IBM Team at NTCIR-10 RITE2: Textual Entailment using Temporal Dimension Reduction

Masaki Ohno
IBM Research - Tokyo
moono@jp.ibm.com

Yuta Tsuboi
IBM Research - Tokyo
yutat@jp.ibm.com

Hiroshi Kanayama
IBM Research - Tokyo
hkana@jp.ibm.com

Katsumasa Yoshikawa
IBM Research - Tokyo
katsuy@jp.ibm.com

ABSTRACT

Our system for the Japanese BC/EXAM subtasks in NTCIR-10 RITE2 is an extension of our previous system for NTCIR-9 RITE. The new techniques are (1) Case-aware noun phrase matching using ontologies: The motivation of the feature is to capture finer syntactic structures than simple word matching. We uses ontologies to allow flexible matching of noun phrases. (2) Temporal expression matching after mapping historical entities to specific time intervals: The motivation of historical entity mapping is to expand the capabilities of the temporal expression matching. From the experimental results, we found that the coverage is more important than the accuracy in the temporal entity mapping. The scores of the formal runs were 74.9% (accuracy in BC) and 64.5% (accuracy in EXAM), which outperformed the baselines provided by the organizer.

Team Name

IBM

Subtasks

Japanese BC, EXAM

Keywords

NTCIR, RITE, textual entailment recognition, machine learning, ontology

1. INTRODUCTION

Textual entailment recognition is a task to determine whether the meaning of a hypothesis (H, the entailed text) can be inferred from a text (T, the entailing text). Understanding the language and background knowledge at a human level is required in textual entailment recognition. Many NLP applications, such as Question Answering and Information Retrieval, need textual entailment recognition technologies to improve their performances.

This paper describes new techniques for entailment recognition, case-aware noun phrase matching using ontologies and advanced temporal expression matching, which we integrated with our previous system [7] from NTCIR-9 RITE [5]. In the previous system we focused only on numerical temporal expressions, such as “1620 年代” (‘1620s’). The newly added features handle not only numerical temporal expressions but also non-numerical temporal expressions.

The architecture of our system for NTCIR-10 RITE2 [8] is almost the same as the previous system. To capture complex syntactic and semantic relationships between two different texts, a supervised machine learning approach is used. Given a pair of texts H and T, the system parses them and extracts features for the classifier. To represent a text pair, the features are located in the pair feature space [9]. Resources created from Wikipedia were used in some of the pair feature extractions. Finally, a classifier predicts a label representing whether the meaning of H can be inferred from T.

We used these feature sets for the classifier in this work. The names in parentheses denote the ID of each feature set that is referred in the rest of this paper.

1. Character overlap ratio (*Char*)
2. Word overlap ratio and word pairs (*Word*)
3. Predicate-argument structure matching (*PAS*)
4. Matching of noun phrases followed by a common case marker (*Case*)
5. An extension of *Case*, using the ontology generated from Wikipedia categories (*CaseOnt1*)
6. An extension of *Case*, using word classes extracted from Wikipedia abstracts (*CaseOnt2*)
7. Temporal expression matching (*Temporal*)
8. An extension of *Temporal*, using a dictionary created from Wikipedia chronological tables (*Time1*)
9. An extension of *Temporal*, using a dictionary created from Wikipedia infoboxes and abstracts (*Time2*)
10. The same as *Time2*, but without numerical temporal expressions (*TimeE*)

Section 2 describes the first three feature sets (1 to 3) derived from our previous system. Section 3 describes the next three features (4 to 6) generated by the matching of noun phrases followed by a case marker. Section 4 describes the last four features (7 to 10), which provides the temporal expression matching after mapping any historical entities to specific time intervals.

The best accuracy of the formal runs we submitted in the BC subtask was 74.9% and our result was 11.0 points better than the baseline provided by the organizer. The best accuracy of the formal runs we submitted in the EXAM subtask was 64.5% and our result was 8.0 points better. To accurately evaluate the proposed method described in this paper, we investigated all of the combinations of the feature sets. The best accuracies of the formal runs were 77.5% in the BC subtask and 69.0% in the EXAM subtask.

2. TEXTUAL ENTAILMENT SYSTEM AT NTCIR-9 RITE TASK

The system proposed here exploits four features and one classifier that were effective in the NTCIR-9 RITE task. We describe the classifier in Section 2.1 and the four feature sets in Section 2.2.

