Proceedings of the 10th NTCIR Conference, June 18-21, 2013, Tokyo, Japan

WSD Team's Approaches for Textual Entailment Recognition at the NTCIR10 (RITE2)

Daiki Ito
School of Science and Engineering, Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
ito_d@yama.info.waseda.ac.jp

Masahiro Tanaka
School of Science and Engineering, Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
tanaka_m@yama.info.waseda.ac.jp

Hayato Yamana
Faculty of Science and Engineering, Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
yamana@yama.info.waseda.ac.jp

ABSTRACT

In this paper, we describe the WSD team's approaches to textual entailment recognition task (RITE) at NTCIR-10[1], a conference held in June 18-21, 2013, at NII in Tokyo, Japan, and present experimental results for three Japanese subtasks, called "Binary Class" (BC), "Multi Class" (MC) and "Entrance Exam BC" (ExamBC). Our approach employs two supervised learning techniques: support vector machine (SVM) and logistic regression (LR). For the binary classification subtasks (BC and ExamBC), we propose hand-coded rules to classify text into entailment or non-entailment categories, while for the multi-classification subtask (MC), we used the bidirectional features for the texts. The best performance in three runs achieved precision of 80.66% in BC subtask, 69.53% in MC subtask and 67.86% in ExamBC subtask. We won second place for BC and MC, and third place for ExamBC.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Text Analysis, Language parsing and understanding.

General Terms

Theory, Languages

Keywords

Textual Entailment, Machine Learning

Team Name

[WSD]

Subtasks

[Binary Class][Multiple Class][Entrance Exam BC]

1. Introduction

This paper describes our Textual Entailment Recognition system constructed from scratch to participate in the Japanese BC, MC and ExamBC subtasks at NTCIR-10 RITE-2[1].

Natural language is relativity disordered compared to artificial languages like programming languages because natural language has many different ways of saying the same thing, like paraphrases and metaphors. Machine learning methods have been widely used to conduct textual entailment recognition in Japanese texts. Thus, many teams working on NTCIR-9 RITE tasks adopted machine learning methods; however, it was difficult to find features upon whose basis entailment relationships in less systematic natural languages could be decided. Therefore, we propose a new method which classifies texts into categories; texts that seem to have entailment relationships, those texts that seem to have non-entailment relationships, and other unclassified texts. To detect entailment relationships in these texts, we compare predicate-argument structures. We also use limitation expressions, number expressions, and named entities in texts to detect non-entailment relationships. Finally, we adopt machine learning for texts which not all the above approaches apply.

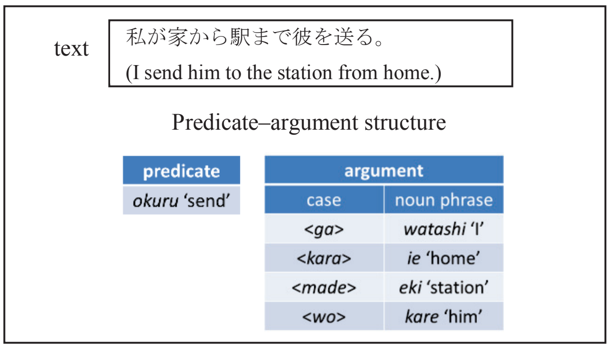
This paper is organized as follows. We describe our techniques and methods in Section 2. Then, we describe our approaches to the BC, MC and ExamBC subtasks in Section 3. In Section 4, we present experimental results for these subtasks. We consider our results in Section 5, and offer concluding remarks in Section 6.

2. Hand-Coded Rules

We adopt hand-coded rules as a way to detect (non-)entailment relationships between texts. In this section, we explain those rules and techniques as we have used them.

2.1 Predicate-Argument Structure

A predicate-argument structure identifies semantic relations between predicates and related arguments. Figure 1 shows an example. In Japanese, arguments are usually composed of a "case marker" and a "noun". In Figure 1, *okuru* 'send' is the predicate and other words are arguments. For example, "*<made>*, *eki*" is an argument with a particle *made* and a noun phrase *eki* 'station'. Henceforth, we use these terms.


text
私が家から駅まで彼を送る。
(I send him to the station from home.)
Predicate–argument structure

predicate	argument	
	case	noun phrase
okuru 'send'	*<ga>*	*watashi* 'I'
	<kara>	*ie* 'home'
	<made>	*eki* 'station'
	<wo>	*kare* 'him'

Figure 1 An example of predicate-argument structure

Some previous research uses predicate-argument structure for textual entailment recognition. For example, Shibata and Kurohashi [2] in NTCIR-9 RITE, constructed predicate-argument structures for two texts, "t1" and "t2," and assumed that t1 entails t2 if the predicate and all arguments in t2 are include those in t1 at a rate above, some threshold. In some cases, they identify entailment when the predicate-argument structure of t2 has an "is-a relation" with t1.

