Learning to Recognize Textual Entailment in Japanese Texts with the Utilization of Machine Translation

MINH QUANG NHAT PHAM, MINH LE NGUYEN, and AKIRA SHIMAZU,
Japan Advanced Institute of Science and Technology

Recognizing Textual Entailment (RTE) is a fundamental task in Natural Language Understanding. The task is to decide whether the meaning of a text can be inferred from the meaning of another one. In this article, we conduct an empirical study of recognizing textual entailment in Japanese texts, in which we adopt a machine learning-based approach to the task. We quantitatively analyze the effects of various entailment features, machine learning algorithms, and the impact of RTE resources on the performance of an RTE system. This article also investigates the use of machine translation for the RTE task and determines whether machine translation can be used to improve the performance of our RTE system. Experimental results achieved on benchmark data sets show that our machine learning-based RTE system outperforms the baseline methods based on lexical matching and syntactic matching. The results also suggest that the machine translation component can be utilized to improve the performance of the RTE system.

Categories and Subject Descriptors: I.2.7 [**Natural Language Processing**]: Text Analysis, Language Parsing and Understanding

General Terms: Theory, Languages

Additional Key Words and Phrases: Textual entailment, machine learning, machine translation, Japanese texts

ACM Reference Format:
Pham, M. Q. N., Nguyen, M. L., and Shimazu, A. 2012. Learning to recognize textual entailment in Japanese texts with the utilization of machine translation. ACM Trans. Asian Lang. Inform. Process. 11, 4, Article 14 (December 2012), 23 pages.
DOI = 10.1145/2382593.2382596 http://doi.acm.org/10.1145/2382593.2382596

1. INTRODUCTION

Recognizing Textual Entailment (RTE) is a fundamental task in Natural Language Understanding. It has been proposed with the aim of building a common applied semantic framework for modeling language variability [Dagan et al. 2006]. Given two text portions T (text) and H (hypothesis), the task is to determine whether the meaning of H can be inferred from the meaning of T.

RTE can potentially be applied in many NLP tasks, such as question answering or text summarization. Applications of RTE have been reported in several studies: question answering [Harabagiu and Hickl 2006] and information extraction [Romano et al. 2006]. In these studies, RTE has been integrated as an important component.

Authors' address: M. Q. N. Pham, M. L. Nguyen, and A. Shimazu, School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan; email: minhpqn@jaist.ac.jp.

DOI 10.1145/2382593.2382596 http://doi.acm.org/10.1145/2382593.2382596

For instance, in question answering [Harabagiu and Hickl 2006], an RTE component was used to determine if a candidate answer is the right answer for a question or not.

Recently, RTE tasks have received much attention in NLP research community. There have been several RTE shared tasks held by the TAC conference [Bentivogli et al. 2009] and many dedicated RTE workshops. However, to our knowledge, most published articles on RTE are for English, and only a very few are for other languages. It may be due to the fact that performance of state-of-the-art RTE systems significantly depends on RTE resources such as evaluation data, WordNet, or database of inference rules, which have been mainly developed for English. Studies of RTE for other languages rather than English are useful, because for a specific language, there are language-specific linguistic phenomena which we need to take into account.

In this article, we conduct an investigation of a machine learning approach to RTE task for Japanese. We build a lightweight RTE system that is based on machine learning. We formalize RTE task as a binary classification problem and apply machine learning algorithms to combine entailment features extracted from each pair of text T and hypothesis H. We evaluate our system using benchmark data sets from NTCIR9 RITE workshop [Shima et al. 2011]—the first attempt of constructing a common benchmark for evaluating systems which automatically detect entailment, paraphrase, and contradiction in texts written in Japanese. This work is an extension of previous work in Pham et al. [2011, 2012].

The use of bilingual corpora and machine translation for RTE tasks has been explored in the cross-lingual textual entailment recognition task [Mehdad et al. 2010, 2011] which is the task of recognizing textual entailment relationships between two text portions in different languages. Different from Mehdad et al. [2010, 2011], in this study, machine translation is used for monolingual RTE. In our system, a machine translation component is used to produce English translations of original Japanese texts, and both original Japanese texts and their translations are used to learn an entailment classifier. Our method is based on a reasonable assumption that if T entails H, then the translation of T should also entail the translation of H. Although to the best of our knowledge, state-of-the-art machine translation systems perform much worse than human translators, our method of using machine translation for Japanese RTE has some advantages. First, we can utilize RTE resources and tools for English languages, which are not available for Japanese. Second, some semantic relations of Japanese words which cannot be recognized due to the limitation of semantic resources in Japanese, may be recognized in English translations. We expect that English translations of a text and a hypothesis can provide more useful features for the RTE system to determine the entailment relationship in the pair more correctly. Pham et al. [2011, 2012] did not investigate how the performance of the RTE system changes if different MT engines are used. In this article, we present experimental results with two popular MT engines: Google Translator and Bing Translator.

Generally, in a machine-learning-based framework, there are two main elements which need to be considered: features and machine learning algorithms. These two aspects are extensively analyzed in this article. Especially, in order to improve the accuracy of the system, we apply several ensemble learning algorithms which train multiple classifiers and combine their outputs. In experiments, we adopt two ensemble learning approaches: bagging [Breiman 1996] and boosting [Freund and Schapire 1996]. Experimental results showed the performance improvement although the improvement is moderate because of the nature of training data and extracted features.

In short, the main contributions of our article are as follows.

— The article investigates a machine learning approach to recognizing textual entailment in Japanese. The effects of entailment features and machine learning algorithms on the performance of our Japanese RTE system are extensively analyzed. Experimental results showed that our proposed method significantly outperforms baseline methods using lexical matching and syntactic matching.

- We analyze the impact of various resources on the performance of our RTE system by conducting ablation tests.
- In our study, we propose using machine translation to improve the performance of our Japanese RTE system. Experimental results indicated the effectiveness of using machine translation for RTE on Japanese data sets.

The remainder of the article is organized as follows. Section 2 presents some related work to our research. Section 3 briefly presents background on two approaches to RTE and ensemble learning methods in machine learning. In Section 4, we describe our machine learning-based RTE system. Section 5 gives experimental settings and Section 6 presents experimental results achieved on two Japanese RTE data sets. In Section 7, we discuss hard phenomena observed in Japanese RTE data sets. Section 8 discusses approaches to Japanese RTE presented at NTCIR9-RITE. Finally, Section 9 gives conclusions and some remarks.

2. RELATED WORK

Mehdad et al. [2010] proposed the cross-lingual textual entailment (CLTE) task where text T and hypothesis H are written in different languages. A basic solution for CLTE task was proposed, in which a machine translation (MT) system is added to the front-end of an existing RTE engine. For instance, for a pair of English text and Spanish hypothesis, the hypothesis will be translated into English and then, the RTE engine will be run on the pair of the text and the translation of the hypothesis. This approach has advantages in terms of modularity but suffers from the error propagation caused by the MT component [Mehdad et al. 2011]. Another limitation of the basic solution is that it reduces the possibility to control the behavior of the RTE engine because of unpredictable errors propagated from the MT system.

Mehdad et al. [2011] proposed a new method for the CLTE task, which takes advantages of bilingual parallel corpora by extracting information from the phrase-table to enrich inference and entailment rules, and using extracted rules for a distance-based entailment system. The use of bilingual parallel corpora for monolingual textual entailment was also explored. The main idea of that work is to increase the coverage of monolingual paraphrase tables by extracting paraphrases from bilingual parallel corpora and use extracted paraphrases for monolingual RTE. The proposed method in Mehdad et al. [2011] allows a tighter integration of MT and RTE algorithms and avoids any dependency of external MT components.

Different from previous work mentioned above, our approach makes use of machine translation for monolingual RTE in a machine learning-based framework. In our machine learning-based RTE system, we combine both features extracted from data in original language and features extracted from translation data which were produced by an MT component to learn an entailment classifier. The main advantage of our proposed method is that it can make use of variability of words/phrases via translation.

