVAL-IMTKU-CS-SVMC-01 RITEVAL-IMTKU-CT-SVMC-01	Bilingual Wordnet, HIT TongYiCiLing	Antonym, Negation, Word Based Similarity, LCSS , Word Length and Length Ratio and Length Difference, Word-Based Edit Distance, Noun/Verb Number, Wordnet Similarity and Ratio and Minimum
RITEVAL-IMTKU-CS-SVMC-02 RITEVAL-IMTKU-CT-SVMC-02	HIT TongYiCiLing	Antonym, Negation, String Length and Length Ratio and Length Difference, LCSS, Word Semantic Similarity
RITEVAL-IMTKU-CS-SVMC-03 RITEVAL-IMTKU-CT-SVMC-03	HIT TongYiCiLing	Antonym, Negation, String Length and Length Ratio and Length Difference, LCSS, Word Semantic Similarity, Word Length and Length Ratio and Length Difference, Char-Based Edit Distance, Word-Based Edit Distance

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

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITEVAL-IMTKU-CT-SVBC-01	**0.4403**	**0.5063**
RITEVAL-IMTKU-CT-SVBC-02	**0.4218**	**0.5275**
RITEVAL-IMTKU-CT-SVBC-03	**0.4271**	**0.4425**

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

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITEVAL-IMTKU-CS-SVBC-01	**0.4177**	**0.5275**
RITEVAL-IMTKU-CS-SVBC-02	**0.4254**	**0.5317**
RITEVAL-IMTKU-CS-SVBC-03	**0.428**	**0.5325**

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

IMTKU MC Subtasks Official Runs	Macro-F1	Accuracy
RITEVAL-IMTKU-CT-SVMC-01	**0.1901**	**0.2911**
RITEVAL-IMTKU-CT-SVMC-02	**0.1848**	**0.2894**
RITEVAL-IMTKU-CT-SVMC-03	**0.1963**	**0.2808**

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

IMTKU BC Subtask Official Runs	Macro-F1	Accuracy
RITEVAL-IMTKU-CS-SVMC-01	**0.1902**	**0.2917**
RITEVAL-IMTKU-CS-SVMC-02	**0.1867**	**0.2908**
RITEVAL-IMTKU-CS-SVMC-03	**0.1954**	**0.2792**

The confusion matrices of RITE-VAL IMKTU CT BC subtask official runs are shown in Table 6, 7, 8. CS BC subtask official runs are shown in Table 9, 10, 11; CT MC subtask official runs are shown in Table 12, 13, 14. CS MC subtask official runs are shown in Table 15, 16, 17, respectively.

Table 6, Table 7, Table 8 are three experimental results on CT BC subtask. In Table 6 and 7, compared with our results and the answer of official runs, we discovered that our most answers were indicated to Y(s), but the actual answers were the half of N and the half of Y. In Table 8, the total number of Y and N were more average than Table 6 and 7, but the accuracy was the lowest than the others.

Table 6. Confusion Matrix of RITE-VAL-IMTKU-CT-BC-01 (Accuracy = 0.5063)

	Y	N	
Y	**510**	90	600
N	502	**98**	600
	1012	188	

Table 7. Confusion Matrix of RITE-VAL-IMTKU-CT-BC-02 (Accuracy = 0.5275)

	Y	N	
Y	**573**	27	600
N	540	**60**	600
	1113	87	

Table 8. Confusion Matrix of RITE-VAL-IMTKU-CT-BC-03 (Accuracy = 0.4425)

	Y	N	
Y	**364**	236	600
N	433	**167**	600
	797	403	

Table9, Table10, Table11 are three results CS BC that shows that we have predicted the Y, N, compared with the actual answer, show that we predicted the answer most emphasis on Y.

Table 9. Confusion Matrix of RITE-VAL-IMTKU-CS-BC-01 (Accuracy = 0.5275)

	Y	N	
Y	**577**	23	600
N	544	**56**	600
	1121	**79**	

Table 10. Confusion Matrix of RITE-VAL-IMTKU-CS-BC-02 (Accuracy = 0.5317)

	Y	N	
Y	**577**	23	600
N	539	**61**	600
	1116	84	

Table 11. Confusion Matrix of RITE-VAL-IMTKU-CS-BC-03 (Accuracy = 0.5325)

	Y	N	
Y	**576**	24	600
N	537	**63**	600
	1113	**87**	

The results of CT MC and CS MC were Table 12, 13, 14, 15, 16, and 17. Those numerical results were very close to each other. On the comparison of the total numbers, B was the most one than C and F. Then C was more than F. According the results, we found that the results of B, C, and F were the same problem which we cannot predict any answers for I.