2.1 Classifier

To capture complex syntactic and semantic relationships between two different texts, we used a supervised machine learning approach. Each pair of different sentences (denoted as T and H) are represented in a *pair feature space* [9] 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 H 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. the LR model represents a conditional probability $\mathrm{P}(y|\boldsymbol{x})$ in a *log*-linear form:

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

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 the training data $E \equiv \{(\boldsymbol{x}, y)\}$, the parameter $\boldsymbol{\theta}$ can be estimated by maximizing the regularized *log*-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 [3] and the hyper-parameter σ was determined by grid search using 5-fold cross-validation (CV).

2.2 Features

In this section, we describe four features used in both the NTCIR-9 RITE task and the NTCIR-10 RITE2 task.

Overlap ratios of characters (Char).

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

Overlap ratios of words and word pairs (Word).

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

Fulfillment tests with Predicate-Argument Structures (PAS).

The syntactic tree is converted to a set of predicate-argument structures to examine whether H covers all of the information in T. The predicate-argument structures used here are one 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

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

Table 1: Examples of sentence pairs that activate the feature f_{PAS}.

T	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 ...’
H	スーザン・トレスさんは脳死になった。 ‘Ms. Susan Torres became brain dead.’
T	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.’
H	日本で臓器移植法は施行された。 ‘The organ transplantation law became effective in Japan.’

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

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

(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} in the fulfillment test. The $f_{PAS} = 1$ only if all of the predicate-argument structures in H are subsumed by one of those in T, and otherwise $f_{PAS} = 0$. The phrase “p_1 subsumes p_2” means that p_1 and p_2 have the same predicate and all of the arguments of p_2 in p_1. For instance, a predicate-argument structure “行く (駅)” is subsumed by (P1) in the above example. Table 1 gives examples of sentence pairs in which $f_{PAS} = 1$. This feature appears to be a strong clue for the entailment.

To improve the coverage of this subsumption, the introduction of hyponym and hypernym relationships using WordNet was tested, but few pairs of nouns in T and H matched the relationships, so we abandoned the use of WordNet for this feature.

3. CASE-AWARE NOUN PHRASE MATCHING

We added new feature sets based on matching of noun phrases followed by case markers to capture finer syntactic structures than word matching.

3.1 Features with case-aware noun phrase matching

The feature sets represent the number of matchings of the noun phrases followed by the specific case markers in H and T. We focus on these case makers:

> ガ格 (“Ga-case”), ヲ格 (“Wo-case”), ニ格 (“Ni-case”), ト格 (“To-case”), デ格 (“De-case”), カラ格 (“Kara-case”), ヨリ格 (“Yori-case”), ヘ格 (“He-case”), マデ格 (“Made-case”)

For example, when a noun phrases "条約を" ('Treaty-ACC') appears both in H and T, a feature value "ヲ:1" is generated because the noun in ヲ格 ("Wo-case") is the same.

Three feature sets are defined for case-aware noun matching. The first one requires exact matching and the others allow flexible matching of noun phrases using ontologies. We compare the word classes of the pair of case-aware nouns in the other two feature sets. We use ontologies is to improve the matching accuracy. For example, "悪党" ('bad guy') and "悪役" ('badman') have also same meanings, and the different surface forms. To recognize the meaning match of the text pair, we used the word class in the comparison.

Matching of case-aware noun phrase (Case).

A simple matching algorithm is used for the *Case* feature. The *Case* feature uses surface forms of the input noun phrases for matching.

Matching of Case-aware noun phrase using ontology created from Wikipedia category (CaseOnt1).

We added the feature set *CaseOnt1* where the nouns are compared by word classes, instead of their surface forms. To get word class, we use Shibaki's ontology [4], an IS-A ontology generated automatically from Wikipedia, which contains 420,000 words and 34,000 hierarchized word classes. It is broad and can provide word classes for many words. Here the eight nodes in the first level in the Shibaki ontology are used as the word classes for matching.

Matching of case-aware noun phrase using ontology created from Wikipedia abstracts (CaseOnt2).

The *CaseOnt2* feature is also an extension of the *Case* feature and uses the same strategy as *CaseOnt1*. The value of the *CaseOnt2* feature is derived from comparisons of the word classes of the noun phrases. The difference between *CaseOnt1* and *CaseOnt2* is only the resource used for obtaining the word classes.

The pairs of a word and its word class were automatically extracted from the Wikipedia abstract with Kazama's method [2]. There were 250,000 instances and 33,000 word classes were extracted.