3. BACKGROUND

3.1. Similarity/Distance Based Approaches

Several methods compute semantic similarity between the text T and the hypothesis H and decide if entailment relationship exists in the pair T/H by comparing the similarity score with a manually chosen threshold. For example, the pair T/H is decided to be an entailment pair if the similarity of T and H is equal or greater than a threshold. Text similarity between the text and the hypothesis can be computed based on surface or syntactic representations. For instance, one can use word overlap or count the number of common edges of dependency parses derived from T and H. For a more complete survey of text similarity measures used in RTE task, see Androutsopoulos and Malakasiotis [2010].

Similar to text similarity, distance between the text T and the hypothesis H can be used to decide the entailment relation of the pair T/H. Edit distance of the pair T/H is defined as the cost of the edit sequence (string or tree edits) needed to transform T into H. The intuition is that the smaller the edit distance between T and H is, the more likely that T entails H. Several methods that apply string edit distance [Levenshtein 1966] or tree edit distance [Zhang and Shasha 1989] have been reported [Kouylekov and Magnini 2005].

3.2. Machine Learning-Based Approaches to RTE

In these methods, the RTE task has been formulated as a classification problem. Multiple entailment features extracted from each pair T/H are combined using machine learning methods [Malakasiotis and Androutsopoulos 2007]. Features may be similarity measures applied on the pair or other features such as polarity difference between T and H.

3.3. Ensemble Learning Methods

Ensemble learning involves the procedures employed to train multiple learning machines and appropriately combine their outputs in order to obtain better prediction performance [Brown 2009; Dietterich 2000]. The principle of ensemble learning is that on average, committee decision should have better overall accuracy than individual predictions.

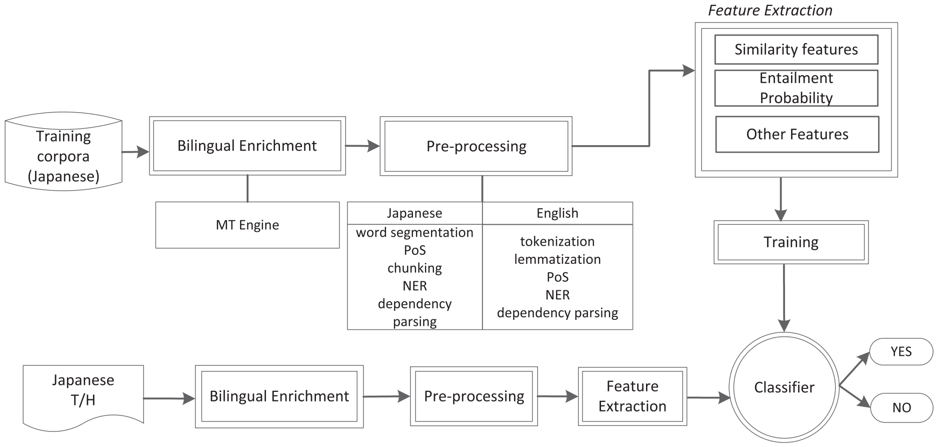
In our case, we applied ensemble learning methods for the binary classification problem. Specifically, we adopted three common ensemble learning algorithms: bagging [Breiman 1996], random forest [Breiman 2001], and AdaBoost [Freund and Schapire 1996].

3.3.1. Bagging. In the bagging (Boosting Aggregating) algorithm [Breiman 1996], each member classifier of the ensemble is constructed from a different training dataset, and the predictions are combined either by uniform averaging or voting over class labels. Each training dataset is created by uniformly sampling the total N data examples in the original training data set.

Similar to many ensemble methods, the base models in bagging methods should be unstable models which produce different behaviors with small changes to training data. In experiments, we choose Ripper rule learners [Cohen 1995] as base models. We also tried the random forest method [Breiman 2001] which combines the bagging algorithm with random subspace method [Ho 1998].

3.3.2. AdaBoost. The AdaBoost algorithm, short for adaptive boosting, introduced by Freund and Schapire [1996] has been seen as an effective boosting algorithm. In this section, we briefly describe the AdaBoost algorithm.

The input of the algorithm is a training set $(x_1, y_1), ..., (x_m, y_m)$ where each x_i belongs to the instance space X, and each label y_i is in a label set Y. In our case, $Y = \{-1, +1\}$. AdaBoost calls a given weak (base) learning algorithm repeatedly in a number of rounds $t = 1, ..., T$. There are weights associated with data examples x_i in the training set. At the beginning, all weights are set equally. The main idea of AdaBoost is to add one classifier on each round. Each new classifier is constructed by a learning algorithm so that the classification error on the weighted training data set is minimized. In order to achieve that, weights of data examples are updated on each



Japanese	English
word segmentation PoS chunking NER dependency parsing	tokenization lemmatization PoS NER dependency parsing

Fig. 1. Architecture of the Japanese RTE system.

round. Specifically, on each round, weights of incorrectly classified data examples are increased so that the base learner focuses on hard examples which are misclassified by previous classifiers.

Although, the actual performance of AdaBoost on a particular problem is dependent on data and the chosen weak learner, the algorithm has shown its advantages in several studies [Bauer and Kohavi 1999; Freund and Schapire 1996]. In experiments, we used "decision stumps" [Dietterich 2000] as weak learners.

4. PROPOSED METHOD

In our article, we adopt the machine learning approach to building an RTE system. The RTE task is formulated as a binary classification problem in which each instance consists of a pair of a text T and a hypothesis H.

In this section, we describe our RTE system. The RTE system is divided into five main modules as shown in Figure 1: bilingual enrichment, preprocessing, feature extraction, training, and classification.

First, each Japanese pair T/H is automatically translated into English using an MT engine. Then, in preprocessing, both the Japanese pair and its associated translation pair are analyzed. After that, extracted features from the pair and its translation pair are input to an entailment classifier to determine if the entailment relationship exists in the pair or not. The entailment classifier is trained on the training set consisting of pairs T/H with their gold labels.

In experiments, we investigate several machine learning algorithms for the RTE task: support vector machines (SVMs) [Vapnik 1998]; maximum entropy model [Berger et al. 1996]; and three ensemble learning algorithms: bagging [Breiman 1996], random forest [Breiman 2001], and AdaBoost [Freund and Schapire 1996].

4.1. Bilingual Enrichment

In order to make use of the bilingual constraint for RTE, the original RTE corpus in Japanese is automatically translated into English using Google Translator Toolkit[1]. In experiments, we try Microsoft Bing Translator[2] and compare the overall accuracy with the accuracy when we use Google Translator in the Bilingual Enrichment module.

[1] Google Translator Toolkit: http://translate.google.com/toolkit.

[2] Microsoft Bing Translator is available online on http://www.microsofttranslator.com/.

4.2. Preprocessing

4.2.1. Japanese Pairs. We use Cabocha tool [Kudo and Matsumoto 2002] for data preprocessing. For each pair, preprocessing consists of tokenizing, chunking, named entity recognition, and dependency parsing. Parsed content of each sentence is represented in XML format.

4.2.2. English Pairs. Each Japanese T/H pair in our corpus is associated with its English translation. We use Stanford-CoreNLP tool to perform preprocessing for English pairs[3]. Stanford-CoreNLP provides a set of fundamental natural language processing tools which can take raw English text input. At lexical level, we use the tool to perform tokenization, lemmatization, part-of-speech tagging, and named-entity recognition. At syntactic level, dependency parsing is done.

4.3. Feature Design

In the system, we train an entailment classifier on the training set consisting of annotated pairs T/H. Each pair T/H is represented by a feature vector $\langle f_1, ..., f_m \rangle$ which contains multiple similarity measures of the pair and some other features. For each training instance consisting of a pair T/H, features are extracted from both the original pair in Japanese and its associated English translation pair. In this section, we describe features used in the entailment classifier.

4.3.1. Similarity Features. A large part of the similarity features used in the entailment classifier is similar to features used in Malakasiotis and Androutsopoulos [2007]. We use different kinds of text similarity/distance measures applied on each pair T/H and its English translation pair. These measures capture how H is covered by T.

For each pair T/H (Japanese pair or English translation pair), text similarity and distance measures are applied on two pairs as follows.

— *Pair 1* is two sequences of words of T and H in surface forms. Punctuations and special characters are removed. Stop words are removed for English pairs.
— *Pair 2* is two sequences of words in T and H in base forms. Punctuations and special characters are removed. Stop words are removed for English pairs.

