JAIST Participation at NTCIR-10 RITE-2

Minh Quang Nhat Pham
Japan Advanced Institute of Science And Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
minhpqn@jaist.ac.jp

Minh Le Nguyen
Japan Advanced Institute of Science And Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
nguyenml@jaist.ac.jp

Akira Shimazu
Japan Advanced Institute of Science And Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
shimazu@jaist.ac.jp

ABSTRACT

Textual entailment recognition is a fundamental problem in natural language understanding. The task is to determine whether the meaning of one text can be inferred from the meaning of the other one. At NTCIR-10 RITE-2 this year – our second participation in this challenge, we use the modified version of our RTE system used at NTCIR-9 RITE for four subtasks for Japanese: BC, MC, ExamBC, and Unit Test. In the feature aspect, we remove features which do not have benefits on development set of each subtask and add some new features. In the machine learning aspect, we employ the Bagging method – a robust ensemble learning method. We conduct extra experiments to evaluate the effects of features and the Bagging method on the accuracy of the RTE system.

Team Name

JAIST

Subtasks

BC, MC, ExamBC, UnitTest (Japanese)

Keywords

Machine Learning, Bilingual Features, Machine Translation, Ensemble Learning, Support Vector Machines

1. INTRODUCTION

NTCIR-10 RITE-2 challenge [12] continues the first version NTCIR-9 RITE [9] of RITE task. The challenge aims to provide the common benchmark for evaluating systems which recognize semantic relations between sentences written in Japanese and Chinese. The most focused semantic relation in NTCIR-10 RITE-2 is textual entailment which is a directional relationship between a text T and a hypothesis H, in which a human being reading T will infer that H is most likely true [5].

The participated system of JAIST team [8] for Japanese BC subtask at NTCIR-9 RITE adopted the machine learning approach to RTE. We formalize RTE as a binary classification problem and train an entailment classifier from the training data set provided for the task. In the learning model, various features are extracted, and most of them are based on the overlapping between the text and the hypothesis in a pair. The main novelty of the proposed system is the utilization of bilingual features extracted from English translation pairs of original Japanese pairs. Experimental results achieved on evaluation data sets showed that bilingual features can be used to improve the performance of the machine learning-based RTE system for Japanese.

This year, we use the participated system at NTCIR-9 RITE with some modifications in both feature and machine learning aspect. In feature extraction, we removed the some features which show negative effects in system development. We added some features from character-based representation of Japanese pairs. Details of those features will be described in next sections.

In the system used this year, ensemble learning methods, in addition to Support Vector Machines, are employed. Ensemble learning involves the procedures employed to train multiple learning machines and appropriately combine their outputs in order to obtain better prediction performance [2, 6]. The principle of ensemble learning is that on average, committee decision should have better overall accuracy than individual predictions. In some submitted runs for the challenge this year, we employ Bagging method [1] with Ripper rule learners [4] as base models.

JAIST team participated in four Japanese RITE subtasks: BC, MC, ExamBC, and UnitTest. We mainly focus on subtasks that use binary classification setting. For multi-class subtask, as a preliminary solution, we use the same feature space as that of BC subtask. However, experimental results show that feature space used for the BC subtask needs to be refined to deal with contradiction examples in data sets of MC subtask.

The rest of this paper is organized as follows. In Section 2, we describe the system used this year, in which we focus on modifications in the system. In Section 3, we present official runs of four sub-tasks that we participated. Section 4 presents extra experiments for each subtask. Finally, in Section 5, we conclude the paper and give some future directions.

2. SYSTEM DESCRIPTION

Figure 1 shows the architecture of our participated system. In training, the system takes as input a training set of Japanese T/H pairs with their gold labels. In the module Bilingual Enrichment, we use a machine translation engine to translate pairs in training set into English. In experiments, we used Google Translator Toolkit[1] as the machine translation engine although we can use any available Japanese-English MT systems. After performing preprocessing, the training set and their English translation are

