Syntactic Difference Based Approach for NTCIR-9 RITE Task

Yuta Tsuboi Hiroshi Kanayama Masaki Ohno
IBM Research - Tokyo, IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato-shi
Kanagawa, Japan
{yutat,hkana,moono}@jp.ibm.com

Yuya Unno
Preferred Infrastructure
2-40-1-4F Hongo, Bunkyo-ku
Tokyo, Japan
unno@preferred.jp

ABSTRACT

This paper describes the IBM team's approach for the textual entailment recognition task (RITE) in NTCIR-9 [10] with experimental results for four Japanese subtasks: BC, MC, EXAM, and RITE4QA. To tackle the data set with complicated syntactic and semantic phenomena, the authors used a classification method to predict entailment relations between two different texts. These features were used for classification: (1) Tree edit distance and operations, (2) Word overlap ratios and word pairs, (3) Sentiment polarity matching, (4) Character overlap ratios, (5) Head comparisons, (6) Predicate-argument structure matching, and (7) Temporal expression matching. Feature (1) reflects the operations in the edit distance computation between the text and the hypothesis, which can capture the syntactic differences between two sentences. In the RITE task, Feature (1) is effective for the MC subtask and Feature (7) is effective for the EXAM subtask.

Keywords

NTCIR, RITE, entailment, machine learning, syntax, tree edit distance

Team Name: IBM

Subtasks: Japanese BC, MC, EXAM, and RITE4QA

External Resources: *Bunrui-goi-hyo* and Shibaki's ontology

1. INTRODUCTION

Textual entailment is a very complex natural language phenomenon. Although supervised classifiers have been used for this task, textual entailment recognition requires detecting complex syntactic and semantic relations between two different texts. To exploit these relations, we train classifiers in the pair feature space [11], in which a text pair are represented. In Section 2, we will explain our supervised machine learning approach. Since the number of the example is usually limited in the task, we propose the conversion of the labeled data between different entailment tasks.

In Section 3, we propose to use the operations in the tree edit distance computation between two dependency parse trees as the pair features to exploit syntactic differences of a sentence pair. We also employed several kinds of the pair features from a lexical similarity to the matching of temporal expressions. We will explain these pair features in Section 4.

We experiment with our approach on the NTCIR-9 RITE task in Section 5, and we will discuss the experimental results in Section 6. We show that the use of operations based on edit distance was effective for the MC subtask and that the temporal expression matching worked well for the EXAM subtask.

In Section 7, we will conclude our work and discuss future directions.

2. SUPERVISED MACHINE LEARNING

Since the BC, EXAM, and RITE4QA subtasks are binary classification problems (true or false entailment) and the MC subtask is a multi-class classification problem, a supervised machine learning approach is employed in these tasks. The pair of two different sentences (denoted as S and T [1]) are represented in a *pair feature space* [11] and a *logistic regression* (LR) model is trained using labeled examples.

Let $\boldsymbol{x} \in \boldsymbol{X}$ be the feature representation of a pair of S and T, $y \in Y$ be an entailment label of a label set Y, and $\boldsymbol{\phi}(\boldsymbol{x}, y) : |\boldsymbol{X}| \times |Y|$ be the cartesian product of $\boldsymbol{x}$ and a label assignment vector. LR model represents a conditional probability $\mathrm{P}(y|\boldsymbol{x})$ in a *log*-linear form:

$$\mathrm{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \frac{1}{Z} \exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}, y)), \quad (1)$$

where $\boldsymbol{\theta}$ is the parameter vector of LR model and $\boldsymbol{u}^\top \boldsymbol{v}$ denotes the inner product of the vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$. Note that the denominator is the partition function:

$$Z = \sum_{y \in Y} \exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}, y)).$$

Using a training data $E \equiv \{(\boldsymbol{x}, y)\}$, the parameter $\boldsymbol{\theta}$ can be estimated by the maximization of regularized *log*-

[1] S and T correspond to T1 and T2 in the RITE data set, respectively.

Table 1: Data conversion. The left columns denote the original relations between S and T, and the right columns denote the converted relations. The relation symbols Y and N denote true and false entailment, F, R, and B denote forward, reverse, bidirectional entailment, C and I denote contradiction and independence, respectively.

MC relation	BC relation
$S \xrightarrow{\mathrm{F}} T$	$S \xrightarrow{\mathrm{Y}} T$, $T \xrightarrow{\mathrm{N}} S$
$S \xrightarrow{\mathrm{R}} T$	$S \xrightarrow{\mathrm{N}} T$, $T \xrightarrow{\mathrm{Y}} S$
$S \xrightarrow{\mathrm{B}} T$	$S \xrightarrow{\mathrm{Y}} T$, $T \xrightarrow{\mathrm{Y}} S$
$S \xrightarrow{\mathrm{C}} T$	$S \xrightarrow{\mathrm{N}} T$, $T \xrightarrow{\mathrm{N}} S$
$S \xrightarrow{\mathrm{I}} T$	$S \xrightarrow{\mathrm{N}} T$, $T \xrightarrow{\mathrm{N}} S$

(a) From MC data to BC' data

BC relation	MC relation
$S \xrightarrow{\mathrm{Y}} T$	$S \xrightarrow{\mathrm{F,B}} T$
$S \xrightarrow{\mathrm{N}} T$	$S \xrightarrow{\mathrm{R,C,I}} T$

(b) From BC data to MC' data

likelihood:

$$\sum_{(\boldsymbol{x},y)\in E} \ln \mathrm{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) + \frac{||\boldsymbol{\theta}||^2}{2\sigma^2}, \tag{2}$$

where the final term is a *Gaussian prior* on $\boldsymbol{\theta}$ with mean 0 and variance σ^2. In the experiments, $\boldsymbol{\theta}$ was optimized by *Newton-CG* methods [5] and the hyper-parameter σ was determined by grid search based on 5-fold cross-validation (CV).