The preceding two pairs are representations for T and H when we compute similarity features.

We give a brief description of similarity features which are used to train the entailment classifier as follows.

Word overlap. The word-overlap feature captures the lexical-based semantic overlap between T and H, which is a score based on matching each word in H with some words in T [Dagan et al. 2007]. Japanese WordNet [Isahara et al. 2008][4] and English WordNet [Fellbaum 1998] are used in computing lexical matching. The matching criterion for two English words is the same as in Dagan et al. [2007]. An English word h_{ew} in H is considered a match with an English word t_{ew} in T if one of the following holds.

— h_{ew} has the same surface or base form with t_{ew}.
— h_{ew} is a synonym of t_{ew}.
— Hypernym or meronym distance from t_{ew} to h_{ew} is not greater than 3.

[3] Stanford CoreNLP is available on http://nlp.stanford.edu/software/corenlp.shtml.
[4] See http://nlpwww.nict.go.jp/wn-ja/index.en.html.

For Japanese, a word h_w in H is considered as a matching word of a word t_w in T if they have the same surface or base form, or h_w is hypernym, meronym, or entailment word of t_w.[5]

Levenshtein distance (string edit distance). The Levenshtein distance [Malakasiotis and Androutsopoulos 2007] of two strings is the minimum number of edit operations needed in order to transform a string to the other one. Allowable edit operations are deletion, insertion, or substitution of a single token. In our system, the Levenshtein distance from T to H is computed. We consider words as the smallest units when computing Levenshtein distances.

BLEU measures. BLEU score is a popular evaluation metric used in automatic machine translation [Papineni et al. 2002]. It measures how a translation generated by an MT system is close to reference translations. The main idea is to compute n-gram matching between automatically generated translations and reference translations. In the RTE problem, we used BLEU precision of H and T based on uni-gram, 2-gram, and 3-gram. In our case, we want to measure how the Text T subsumes the Hypothesis H, so T is cast as the reference translation and H is cast as the candidate translation. We used both BLEU measure and modified n-gram precision.

Longest common subsequence string (LCS). LCS feature computes the length of the longest common subsequence string between T and H [Hirschberg 1977]. The LCS feature is normalized by dividing its value by the length of H.

Jaccard coefficient. The Jaccard Coefficient is defined as the following.

$$\frac{|X \cap Y|}{|X \cup Y|}, \tag{1}$$

where X and Y are the sets of unique words of T and H, respectively; $|X|$ denotes the number of elements in the set $|X|$.

Dice coefficient. The dice coefficient is computed by the following.

$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|}, \tag{2}$$

where X and Y are the same as in the Jaccard Coefficient measure.

Manhattan distance. The Manhattan distance of two vectors $\vec{x} = \langle x_1, ..., x_n \rangle$ and $\vec{y} = \langle y_1, ..., y_n \rangle$ in an n-dimensional vector space is defined as the following.

$$d_1(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

Similar to Malakasiotis and Androutsopoulos [2007], in our case, n is the number of distinct words that occur in T and H; and x_i, y_i show how many times each one of these distinct words occur in T and H, respectively.

Euclidean distance is defined as follows.

$$d_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

[5] Due to technical problems, the use of Japanese WordNet for matching Japanese words is different from the use of English WordNet.

In this case, $\vec{x}$ and $\vec{y}$ are defined the same as those in the previous measure.

Jaro-Winkler distance. The Jaro-Winkler distance [Winkler 1999] is a measure of similarity between two strings. It is a variant of the Jaro distance metric.

The Jaro distance d_j of two given strings s_1 and s_2 is computed by the equation.

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), \tag{5}$$

where $|s_1|$ and $|s_2|$ are lengths of two strings s_1 and s_2, respectively and m is the number of matching characters. Two characters from s_1 and s_2 are considered matching if and only if they are identical and the difference of their positions is not greater than $\frac{\max(|s_1|,|s_2|)}{2} - 1$. Finally, t is half of the number of transpositions. The number of transpositions is the number of matching characters in different sequence order.

The Jaro-Winkler distance d_w is defined as the following.

$$d_w(s_1, s_2) = d_j(s_1, s_2) + (\ell \cdot p \cdot (1 - d_j(s_1, s_2))), \tag{6}$$

where ℓ is the length of the longest common prefix of s_1 and s_2, and p is a constant scaling factor which controls how much the score is adjusted upwards to having common prefixes.

Cosine similarity. The cosine similarity of two vectors $\vec{x} = \langle x_1, ..., x_n\rangle$ and $\vec{y} = \langle y_1, ..., y_n\rangle$ in an n-dimensional vector space is defined as the following.

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|}, \tag{7}$$

where $\vec{x}$ and $\vec{y}$ are binary vectors; $\|\vec{x}\|$ denotes the norm of the vector $\vec{x}$. Elements x_i and y_i indicate whether or not the corresponding word occurs in T or H, respectively.

4.3.2. Entailment Probability. The entailment probability that T entails H is computed based on the probabilistic entailment model in Glickman et al. [2005]. The main idea is as follows. The probability that the entailment relationship exists in the pair, $P(H|T)$ is computed via the probability that each individual word in H is entailed by T. The probability $P(H|T)$ is computed by the following equation.

$$P(H|T) = \prod_j P(h_j|T), \tag{8}$$

where the probability $P(h_j|T)$ is defined as the probability that the word h_j in H is entailed by T. The probability $P(h_j|T)$ is computed by the following.

$$P(h_j|T) = \max_i P(h_j|t_i), \tag{9}$$

where t_i is a word in T.

In Equation (9), $P(h_j|t_i)$ can be interpreted as the lexical entailment score between words t_i and h_j. By this decomposition, the overall probability $P(H|T)$ is computed by the following equation.

$$P(H|T) = \prod_j \max_i P(h_j|t_i) \tag{10}$$

The lexical entailment score of two words w_1 and w_2 is computed by using the word similarity score between them. For English, lexical entailment scores are computed based on Levenshtein distance as in MacCartney [2009].

$$P(w_1|w_2) = 1 - \frac{dist(w_1, w_2)}{max\left(length(w_1), length(w_2)\right)} \tag{11}$$

where $dist(w_1, w_2)$ is Levenshtein distance of two words w_1 and w_2.

Different from English, a Japanese word may be comprised of characters in different character systems. Furthermore, the length of a Japanese word in term of characters is short, so it is not reasonable to use the Levenshtein distance of two Japanese words based on their characters. Therefore, we use the Japanese thesaurus, Nihongo goitaikei [Ikehara et al. 1997] to compute the similarity of two Japanese words.

4.3.3. Dependency Relation Overlap Features. Dependency relation overlap has been used in paraphrase identification [Wan et al. 2006]. For RTE task, we compute the dependency relation overlap of H and T by the following equation:

$$RelationOverlap = \frac{\left|relations(H) \cap relations(T)\right|}{\left|relations(H)\right|} \tag{12}$$

where $relations(s)$ denotes the set of head-modifier relations of the sentence s.

In English, a head-modifier relation of a sentence is defined as a triple of a head word, modifier word and their relation type extracted from the dependency parse of the sentence. In Japanese, a head-modifier relation of a sentence is a pair of two words or two “bunsetsu” segments, one of which depends on the other.

4.3.4. Named-Entity Mismatch. In a pair T/H, if the hypothesis contains a named entity which does not occur in the text, the text may not entail the hypothesis. We use an indicator function π to compute the named-entity mismatch feature of T and H: $\pi(T, H) = 1$ if H contains a named-entity that does not occur in T and $\pi(T, H) = 0$, otherwise. We compute named-entity mismatch for both Japanese pairs and their associated English translation pairs.

4.3.5. Polarity Mismatch. The polarity mismatch in a pair T/H may indicate that T does not entail H. We compute polarity mismatch in a pair T/H using the Polarity Weighted Word List [Takamura et al. 2005]. In that list, each Japanese word is associated with a weight that indicates whether the word has positive meaning or negative meaning. We use an indicator function to capture if words in the root nodes of dependency parses of T and H have opposite polarity. The polarity mismatch is applied only on Japanese pairs.