Odani et al. [3] proposed a normalized predicate-argument structure generated from texts consisting of different surface forms but with the same meaning. Table 1 shows examples of this normalization. They judged there to be an entailment relation between texts if and only if predicate-argument structure of t1 entails that of t2 based on normalized predicate-argument structure.

Table 1 Examples of normalized predicate-argument structure

Example texts	Normalized predicate-argument structure
数学を勉強する 'study math'	勉強する 'study' <ヲ> 数学 (<*wo*> math)
数学の勉強をする 'do math study'	
風が扉を閉めた 'wind closed a door'	閉めた 'closed' <ガ> 風 (<*ga*> wind) <ヲ> 扉 (<*wo*> door)
風で扉が閉まった 'a door was closed by wind'	
車と自転車が衝突した 'car and a bicycle are collided'	衝突した 'collided' <ガ> 車 <*ga*> car <ガ> 自転車 <*ga*> bicycle
車が自転車と衝突した 'a car collided with a bicycle'	

In this way, entailment relations between predicate-argument structures often imply entailment relations between texts. We adopt this idea and assume that texts have an overall entailment only if their predicate-argument structures have this relation.

2.2 Limitation Expressions

Matsuyoshi et al. [4] developed *"Tsutsuji"*, a Japanese dictionary of functional expressions (components of sentences). Some of these expressions express limitation and exception. We have made a list of "limitation expressions" from *"Tsutsuji"* and *"Weblio thesaurus"* [14]. "*Weblio thesaurus*" is an online thesaurus. Figure 2 shows examples of limitation expressions.

だけ	のみ	唯一	除く
'just'	'just'	'only'	'except'

Figure 2 Examples of limitation expressions

We assumed that it is difficult to replace limitation expressions with other expressions. Therefore, we set a new rule; if t2 contains limitation expressions but t1 does not, t1 is not likely to entail t2. Figure 3 shows an example.

t1: 女性がかかる癌の中でも１位を占めているのは乳癌である。
(Breast cancer is the most frequently occurred cancer that women suffer from.)
t2: 乳癌は女性<u>しか</u>かからない。
(<u>Only</u> women suffer from a breast cancer.)

Figure 3 An example of a limitation expression in context

2.3 Number Expressions

If number expressions are mismatched between t1 and t2, there is no entailment relation between t1 and t2. Thus, we classify number expressions by meaning. This classification is based on normalizeNumexp [5]. normalizeNumexp is a tool for normalizing numerical/temporal expression. The categories are defined as shown in Table 2.

Table 2 Categories of number expressions

Category	Examples
AbsTime	1989 年 1 月 7 日 "January 7th, 1989"
Duration	一年間 "during a year"
RelTime	一世紀前 "a century ago"
Numerical	1,000 円 "1,000 yen"

We define "mismatch" as follows: where

- t2 contains number expressions, but t1 does not contain; or
- the range of such expressions in t1 is a subset of that in t2.

2.4 Named-Entities

If t1 does not contain a named-entity included in t2, we assume that t1 does not entail t2. We consider synonyms on the basis of Japanese WordNet[6]. Figure 4 shows an example of named-entity mismatch. Here, t1 does not contain "Tokugawa Ieyasu", a named-entity included in t2.

t1: 清浄院は加藤清正の継室である。
(Shoujouin is the second wife of Kato Kiyomasa.)
t2: 清浄院は、徳川家康の養女である。
(Shoujouin is an adopted daughter of Tokugawa Ieyasu.)

Figure 4 An example of named-entity mismatch

3. System Description

3.1 Machine Learning Method and Features

In this subsection, we describe SVM and LR, the machine learning methods we used. We adopted LIBSVM [7] as a tool for SVM and the LIBLINEAR [8] as a tool for LR. We used different features for each subtask as described later in the subtask explanation.

- Morpheme Overlap ratio (MO)

We adopt MeCab [9] to split t1 and t2 into morphemes. We define the morpheme overlap score between t1 and t2 by equation (1).

$$MO(t1, t2) = \frac{numMorheme(t1 \cap t2)}{numMorpheme(t2)} \qquad (1)$$

In equation (1), $numMorpheme\ (t2)$ represents the number of morphemes in t2, and $numMorheme(t1 \cap t2)$ represents the number that appears in both t1 and t2.

- Clause Overlap ratio (CO)

We adopt CaboCha [10] to split t1 and t2 into clauses and define clause overlap by equation (2).

$$CO(\mathrm{t1}, \mathrm{t2}) = \frac{numClause(t1 \cap t2)}{numClause(t2)} \qquad (2)$$

In equation (2), $numClause(t2)$ represents the number of clauses in t2 and $numClause(t1 \cap t2)$ represents the number of clauses that appear in both t1 and t2. We used the strings of analysis results from CaboCha as clauses.

- Sequence Match Ratio (SMR)

Sequence match ratio is a function that calculates the similarity between two strings. Here, we use SMR in standard library of Python. It is defined by equation (3).

$$SMR(t1, t2) = \frac{numChara(t1 \cap t2) \times 2.0}{numChara(t1) + numChara(t2)} \qquad (3)$$

In equation (3), $numChara(t1)$ represents the number of characters in t1, and $numChara(t1 \cap t2)$ represents the number of characters that appear in both t1 and t2.