Table 2 shows some words and their word classes using the two resources. *Null* means that a resource does not have that word. Since one instance in Shibaki's ontology sometimes has several word classes, a word like "梁塵秘抄" ('Ryojin Hisho') can have several classes. In this case the word can have "芸術" ('art'), "文書" ('document'), "出版物" ('publication') and "平安時代の文学" ('literature in Heian period'). Because the word coverages of the two resources are different , words such "梁塵秘抄" ('Ryojin Hisho') and "日米和親条約" ('Japan-US Treaty of Amity and Friendship') belong to only one of the two resource.

4. TEMPORAL DIMENSION REDUCTION

We are focusing on a notion of *temporal dimension reduction* and added the new feature sets to compare the temporal expressions in the sentence pairs. Comparison of temporal expressions is a method to simplify the semantics of sentences expressed in a complex natural language into forms in which their semantic overlaps are computable.

Here we pick the time axis as a focal dimension and try to map an input sentence into a duration of years. This allows us to estimate the entailment relationships between the two sentences by verifying the overlap of their time intervals derived from the two sentences. In other words, this is an attempt to reduce the dimensionality of the semantics of the natural language expressions into a single dimension.

We introduce four feature sets coming from temporal expressions comparison. The first one is *Temporal*, which is the feature set used in the NTCIR-9 RITE. The others are new in this paper and expansions of the previous feature sets, to handle more temporal expressions.

Temporal expressions can be divided into non-numerical or numerical. *Temporal*, a feature set in the previous system, focuses only on numerical temporal expressions, such as the "1620 年代" ('1620s'). The new features can handle not only numerical temporal expressions, but also non-numerical temporal expressions to expand the capabilities of the temporal expression matching.

A non-numerical temporal expression is defined as a noun phrase with objectively defined start and end years. We call this a "historical entity" in the rest of the paper. To handle historical entities in a standard way with numerical expressions, each historical entity is mapped into a specific time interval (start year to end year). For example "日米和親条約" ('Japan-US Treaty of Amity and Friendship') is a historical entity, which can be mapped to $[1854, 1854]$ because it was signed in 1854.

4.1 Motivation of historical entity mapping

Our work in NTCIR-9 RITE showed that temporal expressions were important in recognizing textual entailment relationships. The feature set coming from temporal expressions comparison was effective not only for the EXAM subtask in which numerical temporal expressions appear frequently but also for the BC subtask.

In the previous work the overlap of the numerical temporal expressions in H and T were used for calculation of the feature value. Here we expand the matching of the time intervals by identifying the non-numerical temporal expressions.

Figure 1 illustrates the concept of the historical entity mapping. The semantics of the sentence can be represented in a word vector space of high dimensionality. The key idea of historical entity mapping is to map those words in the vector space into the time axis of a single dimension. Since a historical entity is a noun phrase with objectively defined start and end years, we can map it into a specific interval on the time axis. For examples, "日米和親条約" ('Japan-US Treaty of Amity and Friendship' can be mapped into $[1854, 1854]$ and "院政期" ('Cloistered Rule') can be mapped into $[1086, 1185]$.

We studied the appearances of historical entities in the data to evaluate the capabilities for historical entity mapping in NTCIR-10 RITE2. Out of 50 sentences randomly selected from EXAM training data set, 37 sentences had one or more numerical temporal expressions, and 22 sentences had one or more historical entities. Therefore such historical entities can be used to increase the likelihood of temporal matching.

Interestingly, a self-contradiction in a text may be detected by using temporal expressions extracted from a text. Table 3 shows an example. With the knowledge that the "世界人権宣言" ('Universal Declaration of Human Rights') was

Table 2: Examples of instances in two ontologies

historical entity	word classes in *CaseOnt1*	word classes in *CaseOnt2*
“平致頼”(‘Tairano-Muneyori’)	“人間”(‘human being’)	“将”(‘commanding’)
“梁塵秘抄”(“Ryojin Hisho”)	“芸術”(‘art’) “文書”(‘document’) “出版物”(‘publication’) “平安時代の文学 (“literature in Heian period”)	*null*
“日米和親条約” (‘Japan-US Treaty of Amity and Friendship’)	*null*	“条約”(‘treaty’)

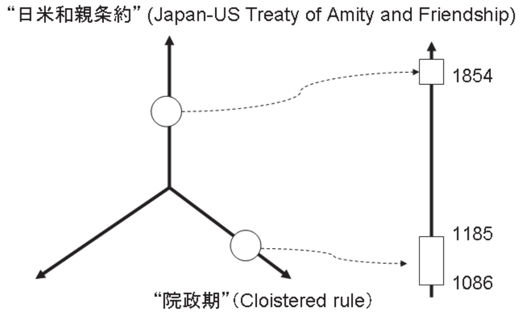
“日米和親条約” (Japan-US Treaty of Amity and Friendship)
1854
1185
1086
“院政期”(Cloistered rule)

Figure 1: Concept of temporal dimension reduction.