Since the number of the training data is limited in the RITE task, we convert the training data for the MC subtask as the additional training data for BC subtask, and vice versa. For this conversion, we assumed the interchangeability of the labels between BC and MC subtasks. By the data conversion from the MC subtask to the BC subtask, two examples for the BC subtask are generated from one labeled examples on the MC subtask since MC labels convey richer entailment information than BC labels (see Table 1(a)). We denote the combined training data of BC data and the examples generated from MC data as *BC+MC'* data. By the data conversion from the BC subtask to the MC subtask, one example with ambiguous labels for the MC subtask are generated from one labeled examples on the BC subtask (Table 1(b)). We denote the combined training data of MC data and the examples generated from BC data as *MC+BC'* data. To deal with label ambiguities in BC' data, we extend the LR objective function (2). Let $L \subseteq Y$ be a label candidates. Using the training data $E' \equiv \{(\boldsymbol{x}, L)\}$ with ambiguous labels, we maximize the regularized *marginal log*-likelihood:

$$\sum_{(\boldsymbol{x},L)\in E'} \ln \sum_{y\in L} \mathrm{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) + \frac{||\boldsymbol{\theta}||^2}{2\sigma^2}.$$

Although this is non-convex optimization problem, we can still use *Newton-CG* method to find a local-optimum. Note that, for MC+BC' data, we use a subset of MC data as the validation set of CV.

In the sequel, we describe the feature representation of a pair of S and T. We first explain notation here. Let $f(S,T)$ be a map function from a pair of S and T to a feature space $\boldsymbol{x} \in \boldsymbol{X}$. Let $\boldsymbol{s} = (s_1, \cdots, s_{|\boldsymbol{s}|})$ be a sequence of *bunsetsu* phrases of length $|\boldsymbol{s}|$ in S and $\boldsymbol{t}$ be that of T, where $\boldsymbol{s}$ and $\boldsymbol{t}$ consist of morpheme sequences $s = (m_1^s, \cdots, m_{|s|}^s)$ and $t = (m_1^t, \cdots, m_{|t|}^t)$, respectively.

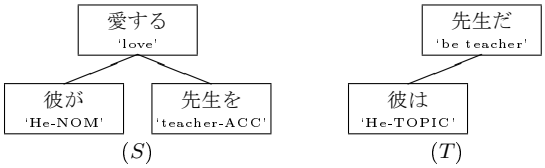

Figure 1: Edit distance minimization on trees

3. SYNTACTIC DIFFERENCE

3.1 Edit distance

In this work, syntactic differences between two sentences, S and T, are represented by edit operations used in tree edit distance calculation on the dependency trees.

The syntactic tree is constructed by the Japanese syntactic parser [4] which uses both the hand-crafted grammar and the statistical model created by a tagged corpus. The parser produces dependency structures between *bunsetsu*s as nodes so that S and T are represented as trees consisting of $s \in \boldsymbol{s}$ and $t \in \boldsymbol{t}$, respectively.

The tree edit distance between two trees S and T is calculated based on edit operations, each of which is one of insertion, deletion, or substitution, in the similar way as edit operations for strings. We define a cost function for an edit operation, $\gamma(s,t)$ on substituting $\boldsymbol{s}$ to $\boldsymbol{t}$. Costs for deletion and insertion operations can be denoted as special cases; $\gamma(s,\epsilon)$ for the cost of deletion of node s and $\gamma(\epsilon,t)$ for the cost of insertion of a node t. An *edit distance mapping* $M \subseteq \boldsymbol{s} \times \boldsymbol{t}$ represents a set of edit operations. Let $D \subseteq \boldsymbol{s}$ and $I \subseteq \boldsymbol{t}$ be the sets of nodes not appearing in M due to the existence in only one tree. Here the edit distance $\delta(\boldsymbol{s},\boldsymbol{t})$ is defined as the minimum cost mapping [1]:

$$\delta(\boldsymbol{s},\boldsymbol{t}) = \min_{M} \sum_{(s,t)\in M} \gamma(s,t) + \sum_{s\in D} \gamma(s,\epsilon) + \sum_{t\in I} \gamma(\epsilon,t).$$

Zhang and Shasha's algorithm [12] can calculate the edit distance $\delta(\boldsymbol{s},\boldsymbol{t})$ between $\boldsymbol{s}$ and $\boldsymbol{t}$ in $O(|\boldsymbol{s}|^2|\boldsymbol{t}|^2)$ time and space.

The edit distance computation between two syntactic trees corresponds to the word alignment between two sentences because the substitution from $\boldsymbol{s}$ to $\boldsymbol{t}$ can be regarded as the alignment of two nodes between $\boldsymbol{s}$ and $\boldsymbol{t}$. Such operations on syntactic trees are advantageous compared to the word-to-word alignment between sentences as sequences of *bunsetsu*s. For example, suppose the sentences S and T are "彼が 先生を 愛する" ('He loves a teacher') and "彼は 先生だ" ('He is a teacher'), respectively. When the edit distance is calculated on the sequence of *bunsetsu*s, the edit operations with the lowest cost should be (1) substitution of "彼は" to "彼が", (2) substitution of "先生を" ('teacher-ACC') to "先生だ" ('be a teacher') and (3) deletion of "愛する" ('love') with a rule that the replacement of functional words can be done with a small cost, and then each pair of two common content words "彼" ('he') and "先生" ('teacher') is connected in the two sentences. However, the semantic roles of "先生" ('teacher') are different in these two sentences, so this alignment is not desirable to capture the difference between two sentences.

In contrast, when the edit distance is calculated on the dependency tree shown in Figure 1, the optimal operations are (1) substitution of "彼が" ('he-NOM') to "彼は" ('he-TOPIC'), (2) deletion of "先生を", (3) deletion of "愛する" and (4) insertion of "先生だ", since the word "先生" appears in different positions in S and T, and thus the tree T can not be generated with a substitution operation of the nodes with "先生." This calculation successfully avoids the connection between two nodes with the word "先生" which behaves differently in two sentences.