- Jaro Distance (JD)

Jaro distance is a measure of similarity between two strings, taken by considering the characters' sequential order.

$$JD(t1, t2) = \begin{cases} 0 \; [m = 0] \\ \frac{1}{3}\left(\frac{m}{len(t1)} + \frac{m}{len(t2)} + \frac{m-t}{m}\right) \; [m \neq 0] \end{cases} \quad (4)$$

In equation (4), m is the number of matched characters appearing in the same sequence in both t1 and t2 and t is the number appearing in both t1 and t2 but with a different sequence.

- 4-gram of Character Overlap ratio (4gramCO)

We adopt 4-gram as a feature based on our experimental results, in which it achieved the highest precision. A 4-gram of character overlap ratio is one of our features defined by equation (5).

$$4gramCO(t1, t2) = \frac{num4gramC(t1 \cap t2)}{num4gramC(t2)} \quad (5)$$

In equation (5), $num4gramC(t2)$ represents the number of 4-grams of characters in t2, and $num4gramC(t1 \cap t2)$ represents the number of 4-grams of characters that appear in both t1 and t2.

- 4-gram of Morpheme Overlap ratio (4gramMO)

The 4-gram of morpheme overlap ratio is the 4-gram version of MO. We define it as equation (6).

$$4gramMO(t1, t2) = \frac{num4gramM(t1 \cap t2)}{num4gramM(t2)} \quad (6)$$

Here, $num4gramM(t2)$ is the number of 4-grams of morphemes in t2, and $num4gramM(t1 \cap t2)$ is the number of 4-grams of morphemes that appear in both t1 and t2.

- Modification Overlap ratio (ModO)

In Japanese texts, there are modification relationships exist from a clause to other clauses, except the last clause. We define ModO by equation (7).

$$ModO(t1, t2) = \frac{numModification(t1 \cap t2)}{numModification(t2)} \quad (7)$$

In equation (7), $numModification(t2)$ represents the number of modification relationships (the number of clauses - 1), and $numModification(t1 \cap t2)$ represents the number of the same modification relationships that appear in both t1 and t2. Here, the same modification relationship means that the source clause and its destination clause are same in t1 and t2.

- Noun Overlap ratio (NO)

Noun overlap is defined as equation (8).

$$NO(t1, t2) = \frac{numNoun(t1 \cap t2)}{numNoun(t2)} \quad (8)$$

In equation (8), $numNoun(t2)$ represents the number of nouns in t2, and $numNoun(t1 \cap t2)$ represents the number of nouns that appear in both t1 and t2.

- Longest Common Subsequence string Rate (LCSR)

We define LCSR by equation (9). Here, *lcs* represents the longest common subsequence between t1 and t2. LCSR is normalized length of *lcs* by dividing length of t2 shown as $len(t2)$.

$$LCSR(t1, t2) = \frac{len(lcs)}{len(t2)} \quad (9)$$

- Subject-Object-Predicate overlap ratio (SOPO)

To extract a subject, a predicate, and object from t1 and t2, we considered the leftmost noun connected to *ha* or *ga* as the subject, a rightmost verb or adjective as a predicate, and a clause dependent on the object as an object clause. We define an element as "none" if no words correspond to the rules described above, and subjects, objects, or predicates the same if they are "none". Across t1 and t2. SOPO is defined by equation (10) and normalized to a 0-1 range by dividing by 3 (=one subject, one object, and one predicate for each of t1 and t2).

$$SOPO(t1, t2) = \frac{numSOP(t1 \cap t2)}{3} \quad (10)$$

In equation (10), $numSOP(t1 \cap t2)$ is the total number of same subjects, objects, and predicates appearing in both t1 and t2.

- Noun-Adjective-Verb Overlap ratio (NAVO)

NAVO is defined by equation (11).

$$NAVO(t1, t2) = \frac{numNAV(t1 \cap t2)}{numNAV(t2)} \quad (11)$$

In equation (11), $numNAV(t2)$ represents the total number of nouns, adjectives and verbs in t2, and $numNAV(t1 \cap t2)$ represents the number of the same nouns, adjectives and verbs (i.e., those with the same basic form) that appears in both t1 and t2.