Table 3: An example of self-contradiction found by the time interval.

H	**第二次世界大戦前**，国際連盟の総会で， **世界人権宣言**が採択された。 ‘The Universal Declaration of Human Rights was adopted by the United Nations General Assembly before World War II.’

adopted in 1948 and the “第二次世界大戦” (‘Second World War’) ended in 1945, we can map them into $[1984, 1984]$, $[\infty, 1945]$. the mismatch of the time intervals of these two historical entities can be detected. It could be regarded as a trigger to detect that this hypothesis can never be entailed from any texts.

We didn’t use this self-contradiction detection in NTCIR-10 RITE2, even though it may be effective. Textual entailment recognition is a task to compare a pair of given texts for a decision, and thus a method using only one text is not suitable. The temporal features we used in NTCIR-10 RITE2 were calculated from comparisons of pairs of texts.

4.2 Temporal entity mapping using Wikipedia chronological tables

One approach for the creation of a lexical resource for temporal mapping is extraction from chronological tables in Wikipedia. We relied on articles whose titles are specific years such as “1192 年”(‘year 1192’) and “紀元前 4 年” (‘Year 4 BC’). These entries typically carry information on multiple people or events.

We regarded the Wikipedia titles appearing in “できごと” (‘Events’) section in such yearly articles as historical entities starting and ending in the title years. For example, since “鎌倉幕府” (‘Kamakura shogunate’) appears only in the “できごと” (‘Events’) section of the article titled “1192 年” (‘year 1192’), a noun “鎌倉幕府” (‘Kamakura shogunate’) can be mapped to 1192.

Table 4: Examples of historical entities from Wikipedia chronological tables

historical entity	start year
師範学校 (“Normal school”)	1872
サラゴサ条約 (“Treaty of Tordesillas”)	1529
バーデン大公カール (“Charles, Grand Duke of Baden”)	1806

From a total of 3,147 articles with chronological tables a total of 14,962 historical entities were extracted. To evaluate the accuracy of the relationships between the historical entities and the time intervals, we randomly picked 100 pairs and manually investigated whether the historical entities were correctly mapped to specific time intervals, and we found that 49 relationships of the 100 were correct.

Table 6 shows three pairs of historical entities and start years extracted by this method. The first and the second pairs are correct and the third one is incorrect. The first “師範学校” (‘Normal school’) appeared the in 1872 and the “サラゴサ条約” (‘Treaty of Tordesillas’) was signed in 1529. The year 1806 is not the start year of “バーデン大公カール” (‘Charles, Grand Duke of Baden’), but he got married in 1806. Most of the incorrect extractions were due to similar problems.

4.3 Temporal entity mapping using Wikipedia infoboxes and abstracts

In this section we describe another method to extract pairs of historical entities and time intervals from Wikipedia. This involved the Wikipedia Infoboxes and abstracts as the information sources.

A typical Wikipedia article contains an infobox that provides a structured summary record for the entity described in the article. Each infobox contains a set of attribute-value pairs that are manually constructed by the crowd editors. Since Wikipedia infoboxes have wide-coverage and high accuracy information, while some other knowledge resources such as DBpedia [1] and YAGO [6] are constructed from these Wikipedia infoboxes.

We focused on specific attributes that seem to be related

Table 5: Example of Wikipedia abstract.

クリストファー・コロンブス（伊: Cristoforo Colombo、英: Christopher Columbus、1451年頃 - 1506年5月20日）は探検家・航海者・コンキスタドール、奴隷商人。 'Christopher Columbus (Italian: Cristoforo Colombo; English: Christopher Columbus; 1451 - 20 May 1506) was an explorer, navigator, conquistador and slave trader.'
日米和親条約（にちべいわしんじょうやく）は、1854年3月31日（嘉永7年3月3日）に江戸幕府とアメリカ合衆国が締結した条約である。 'Japan-US Treaty of Amity and Friendship was a treaty concluded between the United States and the Tokugawa shogunate On March 31, 1854.'

to start or end years and extracted the values corresponding to those specific attributes. We used the following attribute keys in both English and Japanese to extract the start or end years:

> "death_date", "Date", "founded date", "established_date", "birthdate", "deathdate", "birth_date", "生年月日", "開催年月日", "没年月日", "開始日", "終了日", "誕生日".