5. EXPERIMENTS SETTING

5.1. Data Sets

The NTCIR9 RITE workshop [Shima et al. 2011] provided benchmark data for several Japanese RTE subtasks: binary-class subtask (BC subtask), multi-class subtask (MC subtask), Entrance Exam subtask (Exam), and RITE4QA subtask. In order to evaluate our system, we use data sets of BC subtask and Entrance Exam subtask. The BC subtask is the basic problem setting of RTE which is to determine whether the meaning of a hypothesis H can be inferred from the meaning of a text T. The Entrance Exam subtask is the same as BC subtask in terms of input and output, but data set of Entrance Exam subtask are created from actual college-level entrance exams. Therefore, data of Entrance Exam subtask may be closer to real-world data than those of

Table I. Data Statistics

Dataset	Y	N	Total
BC subtask - Dev set	250	250	500
BC subtask - Test set	250	250	500
Exam subtask - Dev set	204	295	499
Exam subtask - Test set	181	261	442

the BC subtask. The RITE4QA subtask is the same as BC Subtask and Exam subtask in terms of the form of input and output, but the subtask aims to measure the impact of RTE to a Question Answering system. The test set of RITEQA was automatically created toward that purpose. In this study, we evaluate the performance of our RTE system as an independent RTE system, so we do not use the test set of RITE4QA for evaluation. Another reason why we do not use the test set of RITE4QA is that the test set of RITE4QA is quite noisy, namely a Y label does not necessarily represent an entailment between two text segments.

A development set and test set are provided for the BC subtask and the Entrance Exam subtask. Each data set consists of pairs T/H along with their gold-standard labels "Y" or "N". For each subtask, we train an entailment classifier on the development portion and evaluate the trained classifier on the test portion. Table I shows statistical information of each data set.

5.2. Evaluation Measures

In experiments, classification accuracy and average F1 score are used to evaluate RTE methods. The F1 score for each label (Y or N) is computed as follows.

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \qquad (13)$$

In our case, the average F1 score is the average on F1 scores of two labels Y and N.

Since the label distribution in the test sets of the Exam subtask is unbalanced, the average F1 score is a better evaluation measures than classification accuracy for the Exam subtask.

5.3. Machine Learning Tools

In our machine learning based RTE framework, any machine learning algorithms can be used. In experiments, we investigate several machine learning algorithms for the task.

The first machine learning method used in experiments is support vector machines, which is a robust method for classification problems. We used libSVM [Chang and Lin 2011], an efficient SVM tool for classification problems. The second machine learning algorithm is maximum entropy model (MEM) [Berger et al. 1996]. We used the Maximum Entropy Modeling Toolkit (maxent)[6] for experiments.

In order to analyze effects of ensemble learning methods for the RTE task, we used Weka tool [Hall et al. 2009], an open source machine learning and data mining suite. Parameters in bagging, random forest and AdaBoost algorithm are selected by performing five-fold cross-validation on the development set of each subtask.

When we apply Bagging algorithm, we choose JRip, which is an implementation of RIPPER rule learner [Cohen 1995] as the base learner. The number of iterations in

[6] Tsuruoka's implementation of Maximum Entropy Model is available for download on http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/.

the bagging algorithm and the number of trees used in the random forest algorithm are tuned by performing five-fold cross-validation on the training set.

When we applied AdaBoost, we used "decision stumps" [Dietterich 2000] as weak learners. The only parameter of AdaBoost algorithm we need to tune is the number of iterations.

5.4. Baselines

5.4.1. Local Lexical Matching Method (LLM). A trivial baseline for the task is to randomly choose a label for each pair. In experiments, we use a stronger baseline which is based on local lexical matching between T and H. Lexical matching score computed on each Japanese pair is compared with a threshold value tuned on the development portion of each data set.

5.4.2. Predicate-Argument Matching Method (PA-matching). The second baseline which we use in experiments is the PA-matching method [Shibata and Kurohashi 2011]. The PA-matching method is based on matching text and hypothesis, considering predicate-argument structure as a basic unit of handling the meaning of text/hypothesis. In this method, wide-coverage relations between words/phrases were utilized in matching a text and a hypothesis. We refer to the predicate-argument matching as PA-marching.

5.4.3. Two-stage Method. The main problem of the PA-matching method comes from parsing error, the lack of lexical knowledge, and world knowledge. Therefore, Shibata and Kurohashi [2011] also proposed a "two-stage" method. The main idea of the "two-stage" method is as follows. First, the PA-matching method is applied. If "Y" label for binary-class subtask is obtained, then the result is selected; otherwise an SVM-based method which considers shallow features such as overlap ratio of characters and morphemes is applied.

5.5. Questions to Answer

Experiments in the current study are conducted to answer questions as follows.

- First, we determine whether the machine translation component which incorporates extracted features from English translation pairs can be utilized to improve the system performance of the Japanese RTE system. We also investigate how different MT engines affect the system performance.
- What features are effective for Japanese RTE data?
- How do various RTE resources such as Japanese WordNet, Polarity Weighted Words List, or Nihongo goitaikei affect the system performance?
- What is the effectiveness of ensemble learning methods on the performance of our Japanese RTE system?
- In this study, we analyze the main linguistic phenomena in Japanese RTE data sets that need to be considered.

6. EXPERIMENTS

6.1. Official Runs at NTCIR9-RITE

In Pham et al. [2011], we previously presented the official runs of our team at NTCIR9-RITE [Shima et al. 2011]. We submitted three runs for the BC subtask as follows.

- *Run 1* (SVM_bi) used libSVM [Chang and Lin 2011] as the machine learning tool and all features extracted from original Japanese pairs and their associated English translation pairs. We tuned parameters for learning on the development set by using the parameter selection tool in the libSVM package.

Table II. Official Runs on the BC Subtask at NTCIR9-RITE

Methods	**Accuracy**
SVM + all features (SVM_bi)	0.580 (290/500)
SVM + monolingual features (SVM_mono)	0.566 (283/500)
MEM + monolingual features (MEM_mono)	0.552 (276/500)

Table III. Official Runs on Exam Subtask at NTCIR9-RITE

Methods	**Accuracy**
Lexical matching (LLM)	0.622 (275/442)
SVM + all features (SVM_bi)	0.652 (288/442)
SVM + monolingual features (SVM_mono)	0.652 (288/442)

- *Run 2* (SVM_mono) used libSVM as the machine learning tool and monolingual features extracted from the original Japanese pairs. We compare the result obtained in Run 2 with the result of Run 1 to see if bilingual constraints can improve the performance of the system.
- *Run 3* (MEM_mono) used Maximum Entropy Model as the machine learning tool and monolingual features extracted from original Japanese pairs.

For the Exam Subtask, we submitted results obtained by SVM_bi, SVM_mono, and the method based on lexical matching (LLM). The submitted models are learned by using the development portion provided for each subtask. For the Exam subtask, we added tree edit distance measures [Kouylekov and Magnini 2005] applied on T/H pairs in feature extraction.

Tables II and III, respectively, show official results of our team at NTCIR9-RITE on the BC subtask and the Exam subtask.

Although our RTE system is very lightweight and does not require deep semantic analysis, among participated teams at NTCIR9-RITE, our proposed system (SVM_bi) obtained the first rank in BC subtask and the median rank in the Exam subtask [Shima et al. 2011].

6.2. Results

In the current article, we modified the feature extraction module. Specifically, tree edit distance is not used in the Exam subtask[7] and minor bugs in word matching are fixed.

Experimental results achieved on test sets of the BC subtask and the Exam subtask are shown on Tables IV and V, respectively.

For each machine learning algorithm applied in our framework, we run the system in two settings. In the first setting, we use only monolingual features extracted from Japanese pairs for training and testing. In the second setting, we use all features extracted from both original Japanese pairs and their associated English translation pairs. In Tables IV and V, the numbers in parentheses represent the performance improvement when all features are used.

Compared with results in Pham et al. [2011], the accuracies of the SVM_mono, SVM_bi, MEM_mono slightly decrease on BC subtask. However, the accuracy of SVM_bi on the Exam subtask is significantly improved. The reason for this may be that in this current article, we do not use features derived from tree-edit distances which did not show their advantages in system development, especially in the Exam subtask.