- Feeling Polarity Margin Score (FPMarS)

Following Takamura et al. [11], we set a feeling polarity of 1.0 for the most positive word and -1.0 for the most negative, and define feeling polarity score of text, and define feeling polarity score of text (FPST) by equation (12) as the sum of feeling polarities of all words in the text.

$$FPST(t) = \sum_{w_i \in t} FPS(w_i) \quad (12)$$

In equation (12), $FPS(w_i)$ represents the feeling polarity score of w_i, and t means a text. Finally, FPMarS is defined by equation (13).

$$FPMarS(t1, t2) = |FPST(t1) - FPST(t2)| \quad (13)$$

- Feeling Polarity Multiple Score (FPMulS)

FPMulS is defined by equation (14).

$$FPMulS(t1, t2) = FPST(t1) \times FPST(t2) \quad (14)$$

- Cost of Tree Edit Distance (TED)

TED is calculated based on edit operations, namely insertion, deletion, and/or substitution. There are similar to as edit operation for strings. We use Zhang and Shasha's algorithm [12] to calculate TED, and make trees (tree-1 from t1 and tree-2 from t2) by considering a clause as anode and using a modification relationship. Figure 5 shows how to make trees from texts.

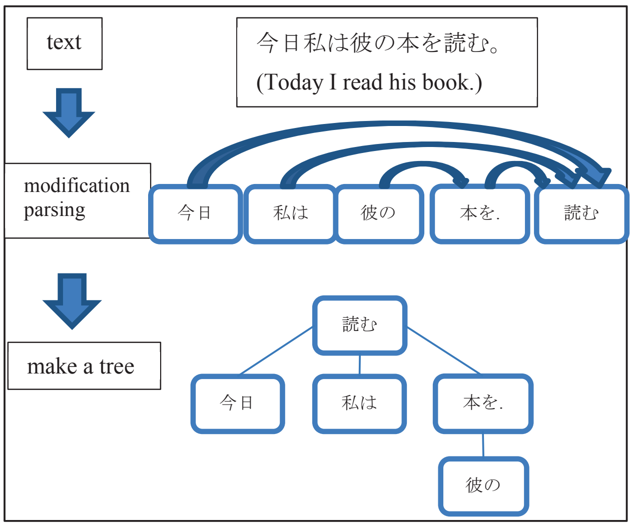


Figure 5 An example of making a tree from a text

We define the cost of tree edit-distance as equation (15).

$$CTed(t1, t2) = \frac{ted}{max(numClause(t1), numClause(t2))} \quad (15)$$

In equation (15), ted is the edit distance between tree-1 and tree-2, and $numClause(t1)$ represents the number of clauses in t1.

- Named-Entity Mismatch

We define a function f as an indicator to compute the named-entity mismatch feature of t1 and t2; f(t1, t2) = 1 if and only if t2 contains a named-entity that does not appear in t1. In other case, f(t1, t2) = 0.

- Number Expressions Mismatch

We define a function g as an indicator to compute the number expressions mismatch feature of t1 and t2; g(t1, t2) = 1 if and only if there is a mismatch between the expressions in t1 and t2. In other case, g(t1, t2) = 0.

3.2 BC Subtask

This subsection describes the method we used for the BC subtask.

3.2.1 Hand-Coded Rules

We adopted "predicate-argument structure," "limitation expressions," "number expressions" and "named-entity" as rule-based classifiers.

3.2.2 Features

We adopt features for a machine learning classifier as below:

- MO, NO, 4gramC, LSCR, JD

3.3 MC Subtask

This subsection describes our method used for the MC subtask. It is difficult to make rules that specify one entailment relation here, because there are four choices in MC, as opposed to two in BC. In MC, we adopt two methods, as shown below.

- Adopting tree edit distance as a feature

At NTCIR-9 RITE, Tsuboi et al. [13] used tree edit-distance as a feature for machine leaning in MC, improving precision to 47.1% from 35.9% and showing TED's effectiveness in MC.

- Adopting similarity features in both directions

For multi-class entailment recognition, there exist many methods using machine learning that adopt similarity between t1 and t2 as a feature. Most methods assume that similarity between t1 and t2 is same as between t2 and t1; however, we argue that it is more effective to adopt two opposite-directional similarities, a feature called *both directions*. Thus, we adopt the similarity of t1 to t2 and that of t2 to t1 for machine leaning, to investigate the possibility that t1 entails t2 or vice versa.

Here, we explain this method using MO. When the number of morphemes in t1 is 10, that of t2 is 7, and the number of morphemes that appear in both t1 and t2 is 3, MO normalized by t2 becomes $\frac{3}{7}$, calculated by equation (1), and MO normalized by t1 becomes $\frac{3}{10}$. We use features normalized by t2 in the BC and ExamBC subtasks; in MC, we add features normalized by t1.

3.4 ExamBC Subtask

This subsection describes our method for ExamBC. We used both hand-coded rules and a machine learning based classifier.

3.4.1 Hand-Coded Rules

We adopted "comparison of limitation expressions" as a rule-based classifier. Based on prior experimental results with NTCIR-9 RITE datasets, precision became higher but F-value to decline if we adopted "number-expressions mismatch" and "named-entity mismatch" as rule-based classifiers. Therefore, we adopted them not as rules but as features of machine learning.

3.4.2 Features

We adopted three features for the ExamBC.

- NO, named-entity mismatch, number expressions mismatch

3.5 System Details for Each Run

In RITE tasks, participants can submit (run) up to three versions. The WSD team submitted three runs for each subtask (BC, MC, and ExamBC). Table 15 shows the details for each run.