If the infobox of an article contains any attribute name in this list, we tried to extract the start or end year from the abstract. Wikipedia abstracts are short descriptions of the articles. Wikipedia abstracts usually contain significant information such as hometown and birthday.

We found the two things through the analysis of Wikipedia abstracts:

- The first sentence in an abstract tends to have the start and end year.
- In particular, the phrase in the first parentheses tends to have the start and end year.

We only process the first sentence or the phrase in the first parentheses. If the first sentence in an abstract has parentheses, the search range is changed to the phrase in the parentheses. If there are no parentheses in first sentence of the abstract, we attempted to extract a temporal entity from the first sentence.

After changing the search range, we extracted each numerical temporal entity by using one simple regular expression. The regular expression we used was "[0-9]+ 年". Phrases that matched this pattern were regarded as temporal expressions. If there were multiple numerical expressions, we extracted the first two expressions as the start and end years. If there was only one numerical temporal expression, it was regarded as a start year.

From 843,784 articles in the Japanese Wikipedia, 88,900 pairs of a historical entity and a time interval were extracted. In the same way as in Section 4.2, we evaluated our extraction method on 100 randomly-selected pairs, and the accuracy was 87%.

Three examples of pairs extracted by this method are shown in Table 6. The first and second pairs are correct. "クリストファー・コロンブス" ('Christopher Columbus') was born in 1451 and "日米和親条約" ('Japan-US Treaty of Amity and Friendship') was signed in 1854. However the third one is incorrect. "平致頼" ('Taira-no-Muneyori') was not born, but died in 1011.

Table 6: Examples of historical entities from Wikipedia infoboxes and abstracts

historical entity	start year
クリストファー・コロンブス ("Christopher Columbusl")	1451
日米和親条約 ("Japan-US Treaty of Amity and Friendship")	1854
平致頼 ("Taira-no-Muneyori")	1011

4.4 Features with temporal expression matching

We describe four feature sets representing the relationships between temporal expressions in T and H. The first one is the feature set used in our NTCIR-9 system and the other is the new set proposed in this paper.

The main difference in the four features from temporal expression matching is the mapping function. The mapping function depends on the resource that provides the pairs of the historical entity and the time interval.

Let $x \in X$ be a temporal expression in the sentences, and let I be a time interval $[a, b] = \{y \in R | a \leq y \leq b\}$. Function $m : X \rightarrow I$ maps x to $[a, b]$.

Temporal expressions are extracted and they are mapped into year intervals using four rules:

Year: "N 年" ('the year of N') is converted to the year range $[N, N]$. All Japanese calendar schemes are 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.

Historical entity mapping: Historical entity is converted to specific time interval. For example, "日米和親条約" ('Japan-US Treaty of Amity and Friendship') can be mapped into $[1854, 1854]$.

Let I_x be the time interval $[a_i, b_i]$. Let I_y be the time interval $[a_j, b_j]$. The three functions which represent the relationships between I_x and I_y can be considered.

$$matchE(I_x, I_y) = \begin{cases} 1 & \text{if } a_i = a_j \wedge b_i = b_j \\ 0 & otherwise \end{cases} \tag{4}$$

$$include(I_x, I_y) = \begin{cases} 1 & \text{if } (a_j \leq a_i \wedge b_i < b_j) \\ & \vee (a_j < a_i \wedge b_i \leq b_j) \\ 0 & otherwise \end{cases} \tag{5}$$

$$matchP(I_x, I_y) = \begin{cases} 1 & \text{if } (a_i < a_j \wedge b_i < b_j) \\ & \vee (a_j < a_i \wedge b_j < b_i) \\ 0 & otherwise \end{cases} \tag{6}$$

Temporal expression matching, focusing on numerical temporal expressions (Temporal).

This feature uses the mapping function that can handle only the numerical temporal expression. Let $X = \{I_i | i = 1, 2...n\}$ and $Y = \{I_j | j = 1, 2...m\}$ be sets of time intervals coming from the temporal expressions in the sentence T and H, respectively.. The feature value is defined:

$$f_{match}(X, Y) = \begin{cases} 1 & \text{if } \sum_{x \in X} \sum_{y \in Y} f_{matchP}(x, y) \neq 0 \\ 0 & otherwise \end{cases}$$

$$f_{unmatch}(X, Y) = \begin{cases} 1 & \text{if } f_{match} = 0 \wedge X \neq \phi \wedge Y \neq \phi \\ 0 & otherwise \end{cases}$$

Temporal expression matching using a dictionary created from Wikipedia chronological tables (Time1).

