III&CYUT Chinese Textual Entailment Recognition System for NTCIR-11 RITE-VAL

Shih-Hung Wu[1*], Li-Jen Hsu[1], Hua-Wei Lin[1], Pei-Kai Liao[1]
[1]Chaoyang University of Technology,
Taichung, Taiwan, R.O.C
{[*]shwu, s10027042, s10027072, s10027024}@cyut.edu.tw

Liang-Pu Chen[2], Tsun Ku[2]
[2]Institute for Information Industry,
Taipei, Taiwan, R.O.C
eit@iii.org.tw, cujing@gmail.com

ABSTRACT
Textual Entailment (TE) is a critical issue in natural language processing (NLP). In this paper, we report how our hybrid approach system works in NTCIR-11 RITE-VAL task [15]. We attended both Fact Validation (FV) and System Validation (SV) subtasks for Chinese. In the SV subtask, we also attended both binary classification (BC) and multi-classification (MC). For the SV BC subtask, our system detects eleven special cases for the input pairs, and uses twelve SVM classifiers to do classification. The results then are integrated as the system report. For the SV MC subtask, we also train four SVM classifiers for the Bidirectional, Forward, Independence, and Contradiction. The results are integrated by rules. For the FV subtask, our system searches the Wikipedia to find the top one T1 and decides the entailment relation to T2 by rules.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Natural language understanding, Textual Entailment

General Terms
Experimentation

Keywords
Chinese Textual Entailment, linguistic features, classifiers

Team Name
III&CYUT

Subtasks/Languages
RITE-VAL (BC, MC, FV), Simplified Chinese, Traditional Chinese

External Resources Used
LIBSVM, Lucene

1. INTRODUCTION
TE can be briefly defined as: "Given a pair of sentences (T1, T2), a program has to decide whether the information in T2 can be inferred by T1". TE can be used in various applications, such as question answering system, information extraction, information retrieval, and machine translation [2]. Traditional approaches to TE are based on the semantic and syntactic similarity of the words in the sentences. Once a system can decide whether T1 entails T2 or not, it can be regarded as an information filter to help users find useful information.

Based on our machine learning approach in NTCIR-10 RITE-2 [1][11], we improve our system with more rules to achieve a hybrid system. We attend all the Chinese subtasks: BC, MC, and FV and submit 5, 4, and 5 runs respectively in formal run for both traditional Chinese (CT) and simplified Chinese (CS) datasets.

In the BC subtask, given a sentence pair (T1, T2) a system has to decide whether T1 can infer T2. If T1 can infer T2, then the system outputs "Y", otherwise, the output is "N".

In the MC subtask, given a sentence pair (T1, T2) a system has to decide whether T1 has any entailment relation with T2 or not. If T1 can infer T2, and T2 cannot infer T1, the system outputs "F (forward)". If T1 can infer T2, and T2 can also infer T1, the system outputs "B (bidirectional)". When T1 and T2 cannot both be true, it is a contradiction, the system outputs "C (contradiction)". If T1 cannot find any relation to T2, the system outputs "I (Independence)". Table 1 shows the examples of sentence pairs in the training corpus.

Fact validation is a new subtask. Given a sentence T2, the system has to find the possible related sentence T1 from a large corpus, such as Wikipedia. Then the system has to decide whether T1 and T2 has one of the following three entailment relations: "E(Entailment)", "C (Contradiction) ", or "U (Unknown) ". Table 2 shows the examples of relations in the training corpus.

Table 1. Examples of four entailment relations in MC subtask

Type	Example
Forward	T1：水蘊草適合生長在營養及光線充足的環境中。
	T2：水蘊草生長需要充足的光線。
Bidirectional	T1：東協自由貿易區(ASEAN Free Trade Area，簡稱 AFTA)於 1992 年提出。
	T2：1992 年提出的東協自由貿易區(ASEAN Free Trade Area)簡稱 AFTA。
Contradiction	T1：吉力馬札羅山的部分山區被指定為吉力馬札羅國家公園，並登錄為世界遺產。
	T2：吉力馬札羅山的部分山區被指定為吉力馬札羅國家公園，並退出世界遺產。
Independence	T1：手語並不是世界共通的
	T2：手語並不是亞洲共通的

Table 2. Examples of three entailment relations in FV subtask

Type	Example
Entailment	T2：歐巴馬是美國的一位總統。
	第 44 任美國總統(詞條:巴拉克·歐巴馬)