4. Experiments and Results

This section describes each subtask experiment and their results.

4.1 Datasets

Table 3 provides statistical information the BC and ExamBC subtask datasets, and, Table 4 on the MC subtask datasets.

Table 3 Statistical information of BC subtask data sets

Dataset	Y	N	Total
BC dev	240	371	611
BC test	256	354	610
ExamBC dev	210	300	510
ExamBC test	173	275	448

Table 4 Statistical information of MC subtask data sets

Dataset	F	B	C	I	Total
MC dev	207	83	65	193	548
MC test	205	70	61	212	548

4.2 Experimental Results for the BC Subtask

4.2.1 RITE2 Results

For BC, we adopt three classification phases as we defined below.

Phase 1: Detecting texts that seem to have entailment relationships.

Phase 2: Detecting texts that seem to have non-entailment relationships.

Phase 3: Classification by SVM

Table 5 shows the number of problems, number of correct answers, and precision of each classification phase in each run.

Table 5 Precisions of rule-based classifiers

Rule		No. of detected text pairs	No. of correct answers	Precisions (%)
Phase1	Predicate-argument structure	7	5	71.4
Phase 2	Limitation Expressions	6	6	100.0
	Number-expressions	66	61	92.4
	Named-entity	93	77	82.8
Total		172	149	86.6

Table 6, Table 7, and Table 8 show the confusion matrices for each run while Table 16 shows the results for the BC subtask and Figure 6 shows the system architecture of WSD-JA-BC-03.

Table 6 Confusion matrix for WSD-JA-BC-01

		Gold		Total
		Y	N	
System	Y	180	49	229
	N	76	305	381
Total		256	354	610

Table 7 Confusion matrix for WSD-JA-BC-02

		Gold		Total
		Y	N	
System	Y	180	48	228
	N	76	306	382
Total		256	354	610

Table 8 Confusion matrix for WSD-JA-BC-03

		Gold		Total
		Y	N	
System	Y	194	56	250
	N	62	298	360
Total		256	354	610

4.2.2 Additional Experiments and Results

To confirm the usefulness of dividing the classifiers into three phases, we built classifiers that adopted our heuristic for hand-coded rules as new features in machine learning, instead of deterministic rules based on WSD-JA-BC-03 system and experimented. Table 9 shows the results of the experiment.

Table 9 Precision using hand-coded rules deterministically or as features for machine learning

	Precision (%)
Using hand-coded rules as deterministically	80.66
Using hand-coded rules as features for machine learning	80.33

4.3 Experiment Results for the MC Subtask

4.3.1 RITE2 Results

Table 10, Table 11, and Table 12 show the confusion matrices for each run. Table 17 shows the results for the MC subtask.

Table 10 Confusion matrix for WSD-JA-MC-01

		Gold				Total
		F	B	C	I	
System	F	158	6	20	35	219
	B	13	56	11	14	94
	C	0	0	0	1	1
	I	34	8	30	162	234
Total		205	70	61	212	548

Table 11 Confusion matrix for WSD-JA-MC-02

		Gold				Total
		F	B	C	I	
System	F	160	8	21	37	226
	B	12	53	10	9	84
	C	2	1	0	0	3
	I	31	8	30	166	235
Total		205	70	61	212	548

Table 12 Confusion matrix for WSD-JA-MC-03

		Gold				Total
		F	B	C	I	
System	F	160	7	19	34	220
	B	12	56	13	13	94
	C	0	0	0	0	0
	I	33	7	29	165	234
Total		205	70	61	212	548

4.3.2 Additional Experiments and Results

To confirm the usefulness of bidirectional similarity as described in Subsection 3.3, we tested three ways of normalizing features.

Way 1: Use features normalized by t1 elements

Way 2: Use features normalized by t2 elements

Way 3: Use both

Table 13 shows the results in WSD-JA-MC-03, which results the effectiveness of bidirectional similarity. To confirm the usefulness of TED, Table 13 also shows TED.

Table 13 Precisions by (non-) use TED

Normalized features	Precision (%)	
	Not using TED	Using TED
t1	62.23	64.60
t2	56.57	66.61
t1, t2	68.98	69.53

4.4 Experimental Results for the ExamBC Subtask

Our proposed system for ExamBC has two classification phases, as follows.

Phase 1: Detecting texts that seem to have non-entailment relationship

Phase 2: Classification by machine learning

Table 14 shows number of problems, number of correct answers and precision of each classification phase in each run.

Table 19, Table 20, and Table 21 show the confusion matrix of each run. Table 18 shows the results for the ExamBC subtask.