3.2 Cost functions

Here we consider four types of cost functions: WO, BGH, HDM and Ontology. These functions calculate the costs of substitution operations differently, but the costs for deletion and insertion are set to one, *i.e.* $\gamma(s, \epsilon) = 1$ and $\gamma(\epsilon, t) = 1$, in the all cases. The feature representation based on the optimal edit operations will be shown in Section 4.

Jaccard distance metrics using word overlap (WO).

One of the cost functions is defined as *Jaccard distance* between the morpheme sets of two *bunsetsu*s $\boldsymbol{s}$ and $\boldsymbol{t}$:

$$\gamma(s,t) = 1 - \frac{|s \cap t|}{|s \cup t|}.$$

Semantic distance metrics using Bunrui-goi-hyou (BGH).

Another method to calculate the cost is based on semantic similarity between a pair of *bunsetsu*s. Here we use the the Japanese thesaurus, extended version of *Bunrui-goi-hyo* [8], which assigns semantic codes around 230,000 words.

We focus on the head content words, when we calculate semantic similarity between a pair of *bunsetsu*s. A head content word is defined as the rightmost content word in a *bunsetsu*, and a content word is either a noun, a verb, an adjective, an adverb, an interjection or their variants.

Given two *bunsetsu*s $\boldsymbol{s}$ and $\boldsymbol{t}$, we compare each code of head content words of them. Let $c_{\boldsymbol{s}}$ and $c_{\boldsymbol{t}}$ be the *Bunrui-goi-hyo* codes and $common(c_1, c_2)$ is a common depth in the thesaurus tree. *e.g.* $common(c_1, c_2) = 3$ when $c_{\boldsymbol{s}} =$ '3630′ and $c_{\boldsymbol{t}} =$ '36352′.

We define a cost function as follows, giving a smaller cost for more similar *bunsetsu*s:

$$\gamma(s,t) = \frac{1}{1 + common(\boldsymbol{c}_s, \boldsymbol{c}_t)}$$

Heuristic distance metrics (HDM).

Another method to measure the tree edit distance is the combination of BGH metrics and other conditions, by comparing head content words and parts-of-speech. We call this method HDM (Heuristic Distance Metrics), which is calculated as follows:

$$\gamma(s,t) = \begin{cases} 0 & \textit{same word} \\ 0.7 - 0.05 \cdot common(\boldsymbol{c}_s, \boldsymbol{c}_t) & \textit{same POS} \\ 1.0 & \textit{otherwise} \end{cases}$$

where 'same word' means that the canonical forms of the head content words of s and t are the same, 'same POS' means that the parts-of-speech of the head content words are the same, and $common()$ is in the BGH part.

Semantic distance metrics using an ontology (Ontology).

We also attempt to measure a semantic similarity using another resource and combine it with **HDM** to define another cost function.

To measure a semantic similarity, we use an ontology automatically generated from Wikipedia with Shibaki's Method[9]. It defines *is-a relations* among about 730,000 words. It has up-to-date knowledge since it was constructed from Wikipedia which is actively updated.

Given a pair of *bunsetsu*s $\boldsymbol{s}$ and $\boldsymbol{t}$, we focus on each head content word, and calculate their semantic similarity with the shortest path length in the ontology. Let $spl(s,t)$ be a function which returns the shortest path length in the ontology between the head content words of two input *bunsetsu*s, then the following condition is newly inserted after the first condition in the cost function of the HDM method defined above. Note that it is valid only when the head content words of the both *bunsetsu*s are found in the ontology.

$$\gamma(s,t) = \begin{cases} 0.7 - 0.5 \cdot \frac{1}{spl(s,t)} & \textit{a path found} \end{cases}$$

4. PAIR FEATURES

In this section, we describe the representation of the edit operations and other features used in the RITE task. The names in parenthesis denote the IDs of feature sets which are referred in the experimental results.

Edit distance and operations (EDO).

The edit distance and operations in Section 3 are represented as elements of a feature vector. We use the normalized value of the edit distance $\delta(\boldsymbol{s}, \boldsymbol{t}) / \max(|\boldsymbol{s}|, |\boldsymbol{t}|)$ ranging from 0 to 1. Although the value of the tree edit distance is used as a feature in the previous work [6], we also use edit operations as the pair features. For each *bunsetsu* in both D and I, we use the base form and part-of-speech (POS) tag of each morpheme and these of the last morpheme as a representation of the *bunsetsu*. For each pair of *bunsetsu*s in M, we use 1) the sequence pair of the base forms, 2) the sequence pair of POS tags, 3) the pair of the base forms of the last morpheme, and 4) the pair of the POS tags of the last morphemes. Note that the granularity of the POS tags described above is coase, such as noun or verb. We also used more fine POS tags, which are denoted as "POS fine" in the experimental results.

The value of the edit operation features is either the number of the edit operations, denoted as "Count", or binary value, denoted as "Binary". To make sure its value ranges from 0 to 1, "Count" is divided by $\max(|\boldsymbol{s}|, |\boldsymbol{t}|)$.

Since the edit operation features varies depending on the cost functions defined in Section 3, we also employed the combination of multiple feature sets based on different cost functions to represent a pair of $\boldsymbol{s}$ and $\boldsymbol{t}$.

Overlap ratios of words and word pairs (Word).

Let $\boldsymbol{m}_S$ and $\boldsymbol{m}_T$ be the set of content words in S and T, respectively. The value of word overlap ratio feature is defined as $|\boldsymbol{m}_S \cap \boldsymbol{m}_T| / |\boldsymbol{m}_T|$. The word pair feature is all the combination $(m_s, m_t) | m_s \in \boldsymbol{m}_S, m_t \in \boldsymbol{m}_T$ of the content words. We use only the word pair features appearing more than once in the training data.