Contradiction	T2：孫中山禁止三民主義。
	他提出《三民主義》等政治綱領(詞條:孫中山)
Unknown	T2：柴契爾在1992年冊封了一位終身貴族。
	無法證實(詞條:柴契爾夫人)

2. Research Methodology

There are various approaches to the TE in previous works, ranging from theorem proving to using linguistic-resource such as WordNet [3]. Our hybrid approach is based on both machine learning and rules. After we made more observation on the dataset, we decide to detect special cases by rules first and then train SVM classifiers [4] for each class. Our special cases are a subset of the unit test in system validation training data types because some semantic classes in the unit test are hard to detect for computer.

3. System architecture

Our system can deal with both CS and CT datasets. Figure 1 shows the system architecture of our SV system. The basic components are “preprocessing”, “word segmentation”, ”special case filter”, ”sub-systems for special cases”, “feature extraction”, and “SVM” classifier.

Figure 2 shows the system architecture of our FV system. The basic components are “preprocessing”, “Indexing”, “Search”, and “Filter”.

3.1 Preprocessing

Here we describe our preprocessing module, which replaces some terms in T1 and T2 before further processing.

3.1.1 Normalizations

The normalizations in preprocessing include several small modifications on the terms that we regard as the same term. For example, **“**葉望輝(Stephen J. Yates)**”** represents the name of a person in both Chinese and English, and our system will normalize them into one common representation. Also, there are many different formats to represent time in Chinese as shown in Table 3, and our system will normalize them into the same representation. After the normalization, sentences with the same meaning but with different terms will be aligned easier. Thus, it can help our system to find features with higher accuracy.

Table 3. Examples of time expressions [11]

Type	Time expressions in text
Chinese only	一九九七年二月廿三日
Full type digit with Chinese	１９９７年２月２３日
Half type digit with Chinese	1997年2月23日
Digit only	1999-05-07
Duration	1999年延長至2001年

3.1.2 Background knowledge matching and substitution

The first part of our preprocessing system is to normalize synonym terms. The necessary knowledge can be collected from Wikipedia, HowNet[5], or TongYiCi CiLin [6].