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

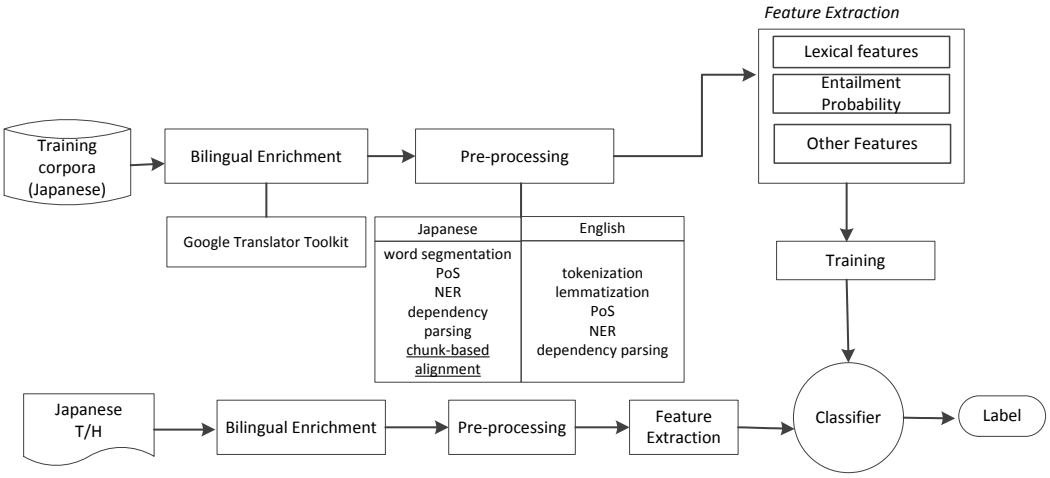

Figure 1: System Architecture of Japanese RTE system

input to the Feature Extraction component. For a pair, we extract features from both the pair itself and its English translation pair. Basically, for the English pair, we use similar features to the original pair. The combination of features of the original pair and bilingual features extracted from its English pair is used as the feature representation for the pair.

2.1 Preprocessing

For Japanese pairs, we use Cabocha tool [10] for data preprocessing. Preprocessing process consists of tokenizing, chunking, named entity recognition, and dependency parsing. Parsed content of each sentence is represented in XML format.

For English pairs, we use Stanford-CoreNLP tool to perform preprocessing for English pairs[2]. Stanford-CoreNLP provides a set of fundamental natural language processing tools which can take raw English sentences as 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.

2.2 Feature Design

Since the RTE system used this year is the modified version of our system at NTCIR-9 RITE, we do not repeat the details of features used in [8]. We focus on modifications made in the system. Table 1 shows the list of features used in the system.

In the current system, we do not use Euclidean distance-based features, named-entity mismatch, and polarity mismatch features. The reason is that in system development, those features do not show benefits on the accuracy of the RTE system.

In [8], we extract distance/similarity features from two representations of T/H pairs. The first representation is a pair of two sequences of words of T and H in surface forms. The second one is a pair of two sequences of words of T and H in base forms. In the current system, we try to extract some similarity features from character-based representations from Japanese T/H pairs. Those features include string edit distance, BLEU measures (1-grams, 2-gram, 3-gram), and longest common subsequence string.

[2]Stanford CoreNLP is available on: `http://nlp.stanford.edu/software/corenlp.shtml`

2.3 Machine Learning Algorithms

In the system used this year, we apply two machine learning methods. The first method is SVM method [11] – a robust method for classification problem. We employ libSVM tool [3] for experiments with SVM method.

The second method is the Bagging algorithm [1]. The basic idea of the Bagging algorithm is as follows. In the Bagging (Boosting Aggregating) algorithm [1], 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. In experiments, we choose Ripper rule learner [4] as the weak learner.

3. EVALUATION RESULTS

3.1 BC Subtask

We submitted three runs for the BC subtask (Japanese) as follows.

- **Run 1 (JAIST-JA-BC-01)**

  In this run, we use the development set for BC subtask at NTCIR-10 RITE-2 for training the classifier. We extract all features from both original pairs and their English translation pairs. SVM method is used for training and testing. We employed libSVM tool [3] in experiments. We tuned parameters for learning on the development set by using parameter selection tool provided in the package.

- **Run 2 (JAIST-JA-BC-02)**

  The setting of Run 2 is the same as that of Run 1 except that we employ ensemble learning methods. We adopted Bagging method [1] and used JRip implementation of RIPPER rule learner [4] as the base learner.

Table 1: List of features in the system. New features incorporated in the system this year are marked with the symbol ∗.