Table 2: Example of 'Sentiment' feature set.

	sentence pairs	features
S	ＰＥＴ（ポジトロン断層撮影）検査は、肺がん、大腸がん、食道がんなど、ほとんどのがんの診療に有効とされている。 'The PET (positron emission tomography) is believed to be effective for the care of most types of cancers such as ...'	$f_{pol} = (+, +)$ $f_{same} = 1$
T	ＰＥＴはがんの診断に役立っている。 'PET helps the care of cancers.'	
S	山田洋次監督は男泣きの場面を作るのがうまい。 'Director Yoji Yamada is good as making scenes of men's weeping.'	$f_{pol} = (+, 0)$ $f_{diff} = 1$
T	山田洋次は映画監督です。 'Yoji Yamada is a film director.'	
S	失われた１０年は立ち遅れた反省や経験が生かされ、無駄でなかった。 'The Lost Decade was not wasted because ...'	$f_{pol} = (+, -)$ $f_{diff} = 1$ $f_{opp} = 1$
T	失われた１０年は無駄だった。 'We learned nothing from the Lost Decade.'	

Polarity matching with sentiment detection (Sentiment).

We also introduced a feature to confirm the semantic orientation of the two sentences. Intuitively if the S has a positive or negative polarity and S entails T, T is likely to have the same polarity. We used an existing sentiment detection engine [3], which detects positive or negative clauses in a sentence with high precision. The engine handles not only subjective utterance but also facts that suggests positive or negative attributes. In most of cases no polarity is detected, and thus this feature is expected to mainly work as a negative clue for the entailment when S and T have opposite polarities. This feature set consists of the following features:

f_{pol}: The combination of the polarities of S and T

f_{same}: Whether S and T has the same polarity (*e.g.* positive vs. positive)

f_{diff}: Whether S and T has the different polarities (*e.g.* positive vs. neutral)

f_{opp}: Whether S and T has the opposite polarities (*e.g.* positive vs. negative)

Table 2 shows examples of assignment of 'Sentiment' features for some sentence pairs.

Overlap ratios of characters (Char).

Let $\boldsymbol{c}_S$ and $\boldsymbol{c}_T$ be the set of characters in S and T, respectively. The value of character overlap ratio feature is defined as $|\boldsymbol{c}_S \cap \boldsymbol{c}_T|/|\boldsymbol{c}_T|$.

Head comparisons (Head).

The head *bunsetsu* in a sentence, the rightmost *bunsetsu* in the Japanese language, usually conveys the main concept of the sentence, and therefore we use a feature to examine whether the head *bunsetsu* of the two sentences are the same. When two *bunsetsu*s are compared, both words are converted to their canonical form and the difference in some symbols such as commas is ignored.

Fulfillment tests with predicate-argument structures (PAS).

Table 3: Example of sentence pairs that activate the feature f_{PAS}.

S	スーザン・トレスさんは極めて悪性度の高いがんの一種メラノーマが脳に広がり、脳死になった。 'Ms. Susan Torres became brain dead due to melanoma ...'
T	スーザン・トレスさんは脳死になった。 'Ms. Susan Torres became brain dead.'
S	日本で臓器移植法が施行されて７年以上になる。 'The organ transplantation law have been effective for 7 years in Japan.'
T	日本で臓器移植法は施行された。 'The organ transplantation law became effective in Japan.'

The syntactic tree is converted to a set of predicate-argument structures to examine whether T covers all information in S. A predicate-argument structure used here is either of these two types:

predicate type: a *bunsetsu* led by a verb or an adjective as a predicate, and zero or more postpositional phrases with a case marker as arguments

modification type: a modifier *bunsetsu* and a modifiee *bunsetsu*, such as adverbial modification

For example, the sentence (3) is converted to the following set of the predicate-argument structures. (P1) is a predicate type, and (P2) and (P3) are examples of the modification type.

彼は 大きな 駅へ ゆっくり 行った。 (3)
('He went to a big station slowly.')

(P1) 行く (彼, 駅) ('go (he, station)')

(P2) 大きな〈駅〉('big ⟨station⟩')

(P3) ゆっくり〈行く〉('slowly ⟨go⟩')

We use a feature f_{PAS} of the fulfillment test. The $f_{PAS} = 1$ only if the all of the predicate-argument structures in T are subsumed by one of those in S, and otherwise $f_{PAS} = 0$. "p_1 subsumes p_2" mean that p_1 and p_2 has the same predicate and all of the arguments of p_2 appear also in p_1. For instance, a predicate-argument structure "行く (駅)" is subsumed by (P1) in the above example. Table 3 exemplifies sentence pairs in which $f_{PAS} = 1$. This feature is expected to be a strong clue for the entailment.

To improve the coverage of this subsumption, the introduction of hyponym and hypernym relations using WordNet were attempted, however, few pairs of nouns in S and T matched the relations, so we gave up to use WordNet for this future.

Matching of temporal expressions (Temporal).

As proven by the successful of the GeoTime task in the previous NTCIR [2], temporal information has an important role in information retrieval, and actually many temporal expressions appear in the development set of the data, especially in the EXAM subtask, and thus we added a feature set to compare the temporal expressions in the sentence pairs, with two features:

f_{match}: Temporal expressions appear in both S and T and at least one pair has an overlap

$f_{unmatch}$: Two temporal features appear in both S and T and none of the pairs have an overlap

where the overlap is determined based on the ranges of the years, by using the following patterns:

Year: "N 年" ('the year of N') is converted to the year range $[N, N]$. The Japanese calendar scheme is also covered, for example, "昭和 50 年" is converted to $[1975, 1975]$.

Decade: "N 年代" ('the decade from N') is converted to the year range $[N, N+9]$. The suffixes "前半" ('the first half') and "後半" ('the latter half') are also considered, for example, "1920 年代前半" is converted to the year range $[1920, 1924]$.

Century: "N 世紀" ('Nth century') is converted to the year range $[100(N-1)+1, 100N]$. The suffixes "前半", "後半" and some other variations such as "初頭" ('beginning') reduce the width of the year range.

For example, when two sentences S include a temporal expression "1620 年代" ('1620s') and T has "17 世紀前半" ('the first half of the 17th century'), their ranges are $[1620, 1629]$ and $[1601, 1650]$, respectively, and thus the feature f_{match} is set to true since the year ranges overlap.