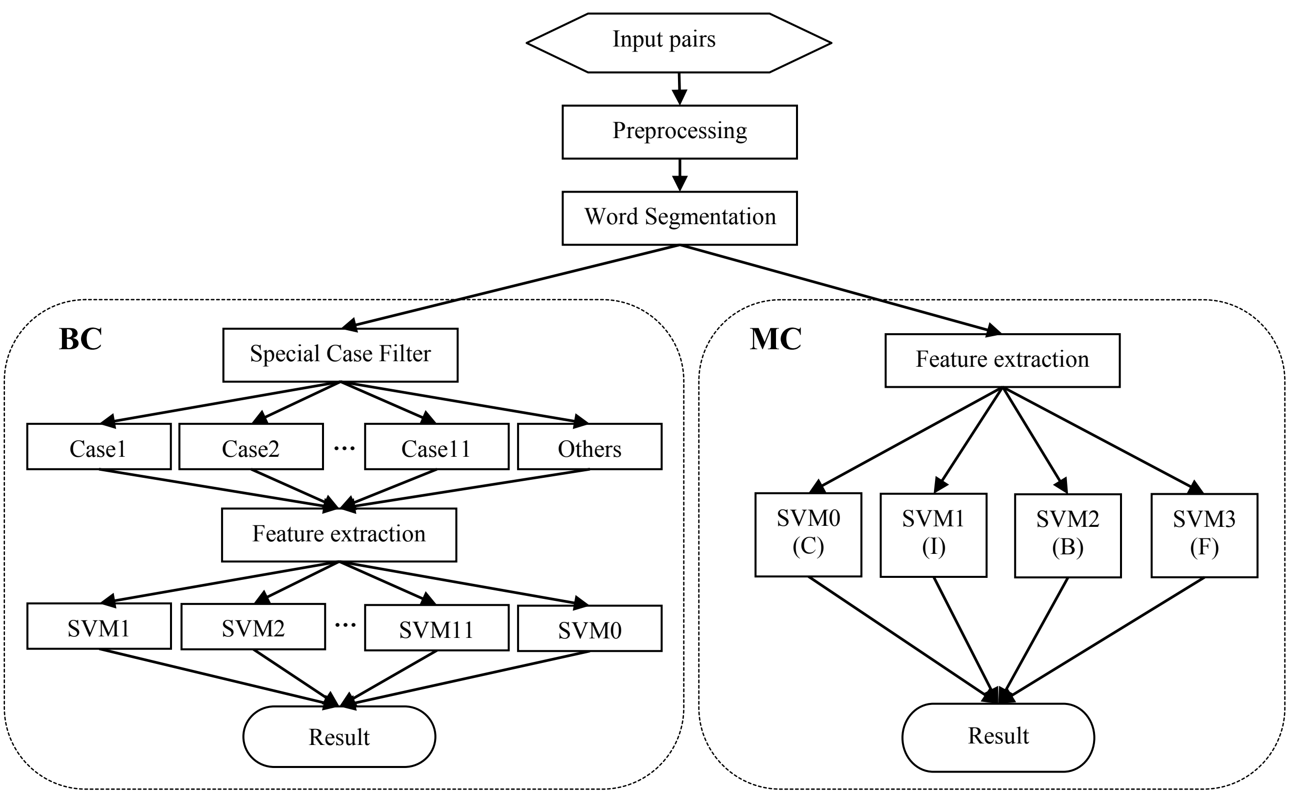

Figure 1. SV System flowchart (BC/MC)

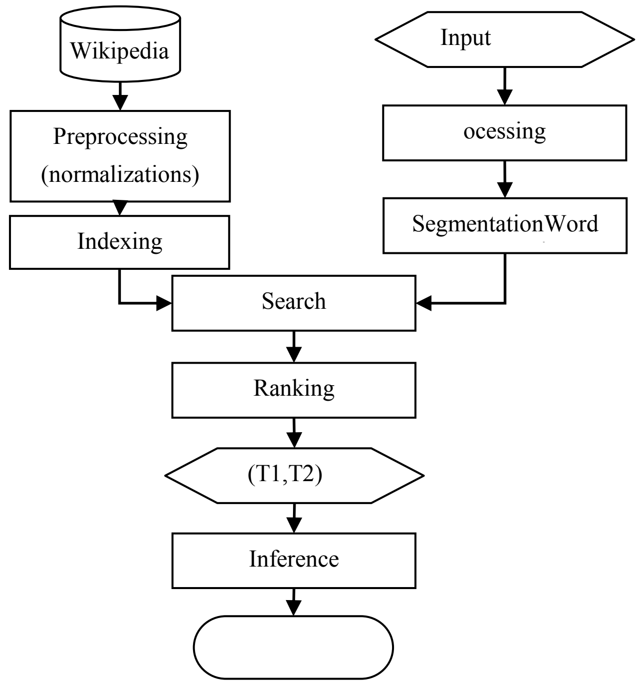

Figure 2. FV System flowchart

3.2 Word Segmentation and Chinese character conversion

Word segmentation of Chinese is necessary and is done by the open source program Jcseg [8] in our system, we use the 1.9.5 version. Jcseg can handle both traditional and simplified Chinese. It only requires another lexicon bases. There are quite large lexicon bases available.

3.2.1 Stop Word

Jcseg also provides the stop word filtering function. When the program does the word segmentation, it can also filter out punctuations, adverbs, or particles[1]. We use the stop word filtering function in our FV subtask. The stop word list is available from Internet, such as the one provided by Harbin Institute of Technology [7].

3.3 Feature extraction

In this section, we briefly introduce the features used in SVM, which are the same as those we used in previous work.

Table 4. Features Used in Our System

No	Feature
1	unigram_recall
2	unigram_precision
3	unigram_F_measure
4	log_bleu_recall
5	log_bleu_precision
6	log_bleu_F_measure
7	difference in sentence length (character)
8	absolute difference in sentence length (character)
9	difference in sentence length (term)
10	absolute difference in sentence length (term)

[1] http://sites.google.com/site/kevinbouge/stopwords-lists

The first three features are the numbers of common terms in both T1 and T2. The next three features are the BLEU scores [13][14]. The rest four features are the numbers and differences of sentence lengths of T1 and T2.

3.4 Special cases in RITE-VAL Chinese dataset

The dataset of SV subtask provides 28 specific language phenomena, which are far more than the special case analysis in our previous work. This year, we implement a more detailed special case analyzer to detect 11 of them, the easier ones.