[7] In system development, the tree edit distance did not show its advantages.

Table IV. Experimental Results on the BC Subtask (with Google Translator)

Method	Acc	Avg F1
LLM Baseline	49.0%	0.480
PA-matching [Shibata and Kurohashi 2011]	49.2%	0.370
Two-stage method [Shibata and Kurohashi 2011]	51.6%	0.465
SVM + monolingual features	56.2%	0.560
SVM + all features	56.4% (+0.2)	0.564
MEM + monolingual features	54.8%	0.540
MEM + all features	55.6% (+0.8)	0.553
Bagging + JRip + monolingual features	57.8%	0.578
Bagging + JRip + all features	**58.6%** (+0.8)	**0.586**
RandomForest + monolingual features	54.2%	0.542
RandomForest + all features	55.8% (+1.6)	0.558
AdaBoost + decision stump + monolingual features	56.4%	0.564
AdaBoost + decision stump + all features	56.6% (+0.2)	0.566

Table V. Experimental Results on the Exam Subtask (with Google Translator)

Method	Acc	Avg F1
LLM Baseline	62.2%	0.612
PA-matching [Shibata and Kurohashi 2011]	59.3%	0.387
Two-stage method [Shibata and Kurohashi 2011]	65.6%	0.617
SVM + monolingual features	64.5%	0.616
SVM + all features	**69.2%** (+4.7)	**0.675**
MEM + monolingual features	65.8%	0.632
MEM + all features	68.5% (+2.7)	0.665
Bagging + JRip + monolingual features	61.5%	0.565
Bagging + JRip + all features	65.6% (+4.1)	0.610
RandomForest + monolingual features	62.0%	0.590
RandomForest + all features	64.7% (+2.7)	0.609
AdaBoost + decision stump + monolingual features	64.0%	0.616
AdaBoost + decision stump + all features	66.0% (+2.0)	0.634

In the BC subtask, our proposed machine learning-based methods significantly outperform three baselines. Especially, we obtained the best accuracy when we use bagging algorithm with all features (accuracy of 58.6%).

In the Exam subtask, combining SVM with all features results in the best performance (accuracy of 69.2%). However, ensemble learning methods did not show their effectiveness on the test set of the Exam subtask.

6.3. Effects of Using Bilingual Features

Tables IV and V show that generally, for each machine learning algorithm, using bilingual features results in better performance than using only monolingual features.

The effect of using bilingual features on the Exam subtask is more significant than in the BC subtask. A possible explanation for this result is that in the BC subtask the data was created so that simple surface overlap does not result in Y or N label easily [Shima et al. 2011]. The performance of the baseline using word overlap shows the evidence for our claim. It obtained low classification accuracy on the BC subtask (accuracy of 49%) while on the Exam subtask, it obtained quite good accuracy (accuracy of 62.2%).

In order to analyze the sensitivity of overall system performance to the machine translation component used in the system, in experiments, we tried two popular MT engines; Google Translator and Microsoft Bing Translator. Table VI shows results in

Table VI. System Performance with Different MT Engines

Setting	BC		Exam	
	Acc	Avg F1	Acc	Avg F1
SVM + Google Translator	56.4%	0.564	**69.2%**	0.675
SVM + Bing Translator	55.0% (−1.4)	0.549	67.0% (−2.2)	0.645
MEM + Google Translator	56.0%	0.557	68.5%	0.665
MEM + Bing Translator	55.6% (−0.4)	0.555	64.2% (−4.3)	0.617
Bagging + JRip + Google Translator	**58.6%**	0.586	65.6%	0.610
Bagging + JRip + Bing Translator	57.8% (−0.2)	0.578	65.6% (+0)	0.611
RandomForest + Google Translator	55.8%	0.558	64.7%	0.609
RandomForest + Bing Translator	57.8% (+2.0)	0.574	66.5% (+1.8)	0.631
AdaBoost + decision stump + Google Translator	56.6%	0.566	66.0%	0.634
AdaBoost + decision stump + Bing Translator	54.8% (−1.8)	0.545	61.5% (−4.5)	0.596

two subtasks when different MT engines are used. In Table VI, the numbers in the parentheses represent the difference in term of accuracies when Microsoft Bing Translator are used. The results indicate that the system performance is sensitive to the MT engine used, especially on the Exam subtask. On average, using Google Translator in the bilingual enrichment component has achieved better results. However, it is difficult to conclude that using "better" MT engines always results in a better overall system performance.

6.4. Machine Learning Algorithms

In the current study, we investigated several machine learning algorithms for the RTE task. Experimental results on Tables IV and V show that methods which apply support vector machines and maximum entropy models obtain more stable performance than ensemble learning methods. In the Exam subtask, using the SVM method with all features obtained the best performance among methods.

The results also indicated that ensemble learning methods show moderate performance improvement. In the BC subtask, using bagging algorithms with all features obtained the best performance among methods. However, using ensemble methods did not show performance improvement in the Exam subtask. This result might be related to the significant dependence of performance of ensemble learning methods on data sets and features used for each subtask.

6.5. Result Analysis

Tables VII and VIII show confusion matrices of the BC test and the Exam test, respectively. We compare the number of false-positive pairs and false-negative pairs predicted by some methods on the test set of each subtask. False-positive pairs are pairs which are predicted as "Y" pairs by a system while in gold standard, they are "N" pairs. False-negative pairs are pairs which are predicted as "N" pairs by a system while in gold standard, they are "Y" pairs.

Analyzing false-positive pairs predicted by methods SVM_bi and SVM_mono, we see that false-positive pairs mainly come from "N" pairs in which H is highly covered by T in terms of lexical, such as pair 15 in Figure 2. Among true-entailment pairs which our systems do not correctly classify, many pairs use complex entailment or paraphrase rules, such as pair 148 shown in Figure 2. Therefore, a large paraphrase table of phrases and a database of entailment rules may be important in order to improve the classification accuracy of the system.

In the BC subtask, the difference between using monolingual features and using all features is not so significant. On the other hand, in the Exam subtask, features derived

Table VII. Confusion Matrix (BC Test)

MEM_mono Method

Accuracy: 54.8% (274/500)

		Gold Label		
		Y	N	all
System	Y	168	144	312
	N	82	106	188
	all	250	250	500

MEM_bi Method

Accuracy: 55.6% (278/500)

		Gold Label		
		Y	N	all
System	Y	156	128	284
	N	94	122	216
	all	250	250	500

SVM_mono Method

Accuracy: 56.2% (281/500)

		Gold Label		
		Y	N	all
System	Y	126	95	221
	N	124	155	279
	all	250	250	500

SVM_bi Method

Accuracy: 56.4% (282/500)

		Gold Label		
		Y	N	all
System	Y	146	114	260
	N	104	136	240
	all	250	250	500

Bagging (all features) Method

Accuracy: 58.6% (293/500)

		Gold Label		
		Y	N	all
System	Y	143	100	243
	N	107	150	257
	all	250	250	500

from English translation pairs show significant contribution to performance of the system. As shown in Tables VII and VIII, the number of false-negative pairs predicted by SVM_bi is less than the number of false-negative pairs predicted by SVM_mono. It may indicate that the MT component used in SVM_bi provides more evidences for detecting entailment relationship in "Y" pairs through translation. Pair 243 in BC's test set and pair 12 in Exam's test set in Figure 2 show two examples in which SVM_bi correctly predicts their entailment labels while SVM_mono does not. A possible

Table VIII. Confusion Matrix (Exam Test)

MEM_mono Method

Accuracy: 65.8% (291/442)

		Gold Label		
		Y	N	all
System	Y	87	57	144
	N	94	204	298
	all	181	261	442

MEM_bi Method

Accuracy: 68.5% (303/442)

		Gold Label		
		Y	N	all
System	Y	97	55	152
	N	84	206	290
	all	181	261	442

SVM_mono Method

Accuracy: 64.5% (285/442)

		Gold Label		
		Y	N	all
System	Y	82	58	140
	N	99	203	302
	all	181	261	442

SVM_bi Method

Accuracy: 69.2% (306/442)