This feature uses the mapping function that can handle the numerical temporal expressions and the historical entities. The first appearing temporal expression is used as the basis of matching

Let $X = \{I_i | i = 1, 2...n\}$ and $Y = \{I_j | j = 1, 2...m\}$ be sets of time intervals coming from the temporal expressions in the sentence T and H. Let I_{t1} and I_{h1} be the time intervals coming from the first appearing temporal expressions in T and H. We use $matchE(I_{t1}, I_{h1})$, $include(I_{t1}, I_{h1})$, $include(I_{h1}, I_{t1})$, $matchP(I_{t1}, I_{h1})$ as feature. We also use $f_{unmatch}(X, Y)$ as feature. $f_{unmatch}(X, Y)$ is defined:

$$f_{unmatch}(X, Y) = \begin{cases} 1 & \text{if } matchE(I_{t1}, I_{h1}) = 0 \\ & \wedge include(I_{t1}, I_{h1}) = 0 \\ & \wedge include(I_{h1}, I_{t1}) = 0 \\ & \wedge matchP(I_{t1}, I_{h1}) = 0 \\ & \wedge X \neq \phi \wedge Y \neq \phi \\ 0 & otherwise \end{cases}$$

For the calculations with *Time1* features, the historical entities are mapped to temporal expressions with the Wikipedia chronological tables. The temporal entity extraction method from the Wikipedia chronological tables was described in Section 4.2.

Temporal expression matching using a dictionary created from Wikipedia infoboxes and abstracts (Time2).

Time2 is almost the same as *Time1*. The feature value is calculated by using sets of the time intervals X, Y and the first appearing temporal expressions I_{t1}, I_{h1}. This feature set use $matchE(I_{t1}, I_{h1})$, $include(I_{t1}, I_{h1})$, $include(I_{h1}, I_{t1})$, $matchP(I_{t1}, I_{h1})$ and $f_{unmatch}(X, Y)$ as feature.

The difference between *Time1* and *Time2* is the mapping function. The historical entities are mapped to temporal expressions using Wikipedia abstracts and infoboxes. The temporal entity extraction method from the Wikipedia infoboxes and abstracts is described in Section 4.3.

Temporal expression matching, focusing on temporal expression about historical entities (TimeE).

TimeE has the same feature values as used in *Time1* and *Time2*. For the calculations with *TimeE* feature, only the first appearing temporal expression related to the historical entity are used. The mapping is based on the same method as used for the *Time2* features.

5. EXPERIMENTAL RESULTS

Table 7 shows the accuracies of the formal runs we submitted. The accuracies in parenthesis are the submitted formal-run results before the bug-fix. The logic is exactly the same as we have intended and as described in this paper. The best accuracy of our formal runs in the BC subtask was 74.9% which was 11.0 points better than the baseline provided by the organizer. The best accuracy of our formal runs in the EXAM subtask was 64.5% which was 8.0 points better.

To accurately evaluate the proposed method described in Sections 2, 3 and 4, we investigated all of the combinations of the feature sets. We basically selected feature sets for the formal run submissions using the average accuracies on 5-fold cross-validation with the training data. The *Char* feature, the overlap ratios of the characters, was used as a baseline here. Note that this is not the same as the baseline provided by the organizer. Table 8 shows some of the results of the experiments. This is what the a metric of row of the table mean:

1st to 3rd rows: Top-3 accuracies and feature set combinations ordered by BC's accuracy in cross-validation.

4th to 6th rows: Top-3 accuracies and feature set combinations ordered by EXAM's accuracy in cross-validation.

7th to 9th rows: Top-3 accuracies and feature set combinations ordered by BC's accuracy in formal run.

10th to 12th rows: Top-3 accuracies and feature set combinations ordered by EXAM's accuracy in formal run.

Last row: Our baseline

5.1 BC subtask

The best performance in cross-validation is 81.0% which is 2.1% above our baseline. The best performance in a formal run is 77.5% which 2.1% above our baseline. In the cross-validation and formal runs the features from temporal expression matching (*Temporal*, *Time1*, *Time2*, and *TimeE*) were effective. In particular, the *Time1* and *Temporal* features were effective in the cross-validation and formal runs.

Some features changed the performance between the cross-validation and formal runs. Although the *CaseOnt1* feature is effective in the cross-validation, it did not work well in the formal run.

5.2 EXAM subtask

The best performance in cross-validation is 67.6% which is 9.4% above our baseline. The best performance in a formal run is 68.8% which is 3.9% above our baseline.