To detect the special cases, our system first deletes the common terms in a T1 and T2 pair, and applies the following rules to the rest different terms.

Here we list eleven cases:

Case1. Abbreviation

The same terms might appear in both T1 and T2 in their original form or the abbreviation form. Our rule to detect abbreviation is to compare the terms, if a character in one sentence is included in a term in which the same character order appears in another sentence, and then it is considered as an abbreviation.

Case 2.Antonym

Antonym means that a term cannot be true at the same time in the opposite situation. Antonym is a strong indicator of contradiction. Our system uses an antonym list to detect whether there is an antonym or not. The list consists of 814 pairs of antonym.

Case 3.Apposition

Apposition is a way to give an alternative name to an entity, which refers to the same thing. Our system finds apposition when it is detecting between T1 and T2 an additional term following right away the original entity. This rule is not very accurate since the additional term might not be an apposition.

Case 4.Case_alternation

Case alternation means T1 and T2 shared the same verb and the voice of the verb is different from each other. Our system detects the case alternation by finding whether the "被(bei)" structure is applied.

Case 5.Exclusion:quantity

Quantity information exclusion means the numbers are different in T1 and T2. Our system will normalize the numbers before matching the values.

Case 6.Exclusion:temporal

Temporal information exclusion means the time expressions are different in T1 and T2. Our system will also normalize the time expressions before matching the values. If the unit of time is missing in one of the pair, our system will use the unit in the other sentence as the default unit.

Case 7.Negation

If the only different term is a negation term, then T1 and T2 are considered as a contradiction pair. Our system can detect whether or not the different terms contain such Chinese character as "無(wu)"、"不(bu)"、"非(fei)"、"沒(mei)"、"未(wuei)"、"禁(jieng)" or not.

Case 8.Quantity

The only difference between T1 and T2 is the expression of the same number.

Case 9.Scrambling

If there is no different term found, our system will regard T1 and T2 as scrambling.

Cases 10.Synonymy:lex

Synonym means a term has the same meaning with the other one. This is a strong indicator of entailment. Our system uses a synonym list to detect whether there is an synonym or not. The list consists of 630 pairs of synonym.

Case 11.Temporal

The only difference between T1 and T2 is the expression of the same time.

Table 5 lists examples of each case.

Table 5. Examples of special cases

Case	Example
Case 1	喬治·盧卡斯最著名的作品是《星際大戰》和《法櫃奇兵》系列。
	喬治·盧卡斯最著名的作品是《星戰》和《法櫃》系列。
Case 2	水蘊草適合生長在營養及光線充足的環境中。
	水蘊草適合生長在營養及缺乏光線的環境中。
Case 3	乾物女是來自於日本的流行語。
	乾物女這個詞來自於日本的流行語。
Case 4	有時候，生產商會在鉛筆的一端裝上橡皮擦。
	有時候，鉛筆的一端會被生產商裝上橡皮擦。
Case 5	有時候，生產商會在鉛筆的一端裝上橡皮擦。
	有時候，生產商會在鉛筆的兩端裝上橡皮擦。
Case 6	約瑟夫·傅立葉於 1768 年 3 月 21 日在法國約訥省歐塞爾出生。
	約瑟夫·傅立葉於 3 年 1768 月 21 日在法國約訥省歐塞爾出生。
Case 7	水蘊草（Egeria densa）別名蜈蚣草。
	水蘊草（Egeria densa）別名並非蜈蚣草。
Case 8	水蘊草可生長在水深 4 公尺以內的水域，總長可以長達 2 公尺。
	水蘊草可生長在水深四公尺以內的水域，總長可以長達兩公尺。
Case 9	歷史上沒有吉力馬札羅山火山噴發的記錄。
	記錄上沒有吉力馬札羅山火山噴發的歷史。
Case 10	手語主要使用者是失聰者。
	手語主要使用者是聾人。
Case 11	約瑟夫·傅立葉是十九世紀法國數學家、物理學家。
	約瑟夫·傅立葉是 19 世紀法國數學家、物理學家。

3.5 Support vector machine

The SVM tool used in our system is the LIBSVM [11], which can be used to train both binary-class classifier and multiple-class classifier. In the formal runs, sometimes we use only binary class classifier. In BC runs, we separate the input pairs into special cases and train a binary class classifier for each special case. In the MC runs, we use four binary class classifiers in the four classes, and the final result is integrated by rules.