Feature	Japanese	English
Word overlap	x	x
Levenshtein distance	x	x
1-gram, 2-gram, 3-gram based BLEU measures (baseline)	x	x
1-gram, 2-gram, 3-gram based BLEU measures (modified)	x	x
Longest common subsequence string	x	x
Jaccard coefficient	x	x
Dice coefficient	x	x
Manhattan distance	x	x
Jaro-Winkler distance	x	x
Cosine similarity	x	x
Entailment Probability	x	x
Dependency Relation Overlap	x	x
Levenshtein distance based on characters (∗)	x	
Longest common subsequence string based on characters (∗)	x	
BLEU measures (modifier) based on characters (∗)	x	

Table 2: Results on BC subtask (JA)

Run	MacroF1	Acc.	Y-F1	Y-Prec.	Y-Rec.	N-F1	N-Prec.	N-Rec.
JAIST-JA-BC-01	75.56	76.23	71.51	71.94	71.09	79.61	79.27	79.94
JAIST-JA-BC-02	**76.47**	76.89	73.35	71.06	75.78	79.59	81.60	77.68
JAIST-JA-BC-03	73.08	73.77	68.75	68.75	68.75	77.40	77.40	77.40
Baseline	62.53	63.93	55.28	57.63	53.13	69.78	67.91	71.75
1^{st}-rank system	80.49	81.64	75.76	84.95	68.36	85.22	79.95	91.24

We used Weka tool [7], an open source machine learning and data mining suite for experiments with Bagging method. Parameters for Bagging methods are selected by performing fives-fold cross-validation on the training set.

- **Run 3 (JAIST-JA-BC-03)**

  We merge development set, test set at NTCIR-9 RITE, and development set at NTCIR-10 RITE-2 of BC subtask, and use the obtained set for training. In this run, we only extract monolingual features. The reason is that bilingual features does not show benefits on the training set used in the run. We apply the same ensemble learning method as in Run 2.

Table 2 shows results of three submitted runs on the test set of BC Subtask. As indicated on results, the classifier that is trained on development set of RITE-2 BC using Bagging method obtained the best result among our three submitted runs on the test set of RITE-2 BC subtask.

3.2 MC Subtask

MC is 4-way labeling setting for RTE task, in which entailment and non-entailment relations are divided into more refine classes: (forward/bi-directional) entailment, contradiction, and independence. Although we need a specific treatment for the contradiction relationship – complicated semantic relationship, in this preliminary study, we applied the machine learning-based framework used for BC subtask.

We submitted three runs for the MC subtask (Japanese) as follows.

- **Run 1 (JAIST-JA-MC-01)**

  In this run, we use the development set of MC subtask at NTCIR-10 RITE-2 for training the classifier. Since when we performing five-fold cross validation on the training set, bilingual features show negative effects on the accuracy of the system, we extract only monolingual features. We employed libSVM tool [3] in experiments.

- **Run 2 (JAIST-JA-MC-02)**

  We merge development set of RITE-2 MC, development set of RITE-1 MC and test set of RITE-1 MC, and use the obtained data set for training. Similar as Run 1, we extract only monolingual features and applied libSVM tool for training and testing.

- **Run 3 (JAIST-JA-MC-03)**

  We use the same setting as Run 2 except that we use both monolingual and bilingual features in the system.

Table 3 shows results of three submitted runs achieved on the test set of MC subtask. Names of submitted runs are shorten because of the space limitation.

As indicated on Table 3, the classifier trained on the development set of RITE-2 MC by using SVM method obtained the best F1 score among three submitted runs. We use only monolingual features in this run.

The results also so that using the same problem modeling and feature space used for the BC subtask cannot deal with contradiction examples in the test set of MC subtask. The precision and recall of contradiction label are very low. The reason is that similarity/overlapping features is not sufficient to capture contradiction phenomena. Those results suggest that we need a specific treatment for contradiction pairs.

Table 3: Results on MC subtask (JA)

Run	MacroF1	Acc.	B-F1	B-Prec.	B-Rec.	F-F1	F-Prec.	F-Rec.	C-F1	C-Prec.	C-Rec.	I-F1	I-Prec.	I-Rec.
MC-01	**52.60**	66.97	66.67	64.86	68.57	74.55	69.79	80.00	0.00	0.00	0.00	69.20	65.68	73.11
MC-02	52.27	65.33	63.51	60.26	67.14	71.84	59.68	90.24	5.97	33.33	3.28	67.76	80.52	58.49
MC-03	51.48	65.33	67.92	60.67	77.14	71.13	58.49	90.73	0.00	0.00	0.00	66.86	83.69	55.66
Baseline	26.61	45.44	0.00	0.00	0.00	56.18	43.01	80.98	5.41	15.38	3.28	44.88	54.36	38.21
1^{st}-rank	59.96	69.53	67.18	72.13	62.86	76.47	76.85	76.10	21.15	25.58	18.03	75.06	70.54	80.19