Feature combinations working well in the cross-validation did not improve the performance in the formal run. In particular, *Case* and *Time2* did not work well in the formal run.

5.3 Discussion

To evaluate the performance of the case-aware noun phrase matching, we compared the results of the feature sets containing the case-aware noun phrase matching and the results of the feature sets without them. We used the feature sets in the previous system (*Char*, *Word*, *PAS*, *Temporal*) as the base feature sets in this comparison. Table 9 shows the comparison of the result where the top row shows the result of the base feature sets. The most of the feature sets having the case-aware noun phrase matching did not out-

Table 7: Accuracies of our formal runs (changed after submission)

	Case	*CaseOnt1*	*CaseOnt2*	*Temporal*	*Time1*	*Time2*	*TimeE*	*Char*	*Word*	*PAS*	accuracy
BC-01		✓		✓	✓			✓	✓		74.9 (74.3)
BC-02		✓	✓		✓		✓	✓	✓	✓	73.8 (73.8)
BC-03	✓	✓	✓	✓		✓	✓	✓	✓		73.4 (73.8)
EXAM-01	✓	✓		✓			✓		✓		61.6 (57.6)
EXAM-02	✓	✓	✓	✓			✓		✓	✓	61.8 (57.6)
EXAM-03	✓		✓	✓		✓	✓	✓	✓	✓	64.5 (60.9)

Table 8: Accuracies for comparison of the effective feature sets

											Cross-Validation		Formal Run	
	Case	*CaseOnt1*	*CaseOnt2*	*Temporal*	*Time1*	*Time2*	*TimeE*	*Char*	*Word*	*PAS*	BC	EXAM	BC	EXAM
1		✓	✓	✓	✓	✓	✓	✓	✓	✓	**81.0**	68.0	73.8	64.5
2			✓	✓	✓	✓		✓	✓	✓	80.8	68.0	77.1	64.1
3		✓	✓	✓	✓	✓	✓	✓	✓		80.8	67.6	73.8	64.7
4		✓	✓	✓				✓	✓		80.2	**68.6**	73.9	65.2
5		✓	✓	✓	✓	✓		✓	✓		80.5	68.4	73.9	62.1
6	✓		✓	✓				✓	✓	✓	80.5	68.4	73.9	65.0
7				✓	✓		✓	✓	✓	✓	79.7	68.0	**77.5**	67.4
8				✓	✓			✓	✓	✓	79.5	68.0	**77.5**	67.0
9			✓	✓	✓		✓	✓	✓		79.7	67.8	**77.5**	67.6
10					✓		✓	✓	✓		80.4	67.8	77.1	**69.0**
11					✓		✓	✓	✓	✓	80.2	67.8	77.1	68.5
12			✓	✓	✓		✓	✓	✓	✓	79.7	67.3	76.9	68.5
13								✓			78.90	64.7	75.4	64.1

Table 9: Accuracies for comparisons of the features from case-aware noun phrase matching

			Cross-Validation		Formal Run		
Case	*CaseOnt1*	*CaseOnt2*	BC	EXAM	BC	EXAM	average
			80.4	68.2	76.7	**66.3**	**72.9**
✓			80.3	67.8	73.9	65.2	71.8
	✓		80.0	67.3	73.9	64.7	71.5
		✓	80.5	67.8	**76.9**	**66.3**	**72.9**
✓	✓		79.5	67.5	73.9	65.2	71.5
✓		✓	80.5	68.4	73.9	65.0	72.0
	✓	✓	80.2	68.2	73.9	65.0	71.8
✓	✓	✓	80.0	**68.4**	73.9	65.0	71.8

Table 12: Accuracies for comparison the features from temporal expression matching

				Cross-Validation		Formal Run		
Temporal	*Time1*	*Time2*	*TimeE*	BC	EXAM	BC	EXAM	average
				80.2	65.9	76.6	64.7	71.8
✓				80.4	68.2	76.7	66.3	72.9
	✓			80.0	68.0	77.2	66.7	73.0
		✓		**80.5**	67.8	76.7	64.7	72.5
			✓	80.2	65.7	76.7	65.0	71.9
✓	✓			79.5	68.0	**77.5**	67.0	73.0
✓		✓		80.4	68.0	76.6	65.6	72.6
✓			✓	80.2	67.8	76.7	65.6	72.6
	✓	✓		80.4	68.0	76.9	67.2	73.1
	✓		✓	80.2	67.8	77.1	**68.5**	73.4
		✓	✓	79.9	67.5	76.4	64.5	72.1
✓		✓	✓	80.0	67.6	77.1	65.4	72.5
✓	✓		✓	79.7	68.0	**77.5**	67.4	73.2
✓	✓	✓		80.4	**68.4**	77.2	67.9	**73.5**
	✓	✓	✓	80.0	67.8	76.7	66.5	72.8
✓	✓	✓	✓	80.2	68.0	76.7	67.6	73.1