3.6 Indexing and Search

To find the T1 fast in FV subtask, we use open source search engine software Lucene [9] to build the index of Wikipedia. The content of Wikipedia is separated into sentences. We use the keywords in T2 to search the top 100 sentences for further ranking described in the following sub-section.

Each T1 candidate in the Lucene search result will be given a score by the default scoring formula:

$$\mathrm{score}(q,d) = \mathrm{coord}(q,d) \times queryNorm(q) \times \sum_{t\ in\ d} \big(tf(t\ in\ d)\ \times idf(t)^2 \times t.getBoost \times norm(t,d)\big) \quad (1)$$

where coord(*q, d*) is the number of matched terms. And the queryNorm(*q*) is normalized weight of the query terms in *q*.

$$queryNorm(q) = \frac{1}{\sqrt{sum\ Of\ Squared\ Weights}} \quad (2)$$

The *tf(t in d)* is the square root of the term frequency of the term *t* in document *d*.

$$tf(t\ in\ d) = \sqrt{frequwncy} \quad (3)$$

The idf(t) is the inverse document frequency.

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq+1}\right) \quad (4)$$

Where numDocs is the total number of documents, and docFreq is the document frequency of term *t*. The 1 added is to prevent from getting a zero idf(*t*). The $t.getBoost$ is the search weight, the default value is 1. And the *norm(t, d)* is defined as:

$$norm(t,d) = \mathrm{lengthNorm} \times \prod_{field\ f\ in\ d\ named\ as\ t} f.boost \quad (5)$$

Which takes the index weight *f.boost* into account.

3.7 Ranking T1

To find a better T1, we adopt the well-known TFIDF [10] to find the top T1 out of the 100 candidates. The importance of a term in T2 is calculated according to the TFIDF formula, where the relative frequency is defined as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k.j}} \quad (6)$$

where $n_{i,j}$is the frequency of the term in document d_j, and $\sum_k n_{k.j}$ is the sum of the frequency of all the terms. The higher the TF is, the more important the term is. IDF of a term is to calculate the inverse document frequency. The formula is defined as:

$$idf_i = \log\frac{|D|}{1+|\{j: t_i \in d_j\}|} \quad (7)$$

|D| is the number of total documents (100 in our experiment). $|\{j: t_i \in d_j\}|$ is the number of documents contains term t_i .1 is

added to prevent from dividing to zero. The importance of a term in a document is calculated according to the TFIDF formula:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{8}$$

Our system ranks each retrieved sentence according to the score:

$$\text{Score} = 0.2 \times score(q, d) + tfidf_{i,j} \tag{9}$$

which is a weighted sum of the retrieval score of Lucene and the TFIDF values of keywords in both T1 and T2. In our experiments, we find that the 0.2 is a proper weight for the training data.

3.8 Inference

After ranking the search result, our system finds the top T1, then checks the entailment relation of the T1, T2 pair with the following inference rules. Since the T1 is a sentence with the same keywords as T2, our system focuses on the minor differences.

1. Negation detection.
   By detecting the number of certain Chinese negation terms as discussed in 3.4, our system can detect single negation and double negation [13][14]. For example, a double negation might be ”不得不出席 (No show is not allowed.)”, which is equal to a positive statement.
2. Antonym detection
   By detecting the antonym as discussed in 3.4, our system can detect possible contradiction.
3. The number of common terms
   The more common terms they have, the more likely that T1 and T2 are expressing the same meaning.
4. Synonym detection
   In the FV-05 run, we consider theT1 and T2 pair as forward entailment if synonym is detected in both T1 and T2.

When negation or antonym is detected, our system will label the pair as “C”. Otherwise, our system just counts the number of common terms. If the ratio is above 50%, our system will label the pair as “E”. If the ratio is lower than 50%, our system will label it as “U”. If the ratio is further lower than 33%, our system will also label it as “C”. These heuristic rules are designed by human after a careful observation of the system results on training set.

4. Experiment Result

In this section, we will report the experiment results on test set.

4.1 Formal run results

The MacroF1 of formal run results of our system are shown in Table 6 and Table 7. The setting of each run is as follows:

BC-01: Use only the forward pairs in the RITE-VAL SV training set.

BC-02: Use both the forward and backward pairs in the RITE-VAL SV training set.