Table 14 Precision of rule based classifiers

Rules		Num. of detected text pairs	Num. of correct answers	Precisions (%)
Phase1	Limitation expressions	3	3	100.0
Total		3	3	100.0

Table 15 Details for each run

Run ID	Hand-coded rules	Machine learning	Features	Special notes
WSD-JA-BC-01	Predicate-argument structure, limitation expressions, number expressions, named-entity	SVM	MO, NO, 4gramCO, LSCR, JD	
WSD-JA-BC-02	Predicate-argument structure, limitation expressions, number expressions, named-entity	LR	MO, NO, 4gramCO, LSCR, JD	
WSD-JA-BC-03	Predicate-argument structure, limitation expressions, number expressions, named-entity	SVM	MO, NO, 4gramCO, LSCR, JD	We adopted rule-based classifiers for the dev dataset. If we are able to judge a text pair by the rule-based classifiers, we exclude it from machine learning.
WSD-JA-MC-01		SVM	MO, CO, SMR, 4gramCO, 4gramMO, NO, LCSR, SOPO, NAVO, FPmarS, CTed	
WSD-JA-MC-02		LR	MO, CO, SMR, 4gramCO, 4gramMO, NO, LCSR, SOPO, NAVO, FPmarS, CTed	
WSD-JA-MC-03		SVM	MO, CO, SMR, JD, 4gramCO, 4gramMO, ModO, NO, LCSR, SOPO, NAVO, FPmarS, FPmulS, CTed	
WSD-JA-ExamBC-01	Limitation expressions	SVM	NO, named-entity mismatch, number expressions mismatch	Using in-house corpus
WSD-JA-ExamBC-02	Limitation expressions	SVM	NO, named-entity mismatch, number expressions mismatch	Using Wikipedia article titles.
WSD-JA-ExamBC-03	Limitation expressions	SVM, LR	NO, named-entity mismatch, number expressions mismatch	We assessed the same features and learning by SVM and LR. We chose the output with higher degree of conviction.

Table 16 Results for the BC subtask (%)

Run	MacroF1	Precision	Y-F1	Y-Prec	Y-Rec	N-F1	N-Prec	N-Rec
WSD-JA-BC-01	78.61	79.51	74.23	78.60	70.31	82.99	80.05	86.16
WSA-JA-BC-02	78.77	79.67	74.38	78.95	70.31	83.15	80.10	86.44
WSD-JA-BC-03	80.08	80.66	76.68	77.60	75.78	83.47	82.78	84.18

Table 17 Results of MC subtask (%)

Run	MacroF1	Precision	B-F1	B-Prec	B-Rec	F-F1	F-Prec	F-Rec	C-F1	C-Prec	C-Rec	I-F1	I-Prec	I-Rec
MC-01	53.98	68.80	68.29	59.57	80.00	74.70	72.48	77.07	0.00	0.00	0.00	72.93	69.36	76.89
MC-02	54.18	68.98	68.83	63.10	75.71	73.95	70.67	77.56	0.00	0.00	0.00	73.94	70.04	78.30
MC-03	54.39	69.53	68.29	59.57	80.00	75.29	72.73	78.05	0.00	0.00	0.00	73.99	70.51	77.83

Table 18 Results of ExamBC subtask (%)

Run	MacroF1	Precision	CAS	Y-F1	Y-Prec	Y-Rec	N-F1	N-Prec	N-Rec
WSD-JA-ExamBC-01	64.90	67.86	52.78	54.72	60.00	50.29	75.09	71.62	78.91
WSA-JA-ExamBC-02	63.96	67.63	51.85	52.46	60.61	46.24	75.47	70.57	81.09
WSD-JA-ExamBC-03	64.71	67.63	52.78	54.55	59.59	50.29	74.87	71.52	78.55