5. EXPERIMENTAL RESULTS

We employ the sevral combinations of the cost functions for edit distance described in Section 3 and the pair feature sets described in Section 4. We basically select feature sets for the formal-run submissions by the average accuracies on 5-fold cross-validation (CV) using the training data.

Table 4 shows the average accuracies on CV and the formal-run results, and Table 5 and 6 show the confusion matrices of the systems using a feature set which show the best CV accuracies is used. Although we evaluate 280 feature sets for all the subtasks, because of the limitation of space, we only report the results of the submitted systems for the formal-run, all of the feature sets using the HDM cost function, and the best feature sets at any performance measure in Table 4, and report the confusion matrices of the feature set which mark the best CV accuracy in each subtask. In Table 4, the submitted system outputs are denoted by the super script of accuracy values. Since we found and fixed bugs after the formal-run for BC and MC subtasks, we report the revised scores obtained after the bug-fix. The top column names denote test data, the second column names denote training data for LR models, and the third column names denote performance measures where CV stands for the average accuracy (%) on 5-fold cross-validation, AC stands for accuracy (%) on formal-run test data.

5.1 BC subtask

In terms of the formal-run accuracy, LR models trained on the BC data using edit distance and operations (EDO) based on BGH cost function and most of the other pair features achieve relatively high accuracy and the best system achieve 56.0% using binary-valued edit operation features. The LR models trained on the BC data perform better than the LR models trained on the BC+MC' data. Comparing with the best system on the formal-run data, the best feature sets of CV on the training data show slightly worse performance (52.4% for the BC data and 51.6% for the BC+MC' data).

Table 5: The confusion matrices in the BC, EXAM, RITE4QA tasks.

(a) BC

System \ Correct	Y	N
Y	137	125
N	113	125

(b) EXAM

System \ Correct	Y	N
Y	103	45
N	78	216

(c) RITE4QA

System \ Correct	Y	N
Y	65	521
N	41	337

Table 6: The confusion matrix in the MC task.

System \ Correct	F	R	B	C	I
F	87	6	21	30	40
R	7	89	20	9	16
B	7	12	30	16	6
C	4	0	1	5	4
I	5	3	3	5	14

5.2 MC subtask

The LR models using EDO features show much better performance than the LR model using all the features except EDO. Based on LR models trained on the the MC data, the formal-run accuracy averaged over all of the feature sets includes EDO is 47.1% comparing with 35.9% of the system without EDO. The best LR model achieve 51.6% accuracy on the formal-run test data. Again, the LR models trained using the MC data perform better than the LR models trained using the MC+BC' data.

5.3 EXAM subtask

Most of the LR models using the matching of temporal expressions (Temporal) perform better than that not using Temporal. The best LR models achieve 72.6% accuracy on the formal-run test data.

5.4 RITE4QA subtask

The LR model trained on the BC+MC' data using all the pair features except EDO shows the best performance, Accuracy=72.6% , TOP1=18.1%, and MMR5=29.0% . We observed a inverse correlation or week correlation between the CV accuracy on the BC or BC+MC' training data and the accuracy or QA performance measures on the RITE4QA formal-run data.

6. DISCUSSION

The edit distance and operations (EDO) are significantly effective features in the MC subtask. To put it more precisely, the classification of forward (F), reverse (R), and bidirectional (B) entailment using those features is more accurate than that without the edit distance and operations. Table 7 shows confusion matrices in the MC subtask when either of the edit distance and operations are used. As an example, Table 7(a) shows the result of the edit distance and operations derived by the HDM function. Comparing with Table 7(b), the number of the correct predications, the diagonal numbers of the matrices, of F, R, and B is larger in Table 7(a).

Table 4: Accuracies and QA performance scores on subtasks. CV stands for the average accuracy on 5-fold cross-validation using training data, and AC stands for the accuracy on formal-run data. The super script of accuracy values indicates the result of submitted system output:
a) IBM-JA-BC-01 (51.6), b) IBM-JA-BC-02 (52.6), c) IBM-JA-BC-03 (50.0),
d) IBM-JA-MC-01 (45.5), e) IBM-JA-MC-02 (51.1), f) IBM-JA-MC-03 (44.8),
g) IBM-JA-EXAM-01, h) IBM-JA-EXAM-02, i) IBM-JA-EXAM-03,
j) IBM-JA-RITE4QA-01, k) IBM-JA-RITE4QA-02, and l) IBM-JA-RITE4QA-03.
Values in parenthesis are the submitted formal-run results of BC/MC subtasks before the bug-fix. Note that the accuracy of IBM-JA-BC-02 with the bug is same as that without the bug.