BC-03: As BC-01 with special cases analysis。

BC-04: As BC-02 with special cases analysis。

MC-01: Use both the forward and backward pairs in NTCIR10 and NTCIR11 training set with special cases (case2、case5、case6、case7、case10)。

MC-02: Use both the forward pairs in NTCIR10 and NTCIR11 training set for training a multi-class classifier.

MC-03: Training set as MC-01 run, for training a multi-class classifier.

MC-04: Training set as MC-01 run, for training four binary-class classifiers. The result is the integration of the result of the classifiers.

FV-01: Use the top 1 return sentence of Lucene as T1.

FV-02: Re-rank the returned sentences with TFIDF and Lucene score.

FV-03: Re-rank the returned sentences with only TFIDF.

FV-04: Re-rank the returned sentences with the TF and Lucene score.

FV-05: As FV-02. With synonym detection.

Table 6. Formal run MarcoF1 of our system in RITE-VAL SV task

RUN	BC		MC	
	CS	CT	CS	CT
RUN01	34.32	34.46	37.64	35.43
RUN02	52.60	51.99	31.06	31.27
RUN03	56.03	56.00	32.95	32.95
RUN04	**56.75**	**56.24***	**40.41**	40.52
RUN05	-	-	40.32	**40.54***

Table 7. Formal run results MarcoF1 of our system in RITE-VAL FV task

Run	CS	CT
RUN01	36.76	38.04
RUN02	38.78	**39.51***
RUN03	37.00	37.72
RUN04	36.89	37.69
RUN05	**38.93***	39.36

Table 8.Confusion Matrix of FV-02

gold\system	E	C	U
E	167	12	43
C	129	20	52
U	72	31	87

4.1.1 *Formal run error analysis*

Table 8 is the confusion matrix of our RITE-VAL FV-02 run. We can find that our system tends to misclassify C and U into E, there are 201 cases like this.

We fund that there are five major causes of our system errors:

Case1: Substitution failures

One major function of our preprocessing module is to substitute terms with the same meaning. The substitution failures are caused by the difficulty to implement all the necessary rules on one side and still maintain the necessary language resources on the other side.

Case2: Lack of background knowledge

Inference based on knowledge is necessary for many difficult pairs. However our system only uses very simple rules to detect.

Case3: Negation detection error

Our negation detection rules are quite simple. They only check whether the certain characters appear or not without further analyzing the formula of negation.

Case4 : Synonym/antonym detection error

In the formal run, many errors are caused by the fail of synonym/antonym detection. Since our lists of synonym/antonym are quite short, further extension of the list is needed to improve the detection result.

Case5 :T1 candidate search failure

In the formal run, our system fails to find the best T1 candidates and causes some errors. In the formal run FV-02, among the 339 errors of our system, we find that 188 cases are caused by the T1 candidate search fail.

Table 9 Lists examples of the five error types.

Table 9. Error case examples in RITE-VAL FV subtask

Type	Example pairs
Case1	T1: 出芽生殖，是一種**無性繁殖方式**，親代藉由細胞分裂產生子代，但是子代並不立即脫離母體，而與母體相連，繼續接受母體提供養分，直到個體可獨立生活才脫離母體。
	T2: 出芽生殖的生殖方式，可使子代的**基因**發生**重組**。
Case2	T1: 為紀念瓦特的貢獻，國際單位制中的**功率**單位以瓦特命名。
	T2: 瓦特是**電壓**的單位
Case3	T1: 基於它們的物理和化學特性，幾乎所有元素周期表上的化學元素都可被分類為金屬或非金屬；但也有一些特性介於金屬與非金屬之間的元素，稱為類金屬。
	T2: 就導電性，元素大體上可分為金屬、類金屬及非金屬三大類
Case4	T1: 由於人類辨識顏色的基因是來自 X 染色體，故若母親為色盲者，則其所生的兒子必定是色盲。
	T2: 男子的色盲是直接由父親遺傳而來
Case5	T1: 與其起源地點的文化傳統密切相關。
	T2: 恆星的顏色與其大小密切相關

4.2 DISCUSSION

From the formal run results, we can find some facts. In the BC subtask, Table 6 shows that the special case detection helps to increase the performance up to 17.53 and 18.28 for CS and CT respectively, which is very huge. With more training set, the system performance increases further 4.15 and 4.25 for CS and CT respectively. In the MC subtask, Table 10 shows that the performance of our system with more binary classifier (MC-05) is better than with single multi-class classifier (MC-03). In the MC-01 run, we added the special case analysis to the system; the flowchart is shown in figure 3. Table 11 shows that the contradiction can be detect correctly in this run.