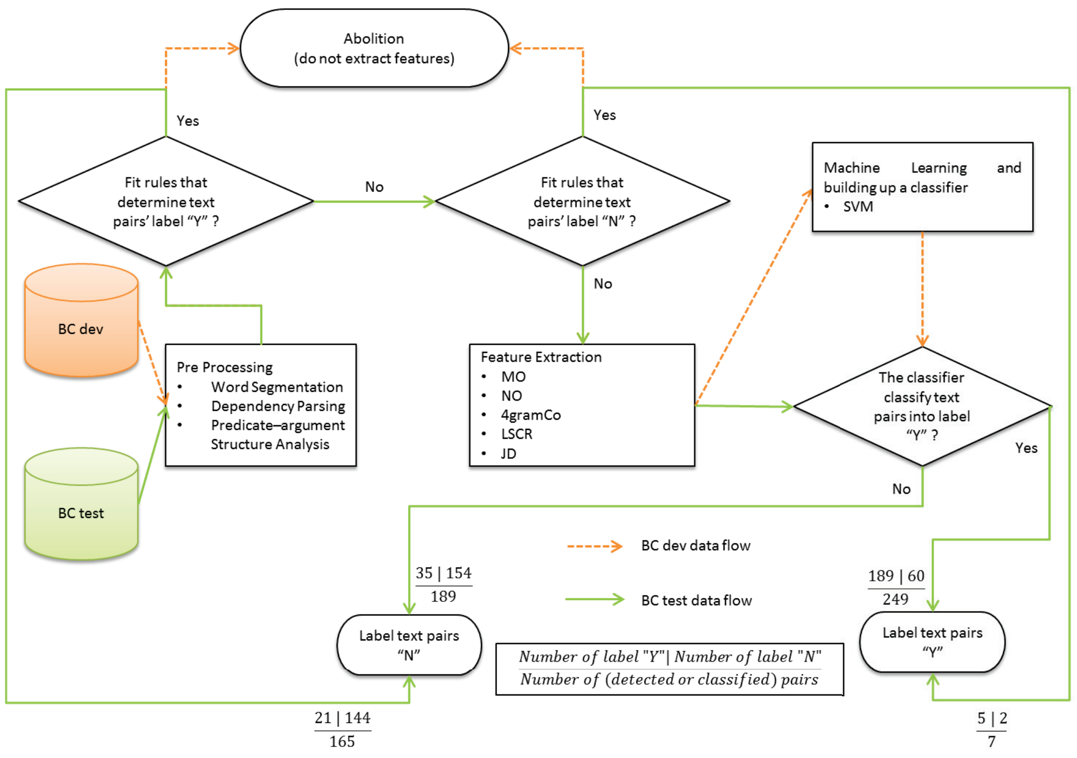

Abolition
(do not extract features)
Yes
Yes
Fit rules that determine text pairs' label "Y" ?
No
Fit rules that determine text pairs' label "N" ?
Machine Learning and building up a classifier
• SVM
BC dev
No
Pre Processing
• Word Segmentation
• Dependency Parsing
• Predicate–argument Structure Analysis
Feature Extraction
• MO
• NO
• 4gramCo
• LSCR
• JD
The classifier classify text pairs into label "Y" ?
Yes
BC test
No
BC dev data flow
BC test data flow
35 | 154 / 189
189 | 60 / 249
Label text pairs "N"
Number of label "Y"| Number of label "N" / Number of (detected or classified) pairs
Label text pairs "Y"
21 | 144 / 165
5 | 2 / 7

Figure 6 System architecture of WSD-JA-BC-03

Table 19 Confusion matrix for WSD-JA-ExamBC-01

		Gold		Total
		Y	N	
System	Y	87	58	145
	N	86	217	303
Total		173	275	448

Table 20 Confusion matrix for WSD-JA-ExamBC-02

		Gold		Total
		Y	N	
System	Y	80	52	132
	N	93	223	316
Total		173	275	448

Table 21 Confusion matrix for WSD-JA-ExmaBC-03

		Gold		Total
		Y	N	
System	Y	87	59	146
	N	86	216	302
Total		173	275	448

5. Discussion

This section further considers the results for each subtask.

5.1 Discussion of BC and ExamBC

Though most teams at NTCIR-9 RITE used many features for the BC subtask, we found that this was not necessarily effective, for either the BC or ExamBC subtask, we used only five features but as seen in Table 5, the rule-based classifier has high precision.

Comparing results from each BC run, we see that WSD-JA-BC-03 has the highest performance. Unlike other runs, WSD-JA-BC-03 successfully excluded text pairs classifiable by the rule-based classifier from the machine learning.

5.1.1 Error Analysis

A few examples were that are misclassified by the rule-based classifier.

Figure 7 shows misclassification using "limitation expressions." The text pair in Figure 7 is included in the RITE2 BC dev dataset. Though t2 contains limitation expression *dake* (だけ) and t1 does not, this text pair is labeled "Y", because the expression *hokubi* 'neck of spike' (穂首) is analyzed to mean *hokubi no bubun dake* 'ears of rice only' (稲穂の部分だけ). To classify this pair correctly, deeper semantic analysis is needed.

t1: 収穫具として、石包丁・石鎌があげられ、ともにイネを刈り取る際に用いられる道具であり、石包丁は穂首狩りに、石鎌は根刈りに主に用いられたとされる。
(As crop tools, there were stone knives and stone sickles, both of them used for reaping rice-plants, the stone knives were also used for reaping neck of spike, and stone sickles were used for trimming roots.)
t2: 石包丁は、収穫時にイネから稲穂の部分だけをむしるように切り取るために使ったものと考えられている。
(It is thought that the stone knives were used for trimming the ears of rice only.)

Figure 7 Misclassification by limitation expressions

Figure 8 shows an example of misclassification by named-entity. It is clear that "DEATH NOTE" is equal to "Death Note" (デスノート). But word is written in *katakana* orthography in t2 and in Roman letters in t1. As the written signs are different our system failed to recognize that these words were the same.

t1: DEATH NOTE で成功した大場つぐみ・小畑健は、2008 年より「バクマン。」の連載を開始した。
(Oba Tsugumi and Obata Ken who found success with DEATH NOTE have been serializing "Bakuman." since 2008.)
t2: 「デスノート」と「バクマン。」の作者は２人組である。
(The author of "Death Note" and "Bakuman." is a duo.)