Table 10: The number of text pairs where case-aware noun phrase matching succeeded

	training set		evaluation set	
	BC	EXAM	BC	EXAM
Case	119	121	126	117
CaseOnt1	131	135	140	124
CaseOnt2	21	25	27	26

Table 11: The number of text pairs where temporal expressions can be compared

	training set		evaluation set	
	BC	EXAM	BC	EXAM
Temporal	35	83	57	119
Time1	184	241	219	209
Time2	37	102	63	123
TimeE	48	48	59	57

perform the base feature sets in both the BC subtask and the EXAM subtask.

We show the number of text pairs where case-aware noun phrase matching succeeded in Table 10. We can investigated the relationship between case-aware noun phrase matching and the accuracy by using Table 9 and 10. In *CaseOnt1* and *CaseOnt2*, the nouns were compared by word classes, instead of by their surface forms to increase the number of matchings. In *CaseOnt1* the numbers of the matching increase in each text set. On the other hand, there is a few matchings for *CaseOnt2*. The number of matchings is not correlated with accuracy. Although *CaseOnt2* has the smallest number of text pairs where case-aware noun phrases matching succeeded, the combination of the base feature sets and *CaseOnt2* achieved the highest accuracy.

The feature set coming from the temporal expression comparisons was effective in both the BC task and the EXAM task. We therefore believe that they will work well in many domains and are important features for textual entailment recognition.

We used the historical entity mapping to expand the capability of the temporal expression matching. The new feature sets can handle not only numerical temporal expressions, but also nonnumerical temporal expressions. Table 11 shows the number of text pairs with temporal expressions. The numbers of text pairs with temporal expressions of *Time1* and *Time2* are larger than for *Temporal*. *Time1* has the largest number of text pairs where temporal expressions can be compared, which were at least doubled. As explained in Section 4, the numbers of appearances of the historical entities are larger than those of the numerical temporal expressions in 50 texts of the EXAM subtask. It is important to treat not only from the numerical temporal expressions but also the historical entities.

Table 12 shows the comparisons of the results of the feature sets coming from the temporal expression comparisons. We used the feature sets (*Char*, *Word*, *PAS*) as the base feature sets. From Table 12 we can know that *Time1* showed very well. As discussed in Section 4, the accuracy of temporal entity mapping in *Time1* is 0.49 and the accuracy of temporal entity mapping in *Time2* is 0.87. The temporal entity mapping in *Time1* has high coverage and low accuracy, and outperformed temporal entity mapping in *Time2* with its low coverage and high accuracy. We conclude that the coverage of temporal entity mapping is important to expand the capabilities of the temporal expression matching. Although the resource used in *Time2* has more vocabulary than the resource used in *Time1*, the numbers of text pairs where temporal expressions can be compared of *Time2* are less than those of *Time1*. We can conclude that the coverage is more important than the accuracy in the temporal entity mapping.

6. CONCLUSIONS

In this paper we described the textual entailment recognition system we built for NTCIR-10 RITE2 and proposed two new kinds of features. One is based on case-aware noun phrase matching using ontologies. The other is based on temporal expression comparison after mapping the historical entities into specific time intervals.

Although the method mapping noun phrase to specific year achieved a certain result, it is needed to treat the behavior of the noun phrase to improve the performance. In *Time1* the system mapped the start year of “バーデン大公カール” (‘Charles, Grand Duke of Baden’) to 1806. However, this year is not his birth year, but his marriage’s year. In this paper we rapidly built the resources from Wikipedia by the naive methods to evaluate the effect of the temporal entity mapping. There is the possibility of improvement in the resource building and entity matching.

Acknowledgments

The authors thank Issei Yoshida for many helpful suggestions and valuable comments.

7. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC’07/ASWC’07, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[3] 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.

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

[5] 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.

[6] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW ’07, pages 697–706, New York, NY, USA, 2007. ACM.

[7] Y. Tsuboi, H. Kanayama, M. Ohno, and Y. Unno. Syntactic difference based approach for NTCIR-9 RITE task. In *NTCIR-9 Proceedings*, 2011.

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

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