Table 10. System performance of using single multi-class classifier or four binary classifiers

RUN	CS		CT	
	MacroF1	Acc.	MacroF1	Acc.
MC-03	32.95	42.17	32.95	42.17
MC-05	40.32	43.08	**40.54***	43.33

Table 11. The number of correct pairs in MC-CS

RUN	B	F	C	I
MC-01	118	141	**211**	23
MC-02	271	205	1	23
MC-03	278	185	1	42
MC-04	168	170	129	41
MC-05	160	165	162	30

In FV subsection, Table 12 shows that re-ranking the search results with both TFIDF and Lucene scores gives better result. The performance of the contradiction class is lower than the other two classes. These results suggest that we need to improve the detection of contradiction.

Table 12. FV-CT runs results

RUN	MacroF1	Acc.	E-F1	C-F1	U-F1
FV-01	38.04	42.90	56.03	**17.78**	40.33
FV-02	**39.51**	**44.70**	**56.61**	15.15	46.77
FV-03	37.72	42.41	54.58	14.87	43.70
FV-04	37.69	44.05	53.82	9.48	**49.76**
FV-05	39.36	44.54	55.61	14.79	47.69

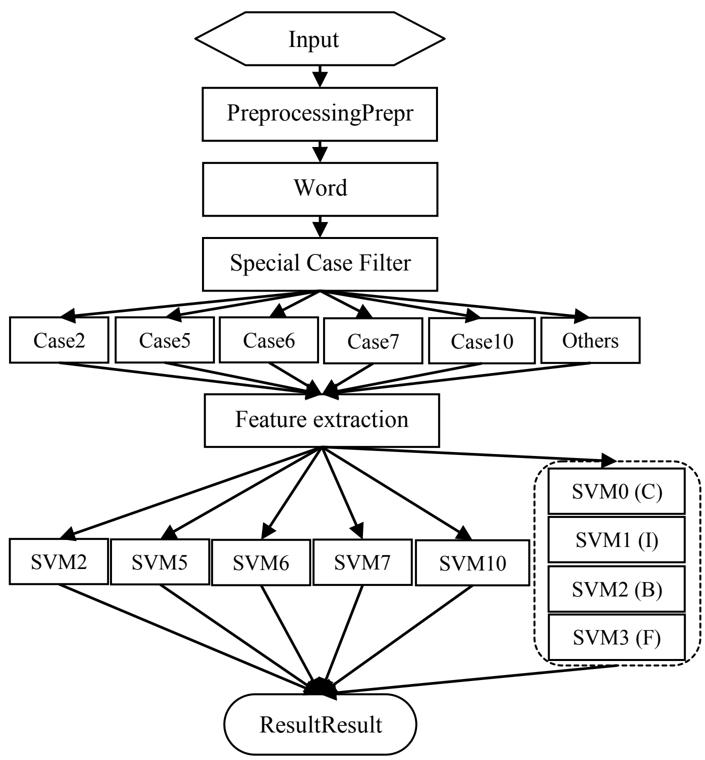

Figure 3. SV-MC-01 System flowchart

4.3 ADDITIONAL RUNS

In the additional runs, we build SVM classifiers to see if the classifiers can help the FV task. The first SVM classifier is a multi-class (MC) classifier and the test result is shown in Table 13. The Entailment recognition result is improved, while the performance of contradiction and the unknown is declined.

Table 13. Confusion Matrix of the MC classifier for FV

gold\sys	E	C	U
E	**206**	0	16
C	178	2	21
U	173	2	15

Then we build three binary class (BC) classifiers for the Entailment, Contradiction, and Unknown classes. The result is the combination of the output from the three classifiers. Experimental result is shown in Table 14. The Entailment recognition result is further improved, while the performance of contradiction and unknown is further declined.

Table 14. Confusion Matrix of the three BC classifiers for FV

gold\sys	E	C	U
E	**212**	0	10
C	190	0	11
U	182	0	8

Table 15 shows the comparison between the formal run and additional run on the FV result. The rule-based approach in formal run performs better than both the MC and BC SVM classifier approaches. We also find that it is hard to recognize contradiction and the unknown in our system.