Table 12. Confusion Matrix of RITE-VAL-IMTKU-CT-MC-01 (Accuracy = 0.2911)

	F	B	C	I	
F	**10**	90	200	0	300
B	8	**281**	11	0	300
C	44	197	**59**	0	300
I	9	187	104	**0**	300
	71	**755**	**374**	**0**	

Table 13. Confusion Matrix of RITE-VAL-IMTKU-CT-MC-02 (Accuracy = 0.2894)

	F	B	C	I	
F	**7**	90	203	0	300
B	5	**282**	13	0	300
C	31	210	**59**	0	300
I	6	187	107	**0**	300
	49	**769**	**382**	**0**	

Table 14. Confusion Matrix of RITE-VAL-IMTKU-CT-MC-03 (Accuracy = 0.2808)

	F	B	C	I	
F	**16**	78	206	0	300
B	28	**261**	11	0	300
C	78	162	**60**	0	300
I	23	169	108	**0**	300
	145	**670**	**385**	**0**	

Table 15. Confusion Matrix of RITE-VAL-IMTKU-CS-MC-01 (Accuracy = 0.2917)

	F	B	C	I	
F	**10**	92	198	0	300
B	8	**281**	11	0	300
C	44	197	**59**	0	300
I	9	189	102	**0**	300
	71	**759**	**370**	**0**	

Table 16. Confusion Matrix of RITE-VAL-IMTKU-CS-MC-02 (Accuracy = 0.2908)

	F	B	C	I	
F	**8**	90	202	0	300
B	5	**282**	13	0	300
C	31	210	**59**	0	300
I	7	187	106	**0**	300
	51	**769**	**380**	**0**	

Table 17. Confusion Matrix of RITE-VAL-IMTKU-CS-MC-03 (Accuracy = 0.2792)

	F	B	C	I	
F	**16**	79	205	0	300
B	29	**260**	11	0	300
C	80	161	**59**	0	300
I	25	167	108	**0**	300
	150	**667**	**383**	**0**	

Table 18 shows in cross-validation the one of AntonymCount feature is the highest, because we have a 16115-antonym-pair list in this feature.

Table 18. RITE-VAL-IMTKU-All-BC-CT-Features-Accuracy

Feature Name	**Cross Validation Accuracy (CT)**
Features_CharLengthT1	61.2737%
Features_CharLengthT2	60.9294%
CharLengthDifference	59.7246%
CharLengthRatio	63.6833%
LCSSequence	60.0688%
WordLengthT1	63.5112%
WordLengthT2	60.7573%
WordLengthDifference	62.4785%
WordLengthRatio	63.5112%
CharBasedED	60.9294%
WordBasedEDC	63.5112%
NounCount	63.1670%
VerbCount	63.6833%
WordSemainticSimilarity	60.9294%
WordNetSimilarity	63.1670%
WordNetSimilarityRatio	63.1670%
WordNetSimilarityMin	63.1670%
NegationCountCard	63.3391%
AntonymCount	**64.8881%**
Dependency Parser	59.5525%

3.2 Discussions

In MC subtask, the serious problem is that the numerical answer of I is zero.

We discover the problem of MC performance is the imbalance data. Because the official training dataset provided the imbalance numerical data which caused us cannot reach the balance results, as Backward (B) has 222, Forward (F) has 148, Contradiction (C) has 152, Independence (I) only 59 by LibSVM tool. Therefore, we randomly picked out 50 pairs in each groups of B, F, C, and I. We utilize 200 pairs to be the data of the training model and the experimental result of Confusion Matrix of RITE-VAL-IMTKU-CT-MC shows in Table 19:

Table 19. Confusion Matrix of RITE-VAL-IMTKU-CT-MC-EXPERIMENT-01 (Accuracy = 0.2433)

	F	B	C	I	
F	178	23	89	10	300
B	151	36	90	23	300
C	91	96	38	75	300
I	146	54	60	40	300
	566	**209**	**277**	**148**	

The experimental result shows that the balanced data could affect the classification of the LibSVM. However, the accuracy of the experimental result is even lower than our official runs because of the feature selective problem.

The biggest difference between NTCIR-10 and NTCIR-11 is that the discussions of the issues of the balance and imbalance data.

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

Dataset	5 Fold CV Accuracy
RITE_VAL_CT_dev_bc_g.txt (gold standard) (BC Development Dataset: 581 pairs)	64.0275%
RITE_VAL_CT_test_bc_g.txt (BC Test Dataset: 1200 pairs)	56.25%
RITE_VAL_CT_dev_test_bc_g.txt (BC Dev+Test Dataset: 581+1200 =1781 pairs)	55.5306%

4. CONCLUSIONS

In this paper, we proposed a textual entailment system using a statistics approach that integrate semantic features and machine learning techniques for recognizing inference in text at NTCIR-11 RITE-VAL task. We submitted 3 official runs for BC, MC subtask. In NTCIR-11 RITE-VAL task, IMTKU team achieved 0.2917 in CS_MC evaluation and 0.2911 in CT_MC evaluation.

Our study of the contributions are as follows: (1) we proposed an RITE-VAL system by integrating semantic the features combinations and machine learning approach; (2) the use of machine learning methods in answer to the proportion of training data will affect the predicted proportion of answers; (3) the balanced data could affect the classification of the LibSVM.

5. 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.