	Test Data	BC				MC				EXAM		RITE4QA					
	Training Data	BC		BC+MC'		MC		MC+BC'		EXAM		BC			BC+MC'		
Cost Function	Performance Measure	CV	AC	CV	AC	CV	AC	CV	AC	CV	AC	AC	TOP1	MMR5	AC	TOP1	MMR5
None	Word + Sentiment + Char + Head + PAS + Temporal	52.8	52.0	57.5	48.2	33.6	35.9	35.5	31.4	67.9	69.5	34.5	5.5	19.8	**63.5**	**18.1**	**29.0**
HDM	EDO Count, POS fine	49.0	47.8	64.1	47.2	49.1	48.2	49.6	47.5	62.5	59.5	38.5	13.9	24.6	26.8	12.7	24.0
	+Word	50.2	51.8	62.4	48.8	49.1	51.1	50.2	49.5	62.1	67.4	29.5	10.0	22.0	34.4	12.8	24.6
	+Sentiment	52.4	50.8	61.7	47.4	48.9	49.5	51.5	48.0	62.1	67.0	32.0	9.5	21.5	32.1	11.8	24.1
	+Char	52.4	50.8	62.5	48.0	48.0	48.6	51.3	47.3	64.1	67.6	31.5	9.3	21.6	35.5	13.1	25.3
	+Head	53.0	50.4	62.2	49.4	49.1	49.1	49.5	45.9	63.7	67.6	31.0	9.8	22.3	34.3	13.8	25.7
	+PAS	54.2	52.6	62.2	49.2	49.1	48.9	49.2	46.1	63.7	67.6	30.8	8.8	21.8	34.0	13.3	25.5
	+Temporal	54.2	51.6	62.3	49.4	49.1	48.6	49.5	45.9	**69.1**	72.2[g]	30.9	10.3	22.8	32.9	14.3	26.0
	EDO Binary, POS fine	50.0	50.0	63.8	46.6	48.6	44.1	47.9	40.9	60.1	60.0	54.5	11.9	23.5	36.4	11.6	23.4
	+Word	50.6	49.6	64.5	47.4	48.6	46.1	48.8	44.1	61.1	61.3	46.9	14.7	24.7	40.5	12.9	24.2
	+Sentiment	50.8	50.2	64.6	46.0	48.9	45.0	50.9	45.2	61.7	62.0	44.8	14.7	24.9	38.1	12.2	24.2
	+Char	50.8	50.2	64.4	48.8	49.3	45.7	50.8	44.5	61.9	62.0	44.8	14.7	25.0	42.9	13.7	25.1
	+Head	51.4	49.0	64.1	47.6	49.8	44.8	51.3	45.7	61.5	62.2	51.2	11.5	23.5	42.3	13.7	25.2
	+PAS	50.8	49.6	64.0	47.4	49.8	44.8	51.3	44.5[f]	61.5	62.2	44.9	14.2	24.8	34.6	13.6	25.2
	+Temporal	51.0	49.6	64.3	50.0	49.8	44.8	51.1	44.5	62.3	62.7	44.7	14.2	24.9	42.5	13.7	25.0
	EDO Count	50.2	48.2	63.9	48.8	49.1	47.0	48.4	47.7	61.9	60.0	37.8	12.9	24.4	26.9	12.2	23.7
	+Word	50.8	52.0	60.8	48.6	49.1	50.0	48.8	50.0	61.3	67.2	28.7	10.3	21.9	33.1	10.8	23.4
	+Sentiment	52.6	51.8	60.9	47.6	47.5	48.9	50.2	46.4	61.3	67.2	30.1	9.0	21.3	40.9	12.0	24.2
	+Char	52.6	51.6	61.9	47.0	48.2	48.9	50.2	47.7	62.5	67.6	30.3	9.3	21.7	38.1	14.6	26.0
	+Head	53.8	51.6	61.8	47.4	48.0	48.6	48.8	46.4	62.5	67.4	30.2	9.6	22.2	37.7	14.6	26.0
	+PAS	54.8	52.2[a]	62.0	46.4	48.2	48.6	49.2	46.1	62.5	67.4	31.6	9.1	21.8	37.1	14.6	25.9
	+Temporal	54.6	51.8	62.0	46.8	48.2	48.6	49.2	46.4	68.3	**72.6**	28.2	9.1	21.9	35.3	14.6	26.1
	EDO Binary	51.2	49.8	65.0	47.6	45.2	45.2	47.7	43.2	60.5	61.8	55.2	10.1	22.5	39.3	10.6	23.2
	+Word	50.8	51.4	64.3	47.4	47.5	45.2	47.3	45.0	61.5	62.0	53.6	10.6	23.0	42.1	12.1	24.1
	+Sentiment	52.0	52.0	63.8	48.6	47.0	44.3	49.1	44.3	61.1	61.5	52.3	10.6	23.0	43.4	12.9	24.4
	+Char	52.4	52.4	64.0	49.0	45.9	45.5	49.3	45.7	61.5	62.7	53.0	11.5	23.5	44.7	14.2	25.6
	+Head	52.6	51.8	63.9	47.8	49.3	45.7	49.1	45.0	61.9	62.4	52.6	12.5	24.2	41.8	14.7	26.0
	+PAS	53.2	52.2	64.1	50.2	48.0	45.7	49.1	44.8	61.9	62.4	51.7	11.5	23.7	44.4	13.2	25.2
	+Temporal	53.0	52.6	64.0	50.2	49.3	45.2	49.1	44.8	63.5	65.8	51.8	11.5	23.7	45.4	14.7	26.0
WO	EDO Binary, POS fine + Word + Sentiment + Char + Head + PAS + Temporal	48.8	50.2	64.1	48.0	48.4	48.0	47.9	45.7	63.3	62.0	53.6	14.3	**25.6**	31.1	14.6	25.7
BGH	EDO Binary, POS fine + Word + Sentiment + Char + Head + PAS	49.4	55.0	63.7	**54.8**	46.8	45.9	48.3	43.9	60.9	59.0	45.5	12.0	24.2	38.6	10.8	24.5
	EDO Binary + Word + Sentiment + Char + Head + PAS	51.8	**56.0**	64.2	53.6	46.1	46.6	45.2	42.0	60.7	60.0	45.6	10.0	23.9	37.3	13.3	26.1
Ontology	EDO Count, POS fine + Word	51.6	51.8	62.2	49.2	49.3	50.7	49.9	**50.2**	62.3	67.4	29.6	11.0	22.2	33.5	12.8	24.7
	+Sentiment + Char + Head + PAS + Temporal	53.8	52.2	62.3	49.6	49.3	48.4	49.5	46.4	**69.1**	72.2	30.3	9.8	22.5	32.9	14.3	25.9
	EDO Binary, POS fine + Word	50.8	50.2	64.3	48.4	49.1	45.7	49.0	45.2	61.5	62.4	46.7	**15.2**	25.0	42.0	12.4	24.2
	EDO Count + Word + Sentiment + Char + Head + PAS	**55.0**	52.4	61.4	48.2	48.2	48.4	49.2	46.1	62.5	67.4[h]	31.6[k]	9.1[k]	21.7[k]	35.3	12.8	25.3
	+Temporal	54.8	52.4	61.7	47.2	48.2	48.4	49.2	46.4	68.3	**72.6**	30.3	9.1	22.1	36.4	14.6	26.1
HDM & WO	EDO Count, POS fine + Word + Sentiment + Char	52.8	50.6	62.8	48.4	50.0	46.8	**52.4**	47.0	64.1	67.6	30.1	9.8	22.0	32.0	11.8	24.8
	EDO Count + Word + Sentiment + Char + Head + PAS + Temporal	**55.0**	51.6	62.6	47.4	47.3	48.2	49.9	47.5	67.9	71.3	32.2	11.6	23.2	32.1	12.1	24.8
HDM & BGH	EDO Count + Word + Sentiment + Char + Head + PAS	54.0	52.6[b]	62.3	48.0	48.9	49.1	49.2	45.9	62.5	67.0	29.4	9.8	22.3	34.8	13.1	25.0
	EDO Binary	52.8	51.6	**65.5**	49.4	46.1	44.3	46.6	44.1	60.7	60.0	54.5	11.9	23.8	40.1[l]	8.7[l]	22.2[l]
	+Word + Sentiment + Char + Head	52.6	52.8	65.2	51.6	46.4	43.6	46.8	43.6[d]	61.3	60.9	52.6	11.5	24.2	41.9	11.1	24.0
HDM & BGH & WO	EDO Count, POS fine	47.8	48.2	64.1	47.0[c]	49.3	48.0	49.0	47.7	61.1	61.1	31.5	11.1	22.7	29.5	13.0	24.1
	+Word	51.0	52.6	63.6	47.4	50.2	50.2[e]	50.9	49.8	62.7	65.8	28.7	11.5	22.7	28.4	11.8	23.8
HDM & Ontology & WO	EDO Count, POS fine + Word	51.4	52.0	63.6	47.6	51.1	**51.6**	51.1	**50.2**	62.1	67.0	31.4	11.5	22.7	28.5	12.3	23.8
	+Sentiment + Char	52.0	51.2	63.7	48.4	**52.3**	47.3	52.4	47.3	64.7	66.7	30.7	10.3	22.5	30.6	11.3	24.4
	+Head + PAS + Temporal	51.4	51.0	63.4	49.0	50.0	48.2	51.5	47.5	67.1	71.5	33.3[j]	11.3[j]	23.3[j]	30.9	14.3	26.0
	EDO Binary	50.6	50.2	63.3	49.2	46.6	45.5	45.9	45.0	59.9	60.9	**55.9**	10.6	22.8	42.7	8.9	22.1
HDM & BGH & Ontology & WO	EDO Count, POS fine + Word + Sentiment + Char + Head + PAS + Temporal	51.8	51.6	63.8	48.6	51.4	48.6	50.8	47.3	66.7	**72.6**	39.8	10.5	23.5	28.6	12.6	24.8
	EDO Binary, POS fine + Word + Sentiment + Char + Head + PAS + Temporal	51.4	47.4	64.4	48.4	47.5	45.7	48.8	41.8	61.5	58.4[i]	49.3	14.0	25.1	43.2	10.8	23.3