		Gold Label		
		Y	N	all
System	Y	103	58	161
	N	78	203	281
	all	181	261	442

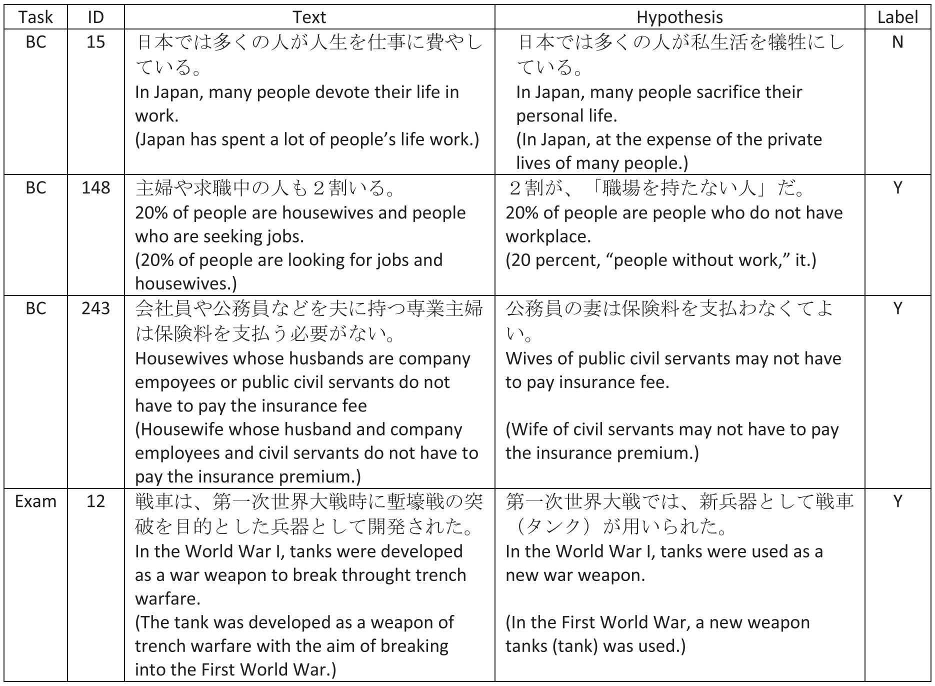
explanation is that in pair 243, the synonym relation between two English words “housewives” and “wives” can be recognized by using English WordNet while the relation between corresponding Japanese words was not detected by using Japanese WordNet. In pair 12, the English translation pair may strengthen the entailment “evidence” in the pair with its high word overlap score.

6.6. Feature Analysis

We conduct feature analysis in order to understand impacts of features on the performance of machine learning-based RTE systems.

We divide the features set into three categories as follows.

- *LemmaSim* consists of similarity features computed on base (lemma) form of each pair T/H.
- *SurSim* consists of similarity features applied on surface form of each pair T/H.
- *SynSem* consists of other features: entailment probability, dependency relation overlap feature, named-entity mismatch and polarity mismatch features.



Task	ID	Text	Hypothesis	Label
BC	15	日本では多くの人が人生を仕事に費やしている。 In Japan, many people devote their life in work. (Japan has spent a lot of people’s life work.)	日本では多くの人が私生活を犠牲にしている。 In Japan, many people sacrifice their personal life. (In Japan, at the expense of the private lives of many people.)	N
BC	148	主婦や求職中の人も２割いる。 20% of people are housewives and people who are seeking jobs. (20% of people are looking for jobs and housewives.)	２割が、「職場を持たない人」だ。 20% of people are people who do not have workplace. (20 percent, “people without work,” it.)	Y
BC	243	会社員や公務員などを夫に持つ専業主婦は保険料を支払う必要がない。 Housewives whose husbands are company empoyees or public civil servants do not have to pay the insurance fee (Housewife whose husband and company employees and civil servants do not have to pay the insurance premium.)	公務員の妻は保険料を支払わなくてよい。 Wives of public civil servants may not have to pay insurance fee. (Wife of civil servants may not have to pay the insurance premium.)	Y
Exam	12	戦車は、第一次世界大戦時に塹壕戦の突破を目的とした兵器として開発された。 In the World War I, tanks were developed as a war weapon to break throught trench warfare. (The tank was developed as a weapon of trench warfare with the aim of breaking into the First World War.)	第一次世界大戦では、新兵器として戦車（タンク）が用いられた。 In the World War I, tanks were used as a new war weapon. (In the First World War, a new weapon tanks (tank) was used.)	Y

Fig. 2. Example pairs in test sets. The meaning of text and hypothesis in English is shown for comprehension. The English texts in the parentheses show the real English translation results produced by the MT component.

Table IX. Feature Analysis

Setting	**BC test**	**Exam test**
SVM_mono + LemmaSim	55.8% (−0.4)	65.4% (+0.9)
SVM_mono + SurSim	57.8% (+1.6)	64.9% (+0.4)
SVM_mono + SynSem	55.2% (−1.0)	61.1% (−3.4)
SVM_mono + LemmaSim + SurSim	56.6% (+0.4)	64.2% (−0.3)
SVM_mono + LemmaSim + SynSem	53.8% (−2.4)	65.1% (+0.6)
SVM_mono + SurSim + SynSem	55.2% (−1.0)	65.1% (+0.6)
SVM_mono + All Features	56.2%	64.5%
SVM_bi + LemmaSim	54.2% (−2.2)	69.7% (+0.5)
SVM_bi + SurSim	55.8% (−0.6)	68.0% (−1.2)
SVM_bi + SynSem	52.4% (−4.0)	65.4% (−3.8)
SVM_bi + LemmaSim + SurSim	55.2% (−1.2)	66.1% (−3.1)
SVM_bi + LemmaSim + SynSem	54.6% (−1.8)	66.1% (−3.1)
SVM_bi + SurSim + SynSem	56.0% (−0.4)	66.5% (−2.7)
SVM_bi + All Features	56.4%	69.2%

Entailment classifiers are trained using above features subsets and a combination of them on the development sets. Table IX shows accuracies of various settings on the test sets of two subtasks. In Table IX, the numbers in the parentheses represent differences between settings and the system using all features.

Feature analysis indicated that similarity features significantly contribute to the performance of RTE systems. As shown in Table IX, without using similarity features (in the group LemmaSim and SurSim), the accuracies of SVM_bi and SVM_mono decrease significantly.

Table X. Affected Features in Ablation Tests

Ablated Resource	**Features**	**Status**
Japanese WordNet	Word Overlap	Changed
	Relation Overlap	Changed
Nihongo goi taikei	Entailment Probability	Removed
Polarity Word List	Polarity Mismatch	Removed

Table XI. Ablation Tests

Ablated Resources	**BC**	**Exam**
JWordNet	55.8% (−0.4)	64.5% (0.0)
Goi Taikei	54.2% (−2.0)	65.4% (+0.9)
Polarity Words	55.2% (−1.0)	65.4% (+0.9)
JWordNet + Goi Taikei	54.0% (−2.2)	65.1% (+0.6)
JWordNet + Polarity Words	55.2% (−1.0)	64.2% (−0.3)
Goi Taikei + Polarity Words	55.2% (−1.0)	64.2% (−0.3)
JWordNet + Goi Taikei + Polarity Words	54.6% (−1.6)	64.7% (+0.2)

Similarity features applied on surface form of each pair T/H and its English translation (in the group SurSim) are important in both subtasks. The contribution of features in the SynSem group to the performance of SVM_mono method is not so significant in both subtasks. However, the contribution of SynSem features is significant to the performance of SVM_bi method in the Exam subtask.

6.7. Ablation Tests

In the RTE task, it is interesting to know how additional resources or components contribute to the performance of our Japanese RTE system. This section presents ablation tests for two subtasks. We only analyze the effects of RTE resources and components to the SVM_mono method to avoid unpredictable errors propagated from the machine translation component.