Table 15. Comparison of the formal run and additional run on the FV results

RUN	MacroF1	Acc.	E-Rec	C- Rec	U- Rec
Formal run	39.51	44.70	75.23	9.95	45.79
MC	23.40	36.38	92.79	0.99	7.89
Three BC	19.97	35.88	95.49	0.0	4.21

5. CONCLUSIONS AND FUTURE WORKS

This paper reports our system in the NTCIR-11 RITE-VAL CT-BC, CT-MC, CS-BC, CS-MC and FV subtasks. This year we use more SVM classifiers instead of more features. Our new approach shows good improvement compared with previous works. Though the approach is promising there is some future work needed to be done.

More types and higher accurate special case detection: From the formal run result, we can find that separating into special cases can improve overall accuracy. In the SV dataset, there are 28 language phenomena; however, currently our system can only detect 11 of them. We need more linguistic resources to detect all of them.

Increase the size of training set: It is a common sense in machine learning that the more training set is the higher the performance is. Our study also shows the same direction. Currently the size of training set is too small. We need to enlarge the training set for better results.

More linguistic resources are needed. Our system performs badly on entity recognition. We need a better way to improve our NER result.

FV contradiction detection needs more study. In our formal run result in RITE-VAL, our approaches to FV shows low performance on the contradiction detection. We need to integrate some keyword for contradiction detection on T1 search to get better result.

ACKNOWLEDGMENTS

This study was conducted under the "Online and Offline Integrated Smart Commerce Platform (1/4)" of the Institute for Information Industry, which is subsidized by the Ministry of Economic Affairs of the Republic of China. This study was also supported by the Ministry of Science under the grant numbers MOST103-2815-C-324-010-E and MOST103-2221-E-324-017.

REFERENCES

[1] Shan-Shun Yang, Shih-Hung Wu, Liang-Pu Chen, Hung-Sheng Chiu, and Ren-Dar Yang, “Entailment Analysis for Improving Chinese Recognizing Textual Entailment System”, Computational Linguistics and Chinese Language Processing, Vol. 18, No. 4, December 2013, pp. 1-16.

[2] Yongping Ou, Changqing Yao, “Recognize Textual Entailment by the Lexical and Semantic Matching”, Computer Application and System Modeling, 2010 International Conference on V2-500 -504

[3] Christiane Fellbaum, “WordNet: An Electronic Lexical Database”, The MIT Press, 1998

[4] Prodromos Malakasiotis, Ion Androutsopoulos, “Learning textual entailment using SVMs and string similarity measures”, In Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 42–47, Prague, Czech Republic, 2007.

[5] Q. Lin, S.J. Li, Word Similarity Computing Based on How-net, Computational Linguistics and Chinese Language Processing ，Vol.7,No.2,August 2002,pp.59-76

[6] Dong-Bin Hua, Jun Ding,” Study on Similar Engineering Decision Problem Identification Based on Combination of Improved Edit-Distance and Skeletal Dependency Tree with POS”, In Systems Engineering Procedia Volume 1, Pages 406–413, 2011.

[7] 哈爾冰工業大學社會計算與訊息檢所研究中心. 哈工大停用詞表[EB/OL]. [2013-05-30]. http://ir.hit.edu.cn/

[8] Jcseg 1.9.4, http://git.oschina.net/lionsoul/jcseg

[9] Lucene 4.6.1, https://lucene.apache.org/

[10] Gerard Salton, Christopher Buckley. “Term-weighting approaches in automatic text retrieval”, Information Processing & Management Vol.24,No.5,pp.513-523, 1988

[11] Shih-Hung Wu, Shan-Shan Yang, Liang-Pu Chen, Hung-Sheng Chiu and Ren-Dar Yang. “CYUT Chinese Textual Entailment Recognition System for NTCIR-10 RITE-2”, in Proceedings of the NTCIR-10 workshop, Tokyo, Japan, 18-21 .June, 2013.

[12] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/lib

[13] 陳東源; 葉淑華, 中文否定詞在句型組成結構中傳播意義的研究, 空大人文學報 No.3, 1994, p131-149.

[14] 何永清, 現代漢語語法新探,台灣商務出版社,2005.

[15] Suguru Matsuyoshi, Yusuke Miyao, Tomohide Shibata, Chuan-Jie Lin, Cheng-Wei Shih, Yotaro Watanabe and Teruko Mitamura, "Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task", in Proceedings of the 11th NTCIR Conference, Tokyo, Japan, 2014.