Table 7: The confusion matrix in the MC task when either of the edit distance and operations are used.

System \ Correct	F	R	B	C	I
F	81	5	26	29	40
R	3	88	18	11	18
B	9	13	28	14	5
C	11	0	2	5	5
I	6	4	1	6	12

(a) With edit distance and operations

System \ Correct	F	R	B	C	I
F	59	28	47	23	25
R	17	60	6	12	26
B	16	9	16	13	7
C	12	3	5	10	9
I	6	10	1	7	13

(b) Without edit distance and operations

Table 8: The average difference number of *bunsetsu*s and the average edit distance of each cost function on the test data of each subtask.

	BC	MC	EXAM	RITE4QA
$\|\boldsymbol{s}\| - \|\boldsymbol{t}\|$	4.68	0.44	6.17	9.11
HDM	7.54	6.43	11.64	12.47
BGH	8.52	7.47	12.59	13.58
WO	8.32	7.26	12.82	13.49
Ontology	7.53	6.42	11.64	12.47

Table 9: Examples of sentence pairs containing phrases with similar meanings.

S	バジル、セージ、タイムなどフレッシュハーブの消費量はウナギのぼりである。 'The consumption of fresh herbs such as basil, a sage, and thyme is rapidly increasing.'
T	ハーブの消費が増えている。 'The herbs consumption is increasing.'
S	今回の事件では、男子学生が「ハロウィーンの夜にやってやる」と知人に漏らすなど、決行日を選んだ上で、事件を起こしていたことが明らかになっている。 'In this incident, it is clear that the boy students chose the day to act in advance and said to his acquaintance, "I am going to act at the Halloween night".'
T	男子学生は決行日を１０月３１日に選んだ。 'The boy student decided to act on October 31. '

Table 10: The confusion matrices in the EXAM subtask when either of Temporal feature value is true.

System \ Correct	Y	N
Y	21	4
N	3	4

(a) $f_{match} = 1$

System \ Correct	Y	N
Y	0	0
N	1	29

(b) $f_{unmatch} = 1$

However, as described in Section 5, the EDO features are not effective in the other subtasks. One reason of the effectiveness in the MC subtask could be the simplicity of the syntactic differences observed in the MC data. As an index of the simplicity of the syntactic differences, the difference in the number of *bunsetsu*s between S and T on test data, denoted as $|\boldsymbol{s}| - |\boldsymbol{t}|$ at the second row of Table 8. Note that these numbers are similar to those at training data. The value of $|\boldsymbol{s}| - |\boldsymbol{t}|$ of MC data is nearly zero (0.44) and much smaller than that of other data. Since our cost functions for insertion and deletion are simply set to one and the cost functions for substitution are less than or equal to one, our design of cost function prefers substitutions and tends to overlook possible important insertions and deletions. However, if the number of nodes and the tree structures of S and T are similar, the edit distance computation with few insertions and deletions could be a natural alignment to exploit syntactic relations of two sentences. On the other hand, the syntactic differences in the other data may be much complicated than that in the MC data. This complexity of syntactic differences is also a possible explanation why LR models using the converted data, BC' and MC', do not perform well on the different subtasks.