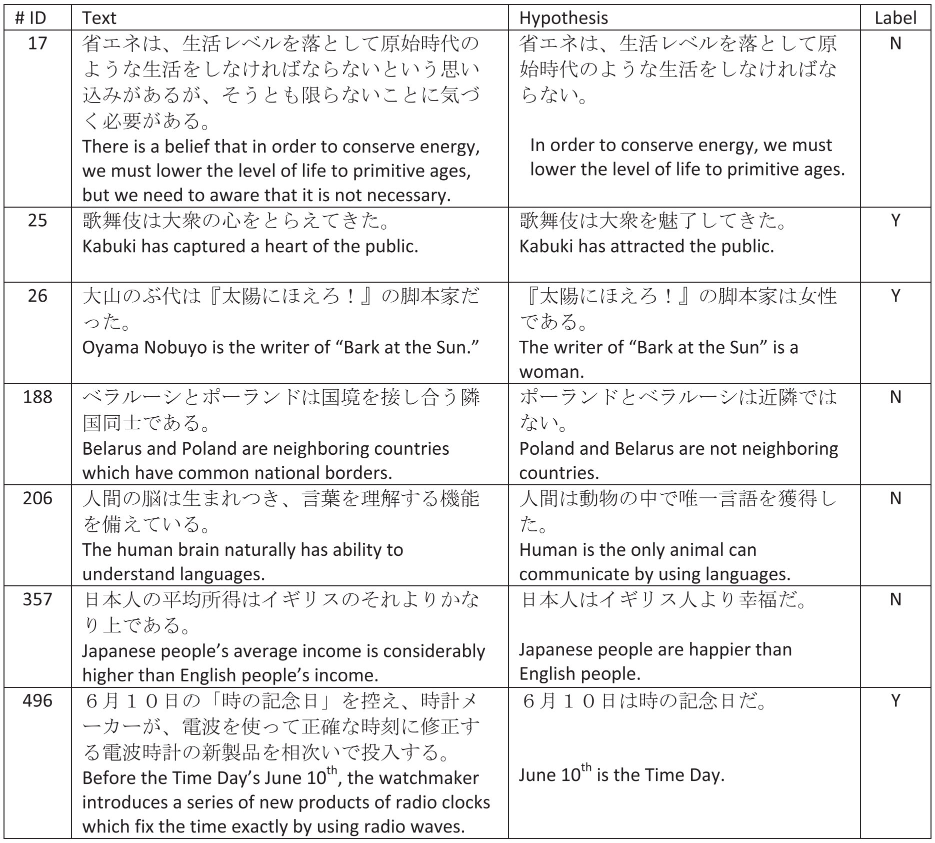
We conduct ablation tests as follows. For each test, we remove one or some resources used in the system. Features corresponding to an ablated resource will be omitted or only be changed depending on whether they are directly derived from the resource or not. Table X shows features corresponding to each ablated resource.

Table XI provides accuracies of the SVM_mono method without using some resources. The percentages shown in Table XI shows the accuracies of SVM_mono method without using some resources. the numbers in parentheses represent the difference of each setting with the system using full resources in terms of classification accuracies. As indicated in the table, the impact of additional resources on the performance of our system is not so significant in the Exam subtask. A possible explanation for this result is that word overlap of true-entailment pairs in data sets of the Exam subtask is very high.

The contribution of semantic resources is significant in BC subtask. It may be related to the way the data sets of BC subtask are created. Data sets of the BC subtask were created so that simple surface overlap does not result in Y or N label easily [Shima et al. 2011]. Therefore, we need to incorporate more semantic resources such as paraphrase corpora or world knowledge in order to improve the accuracy of RTE system in the BC subtask.

7. ENTAILMENT PHENOMENA

This section discusses entailment phenomena in the RTE corpus. We have observed the data and tried to classify the linguistic phenomena of textual entailment. We



# ID	Text	Hypothesis	Label
17	省エネは、生活レベルを落として原始時代のような生活をしなければならないという思い込みがあるが、そうとも限らないことに気づく必要がある。 There is a belief that in order to conserve energy, we must lower the level of life to primitive ages, but we need to aware that it is not necessary.	省エネは、生活レベルを落として原始時代のような生活をしなければならない。 In order to conserve energy, we must lower the level of life to primitive ages.	N
25	歌舞伎は大衆の心をとらえてきた。 Kabuki has captured a heart of the public.	歌舞伎は大衆を魅了してきた。 Kabuki has attracted the public.	Y
26	大山のぶ代は『太陽にほえろ！』の脚本家だった。 Oyama Nobuyo is the writer of “Bark at the Sun.”	『太陽にほえろ！』の脚本家は女性である。 The writer of “Bark at the Sun” is a woman.	Y
188	ベラルーシとポーランドは国境を接し合う隣国同士である。 Belarus and Poland are neighboring countries which have common national borders.	ポーランドとベラルーシは近隣ではない。 Poland and Belarus are not neighboring countries.	N
206	人間の脳は生まれつき、言葉を理解する機能を備えている。 The human brain naturally has ability to understand languages.	人間は動物の中で唯一言語を獲得した。 Human is the only animal can communicate by using languages.	N
357	日本人の平均所得はイギリスのそれよりかなり上である。 Japanese people’s average income is considerably higher than English people’s income.	日本人はイギリス人より幸福だ。 Japanese people are happier than English people.	N
496	６月１０日の「時の記念日」を控え、時計メーカーが、電波を使って正確な時刻に修正する電波時計の新製品を相次いで投入する。 Before the Time Day’s June 10th, the watchmaker introduces a series of new products of radio clocks which fix the time exactly by using radio waves.	６月１０日は時の記念日だ。 June 10th is the Time Day.	Y

Fig. 3. Example pairs in the BC subtask’s development set.

distinguish true-entailment pairs and false-entailment pairs. Table III shows some examples of T/H pairs in the BC-subtask’s development set.

7.1. True-Entailment Pairs

7.1.1. World Knowledge-Based Inference. In order to determine the label for a pair in this type, world knowledge is indispensable. In the pair, we cannot make a decision based on only textual evidences conveyed in the text and the hypothesis. For instance, in the pair 26 shown in Figure 3, we cannot determine whether the text entails the hypothesis if we do not know that the person called Oyama Nobuyo is a woman.

7.1.2. Inference Based on Paraphrasing and Entailment Words/Phrases. In pairs of this type, the decision can be made based on paraphrasing phrases or entailment words. For instance, in the pair 25 (Figure 3), it uses paraphrasing phrases pair, “captured the heart of the public” and “attracted the public.”

7.1.3. Hypotheses Are Facts Extracted from Texts. In a pair of this type, information conveyed in the hypothesis is a fact which can be extracted from the text. An example is the pair 496 as shown in Figure 3.

7.2. False-Entailment Pairs

7.2.1. Negation Structure. In a pair of this type, the hypothesis may use negation structures, and the meaning of the hypothesis contrasts with the meaning of the text. An example is the pair 188 as shown in Figure 3.

7.2.2. Hypothesis Discusses an Aspect of a Topic, Which Is Not Mentioned in the Text. In the pair 206 (Figure 3), the text said that human being could understand language. However, the hypothesis said that human being was the only animal that can acquire language, which is not mentioned in the text.

7.2.3. Factuality Degree of a Statement. In the pair 17, the hypothesis is completely covered by the beginning statement of the text, but the remaining part of the text inverses the veracity of the statement.

7.2.4. Wrong Inference. In pairs of this type, there are inferences that are not necessarily true. For instance, in the pair 357 (Figure 3), the text said that the average income in Japan was higher than that in England, but it is not necessarily true that Japanese people are happier than English people.

Textual entailment phenomena which were discussed above indicated that the BC subtask's data sets have very complicated nature, and extensive encoded world knowledge in the machine-readable form is indispensable for the RTE task.

8. DISCUSSION

Textual entailment recognition is not new in the field of NLP, but for the Japanese language, NTCIR9-RITE is the first shared-task in RTE. There are two main approaches to the RTE task in participant groups. The first approach tries to recognize entailment relation in a pair based on matching constituents of the text and the hypothesis. The matching may be computed in various levels such as lexical matching, syntactic matching, predicate-argument matching. For instance, Shibata and Kurohashi [2011] presented the PA-matching method that is based on matching text and hypothesis, considering predicate-argument structure as a basic unit of handling the meaning of text/hypothesis. Sugimoto [2011] presented a method of computing the overlap of the text and the hypothesis based on dependency triples which are extracted from the text and hypothesis.