Figure 8 Misclassification by named-entity

Figure 9 shows misclassification using number expressions. Using normalizeNumexp [5], *ichinin* 'one [person]' (一人) is incorrectly identified as a number expressions. To improve classification using number expressions, we need to avoid misidentifying them.

t1: 1962 年、DNA 二重らせん構造に関する研究により、ワトソンとクリックはウィルキンスとともにノーベル生理学･医学賞を受賞した。
(In 1962, Watson and Crick won the Nobel Prize in Physiology or Medicine with Wilkins for study of DNA's double helix structure.)
t2: ワトソンは､DNA の分子構造における共同発見者の一人として知られる。
(Watson is known as one of the co-discoverers of DNA structure.)

Figure 9 Misclassification by number expressions

5.1.2 Deterministic Use of Hand-Coded Rules

As shown in Table 9, the precision of our system using hand-coded rules deterministically differed little from that of a system them as features of machine learning. Thus, the precision of our hand-coded rules is as yet insufficient, and we must choose probable rules deterministically.

5.2 Discussion about MC subtask

This subsection considers the results for MC. Table 13 shows that precision improves when using features normalized by both t1 and t2 elements as opposed to by one text only, and also that precision is increased by using TED as a feature for machine learning.

We found that the system hardly classified texts as showing "contradiction," as seen in Table 10, Table 11, and Table 12. Generally, the number of characters in t1 is larger than that in t2 when t1 entails t2, because t1 has to contain all information in t2 in most such cases. Thus, the number of elements such as characters or morphemes in t1 tends to be larger than that in t2 when t1 entails t2. This results in differences between features normalized by t1 elements and by t2 elements when t1 entails t2 and when t2 entails t1. Those differences become effective features for the classifier. In contrast, the relationship "independence" implies that the texts are dissimilar; thus, features become less valuable for the classifier. Note that texts with a "contradiction" relationship have no tendency to differ in length. SVM decided that texts have other relationship except "contradiction" when the lengths of two texts are different. For this reason, fewer texts are classified as "contradiction."

6. Conclusion and Future Work

This paper describes our efforts at NTCIR-10 RITE-2 (team id: "WSD"). In BC and ExamBC subtasks, we found that hybrid systems adopting both rule and machine learning-based classifiers are better than the systems using either of these classifiers alone. MC, both TED and bidirectional similarity are effective.

Although we have put forward several rules, our approach still has limitations. It is hard to develop rules that detect true-entailment in text pairs. Our future work will be to develop such rules.

7. REFERENCES

[1] Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., Lin, C.-J., Shi, C., Mitamura, T., Kando, N., Shima, H., Takeda, K. 2013. "Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Conference", *In Proceedings of NTCIR-10 Workshop Meeting*.

[2] Shibata, T., Kurohashi, S. 2011. "Predicate-argument structure based Textual Entailment Recognition System of KYOTO Team for NTCIR9 RITE", *In Proceedings of NTCIR-9 Workshop Meeting, pp.310-317*.

[3] Odani, M., Shibata, T., Kurohashi, S. 2009. "Normalization of paraphrase representation for predicate-argument structure and use for textual entailment recognition", In *Proceedings of 15th Ann. Meeting of the Association for Natural Language Processing, pp.260-263*.

[4] Matsuyoshi et al. "Tsutsuji", http://kotoba.nuee.nagoya-u.ac.jp/tsutsuji/. (Accessed on February 19, 2013.)

[5] "normalizeNumexp", http://www.cl.ecei.tohoku.ac.jp/~katsuma/software/normalizeNumexp/. (Accessed February 19, 2013.)

[6] "Japanese WordNet", http://nlpwww.nict.go.jp/wn-ja/. (Accessed February 19, 2013.)

[7] "LIBSVM – A Library for Support Vector Machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm/. (Accessed on February 19, 2013.)

[8] "LIBLINEAR – A Library for Large Linear Classification", http://www.csie.ntu.edu.tw/~cjlin/liblinear/. (Accessed on February 19, 2013.)

[9] "MeCab: Yet Another Part-of-speech and Morphological Analyzer", http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html. (Accessed on February 19, 2013.)

[10] "CaboCha – Yet Another Japanese Dependency Structure Analyzer – Google Project Hosting", http://code.google.com/p/cabocha/. (Accessed on February 19, 2013.)

[11] Takamura et al. "Semantic Orientations of Words", http://www.lr.pi.titech.ac.jp/~takamura/pndic_ja.html. (Accessed on February 18, 2013.)

[12] Zhang, K., and Shasha, D. 1989. "Simple fast algorithms for the editing distance between trees and related problems", *SIAM J. on Computing, pp.1245-1262*.

[13] Tsuboi, Y., Kanayama, H., Ohno, M., Unno, Y. 2011. "Syntactic Difference Based Approach for NTCIR-9 RITE Task", In *Proceedings of NTCIR-9 Workshop Meeting, pp.410-411*.

[14] "Weblio thesaurus", http://thesaurus.weblio.jp/. (Accessed on February 19, 2013.)