One remark is that, although we design four different cost functions, WO, BGH, HDM, and Ontology, the difference of the resulting edit distances are small as shown in Table 8. And we could not observe superiority of any cost function in the final entailment classifications.

We compared two resources used for measuring semantic similarity based cost function, the ontology constructed from Wikipedia and Bunrui-goi-hyo. Although the ontology has a larger vocabulary than Bunrui-goi-hyo, there was little difference in their contribution to the final results. One of the reasons is mismatches between the words in the resources and the result of the morphological analysis. For example, the ontology contained "グスタフ・シュトレーゼマン" ('Gustav Stresemann') but our system splited it into three morphologies, "グスタフ" ('Gustav') and "・", "シュトレーゼマン" ('Stresemann'). To increasing the coverage of the detection of similarities, it is important to use the same unit for resource construction and morphological analysis or to adjust the result of the morphological analysis to the resource unit.

Our system with the two resources did not recognize some phrase relations with similar meanings. Table 9 exemplifies two sentence pairs which have phrases with similar meanings. In the former pair, the key to analyze their relation is to recognize "ウナギのぼりである" ('be rapidly increasing') is similar in meaning to "増えている" ('be increasing'). The other pair suggests the need of the knowledge that Halloween is October 31. It is important to arrange some resources constructed on various perspectives for the recognition of phrase-level correspondences.

The feature set 'Temporal' worked very well in the EXAM subtask, due to the frequent appearance of the expressions of the year, decade or century in the EXAM data. Table 10(a) and Table 10(b) show the relations of the system outputs and the correct tags when the temporal expressions matched together ($f_{match} = 1$) and when the temporal expressions did not overlap ($f_{unmatch} = 1$), respectively. These results show that the mismatch of the temporal expressions will be a strong hint for the non-entailment relation. This is because often T in the dataset with the 'N' label includes statements with a wrong year, for example, "The Constitution of the U.S. was established in the early 19th century", and the corresponding S includes the correct year ("1788" in this

Table 11: Pearson correlation coefficient between the average accuracy on 5-fold CV using the training data and the performance measures using formal-run data based on 280 different feature sets.

Test Data	BC		MC		EXAM	RITE4QA					
Training Data	BC	BC+MC'	MC	MC+BC'	EXAM	BC			BC+MC'		
Perf. Measure	AC	AC	AC	AC	AC	AC	TOP1	MMR5	AC	TOP1	MMR5
Correlation	0.43	0.30	0.57	0.49	0.74	-0.34	-0.42	-0.44	0.23	-0.20	-0.31

case). Another remark is that even in the cases of $f_{match} = 1$, more than half of them are correctly predicted as 'N'. This suggests the advantage of the learning method that handles this kind of strong tendencies as a preference, instead of a constraint by a hand-tailored rule.

In our feature sets, there is weak relationship between the accuracies on training data and formal-run performance in BC & RITE4QA subtasks. Although the good feature set on CV also show the good formal-run performance on MC & EXAM subtasks, the best feature set on CV and formal-run are different on BC & RITE4QA subtasks. Table 11 shows a correlation between the average accuracy on 5-fold CV using the training data and the performance measures using formal-run data based on 280 different feature sets. For example, the last column shows that there is a negative correlation between the average accuracy using the BC+MC' training data and the MMR5 score on the RITE4QA test data. The correlation values suggest that our feature representation of training data in BC & RITE4QA subtasks is not good enough to induce general classification rules or , at least, the classification rules for the specific test data.

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed a classification approach for NTCIR-9 RITE. In the experiments, we presented that tree edit distance and operations were effective feature for the MC subtask and temporal expression matching was effective feature for the EXAM subtask.

One drawback of the tree edit distance approach is the difficulty of the design of the cost functions. Although we implemented four different cost functions, the differences of these edit distances were small and the edit distance based features were effective only on the MC subtask. The supervised learning of the edit cost function is one of the interesting research directions [7].

In the EXAM subtask, the Temporal feature set worked to increase the accuracy, and the combined use of temporal and geological expressions is a interesting research line. However, it is far from the complete understanding of the question and the world knowledge. To make the system more general and robust, we need to seek ways to handle the semantics of the whole sentences, and that of the pair of the whole sentences.

8. REFERENCES

[1] T. Akutsu. Tree edit distance problems: Algorithms and applications to bioinformatics. *IEICE Transactions on Information and Systems*, E93-D(2):208–218, 2010.

[2] F. Gey, R. Larson, N. Kando, J. Machado, and T. Sakai. NTCIR-GeoTime overview: Evaluating geographic and temporal search. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, 2010.

[3] H. Kanayama, T. Nasukawa, and H. Watanabe. Deeper sentiment analysis using machine translation technology. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 494–500, 2004.

[4] H. Kanayama, K. Torisawa, Y. Mitsuishi, and J. Tsujii. A hybrid Japanese parser with hand-crafted grammar and statistics. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 411–417, 2000.

[5] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008.

[6] Y. Mehdad and B. Magnini. Tree edit distance for recognizing textual entailment: Estimating the cost of insertion. In *PASCAL RTE-2 Challenge*, 2006.

[7] Y. Mehdad and B. Magnini. Optimizing textual entailment recognition using particle swarm optimization. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 36–43, 2009.

[8] National Language Research Institute. Bunrui-goi-hyo (revised and enlarged edition), 1996. (In Japanese).

[9] Y. Shibaki. Constructing large-scale general ontology from wikipedia. Master's thesis, Nagaoka University of Technology, Japan, 2011.

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

[11] F. M. Zanzotti, M. Pennacchiotti, and A. Romoschitti. A machine learning approach to textual entailment recognition. *Natural Language Engineering*, 15(4):551–582, 2009.

[12] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.