The second approach is the machine learning-based approach. The main idea of this approach is to formalize the RTE task as a classification problem and use machine learning techniques to solve the classification problem. In the machine learning based approach, the important point lies in linguistic analyses and feature extraction. Most of the features are pair features, which are based on matching between constituents of the text and the hypothesis in various levels, such as lexical match, n-gram match, and syntactic dependency relation match, predicate-argument match, and syntactic differences [Akiba et al. 2011; Tsuboi et al. 2011].

In our study, we have adopted the machine learning-based approach. We investigated both features and machine learning algorithms. The novelty of our method is that we propose to use machine translation for RTE. Our proposed method is a very lightweight method. It does not require deep semantic analyses and extensive linguistic engineering. Nevertheless, experimental results achieved on Japanese data sets indicate the advantages of our proposed method. Our system obtained the best performance in BC subtask and competitive results in the Exam subtask among participant groups at NTCIR9-RITE.

However, our study still has some limitations. First, the system is not very precise at detecting hard false-entailment pairs in which H is highly covered by T. Second, due to the lack of entailment and paraphrase knowledge, our system fails to determine the entailment relationship in pairs that need complex inference. We plan to address these problems by developing an alignment component for RTE task and acquiring entailment/paraphrase rules from large text corpora.

9. CONCLUSION

We have presented an empirical study of recognizing textual entailment for Japanese. Our system is based on machine learning, in which multiple entailment features extracted from both original Japanese pairs and their English translation are combined to learn an entailment classifier. Extensive analyses and ablation tests have been conducted to quantitatively measure the effects of various entailment features, machine learning algorithms, and additional resources on the performance of our RTE system. Experimental results achieved on the two benchmark data sets indicated that our proposed method significantly outperforms the baseline method based on lexical matching and syntactic matching, and the machine translation component can be used to improve the performance of the RTE system.

REFERENCES

AKIBA, Y., TAIRA, H., FUJITA, S., KASAHARA, K., AND NAGATA, M. 2011. NTTCS textual entailment recognition system for the NTCIR-9 rite. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

ANDROUTSOPOULOS, I. AND MALAKASIOTIS, P. 2010. A survey of paraphrasing and textual entailment methods. *J. Artif. Intell. Res. 38*, 135–187.

BAUER, E. AND KOHAVI, R. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn. 36*, 1–2, 105–139.

BENTIVOGLI, L., DAGAN, I., DANG, H. T., GIAMPICCOLO, D., AND MAGNINI, B. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference (TAC'09)*.

BERGER, A. L., PIETRA, V. J. D., AND PIETRA, S. A. D. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist. 22*, 39–71.

BREIMAN, L. 1996. Bagging predictors. *Mach. Learn. 24*, 123–140.

BREIMAN, L. 2001. Random forests. *Mach. Learn. 45*, 1, 5–32.

BROWN, G. 2009. Ensemble learning. In *Encyclopedia of Machine Learning*, C. Sammut and G. Webb Eds.

CHANG, C.-C. AND LIN, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*, 27. http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

COHEN, W. W. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*. 115–123.

DAGAN, I., GLICKMAN, O., AND MAGNINI, B. 2006. The PASCAL recognizing textual entailment challenge. In *Proceedings of the Machine Learning Challenges Workshop (MLCW)*. Lecture Notes in Artificial Intelligence, vol. 3944, 177–190.

DAGAN, I., ROTH, D., AND MASSIMO, F. 2007. A tutorial on textual entailment. http://l2r.cs.uiuc.edu/∼danr/Talks/DRZ-TE-Tutorial-ACL07.ppt.

DIETTERICH, T. G. 2000. Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems (IWMCS'00)*. 1–15.

FELLBAUM, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

FREUND, Y. AND SCHAPIRE, R. E. 1996. Experiments with a new boosting algorithm. In *Proceedings of 13th International Conference on Machine Learning (ICML'96)*. 148–156.

GLICKMAN, O., DAGAN, I., AND KOPPEL, M. 2005. Web-based probabilistic textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment (RTE'05)*.

HALL, M., FRANK, E., GEOFFREY, H., PFAHRINGER, B., REUTERMANN, P., AND WITTEN, I. H. The Weka data mining software: An update. *SIGKDD Explor. 11*, 1.

HARABAGIU, S. AND HICKL, A. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the Association of Computational Linguistics (ACL'06)*. 905–912.

HIRSCHBERG, D. S. 1977. Algorithms for the longest common subsequence problem. *J. ACM 24*, 664–675.

HO, T. K. 1998. The random subspace method for constructing decision forests. *IEEE Trans. Patt. Anal. Mach. Intell. 20*, 8, 832–844.

IKEHARA, S., MIYAZAKI, M., SIRAI, S., YOKOO, A., NAKAIWA, H., OGURA, K., OOYAMA, Y., AND HAYASHI, Y. 1997. *Nihon-go goi taikei*. Iwanami, Japan (in Japanese).

ISAHARA, H., BOND, F., UCHIMOTO, K., UTIYAMA, M., AND KANZAKI, K. 2008. Development of Japanese wordnet. In *Proceedings of the Annual Conference on Language Resources and Evaluation (LREC'08)*. Nicoletta Calzolari (Conference Chair), Khalid Choukri Ed., European Language Resources Association.

KOUYLEKOV, M. AND MAGNINI, B. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment (RTE'05)*. 17–20.

KUDO, T. AND MATSUMOTO, Y. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning (COLING'02)*. 63–69.

LEVENSHTEIN, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*, 8, 707–710.

MACCARTNEY, B. 2009. Natural language inference. Ph.D. thesis, Stanford University.

MALAKASIOTIS, P. AND ANDROUTSOPOULOS, I. 2007. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (ACL-PASCAL'07)*. 42–47.

MEHDAD, Y., NEGRI, M., AND FEDERICO, M. 2010. Towards cross-lingual textual entailment. In *Proceedings of the Human Language Technologies 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'10)*. 321–324.

MEHDAD, Y., NEGRI, M., AND FEDERICO, M. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11)*. 1336–1345.

PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*. 311–318.

PHAM, M. Q. N., NGUYEN, M. L., AND SHIMAZU, A. 2011. A machine learning based textual entailment recognition system of JAIST team for the NTCIR-9 rite. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

PHAM, M. Q. N., NGUYEN, M. L., AND SHIMAZU, A. 2012. An empirical study of recognizing textual entailment in Japanese text. In *Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'12)* Part I. A. F. Gelbukh Ed., Lecture Notes in Computer Science Series, Vol. 7181, Springer.

ROMANO, L., KOUYLEKOV, M., SZPEKTOR, I., DAGAN, I., AND LAVELLI, A. 2006. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of the Conference on the European Chapter of the Association for Computational Linguistics (EACL'06)*. 401–408.

SHIBATA, T. AND KUROHASHI, S. 2011. Predicate-argument structure based textual entailment recognition system of the Kyoto team for the NTCIR-9 rite. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

SHIMA, H., KANAYAMA, H., LEE, C.-W., LIN, C.-J., MITAMURA, T., MIYAO, Y., SHI, S., AND TAKEDA, K. 2011. Overview of the NTCIR-9 rite: Recognizing inference in text. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

SUGIMOTO, T. 2011. Experiments for the NTCIR-9 rite task at the Shibaura Institute of Technology. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

TAKAMURA, H., INUI, T., AND OKUMURA, M. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 133–140.

TSUBOI, Y., KANAYAMA, H., AND OHNO, M. 2011. Syntactic difference based approach for the NTCIR-9 rite task. In *Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11)*.

VAPNIK, V. N. 1998. *Statistical Learning Theory*. John Wiley.

WAN, S., DRAS, M., DALE, R., AND PARIS, C. 2006. Using dependency-based features to take the “parafarce” out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop (ALTW’06)*.

WINKLER, W. E. 1999. The state of record linkage and current research problems. Tech. rep., Statistical Research Division, U.S. Census Bureau.

ZHANG, K. AND SHASHA, D. 1989. Simple fast algorithms for editing distance between trees and related problems. *SIAM J. Comput. 18*, 6, 1245–1262.

Received May 2012; revised July 2012; accepted September 2012