Table 4: Results on ExamBC subtask (JA)

Run	MacroF1	Acc.	Corr. Answer Ratio	Y-F1	Y-Prec.	Y-Rec.	N-F1	N-Prec.	N-Rec.
JAIST-JA-ExamBC-01	57.55	63.17	40.74	42.11	53.57	34.68	73.00	66.37	81.09
JAIST-JA-ExamBC-02	**59.04**	63.39	41.67	45.70	53.49	39.88	72.39	67.40	78.18
JAIST-JA-ExamBC-03	58.65	64.96	42.59	42.49	58.00	33.53	74.80	66.95	84.73
Baseline	54.77	56.47	32.41	45.98	44.15	47.98	68.55	65.38	61.82
1^{st}-rank system	67.15	70.31	55.56	56.96	64.71	50.87	77.34	72.76	82.55

Table 5: Results on UnitTest subtask (JA)

Run	MacroF1	Acc.	Y-F1	Y-Prec.	Y-Rec.	N-F1	N-Prec.	N-Rec.
JAIST-JA-UnitTest-01	67.36	79.67	87.40	96.05	80.19	47.31	34.38	75.86
JAIST-JA-UnitTest-02	**74.52**	89.21	93.87	93.87	93.87	55.17	55.17	55.17
JAIST-JA-UnitTest-03	29.46	30.71	38.83	86.89	25.00	20.10	11.67	72.41
Baseline	51.70	86.31	92.58	88.41	97.17	10.81	25.00	6.90
1^{st}-rank system	77.77	90.87	94.84	94.39	95.28	60.71	62.96	58.62

3.3 ExamBC Subtask

We submitted three runs for ExamBC Subtask as follows.

- **Run 1 (JAIST-JA-ExamBC-01)**

  In this run, we use the development set of Exam subtask at NTCIR-10 RITE-2 to train the classifier. Both monolingual and bilingual features are extracted; and libSVM tool is applied.

- **Run 2 (JAIST-JA-ExamBC-02)** We use the same setting as that of Run 1 except that the combination of DevSet of RITE-1 ExamBC, test set of RITE-1 ExamBC, and DevSet of RITE-2 ExamBC is used for training.

- **Run 3 (JAIST-JA-ExamBC-03)**

  We use the same setting as Run 1 for this task except that we extract only monolingual features.

Table 4 shows the results of three submitted runs for ExamBC subtask. Run 2 obtained the best F1 score on the test set. The use of bilingual features did not show benefits on the test set. The F1 score of the Run 1 which used all features is lower than that of the Run 2 which used only monolingual features.

3.4 UnitTest

We used the same settings as those of three runs for BC subtask to generate three runs for UnitTest subtask.

Table 5 shows results achieved on test set of UnitTest subtask. Run 2 which used Bagging method obtained the best result among our three runs for the subtask. The accuracy of Run 3 is significantly low. A possible explaination might be that in Run 3, we used the combination of data sets of NTCIR-9 RITE and the development set of NTCIR-10 RITE2 for training the classifier, so the training set may contain noisy data. Currently, our system cannot deal with such a domain adaptation problem. However, one may argue that the perforamance of Run 3 in BC subtask, which used the same setting as that in the Unit Test is not very low. The main reason is that the test set of the BC subtask is much more balanced than that of the Unit Test. Specifically, in the test set of the Unit Test subtask, the number of positive examples is much greater than that of negative examples.

4. EXTRA EXPERIMENTS

In this study, we investigate the effects of bilingual features and the ensemble learning method on the accuracy of our RTE system. Therefore, for each subtask, conducted extra experiments as described below.

For each subtask, we use the development set for training. The exception is the UnitTest subtask, in which we use the development set of BC subtask for training. We apply two machine learning methods: SVM method and Bagging method. We train classifiers by using those methods in two settings: i) using only monolingual features; and ii) using all features by combining monolingual features with bilingual features. As the result, we analyse four experimental settings for each subtask.

Table 6, 7, 8, and 9 show evaluation results achieved on test set of each subtask. In each table, we use abbreviations: "mono" for monolingual features, and "bi" for bilingual features.

As indicated in Table 6, combining monolingual and bilingual features slightly improved macro F1 and accuracy on the test set of BC subtask. The Bagging method obtained higher macro F1 and accuracy than those obtained by SVM method with the same feature space. As shown in Table 7, the Bagging method obtained completive results compared with SVM method. In MC subtask, using bilingual fea-

Table 6: Extra experiments on BC subtask (JA)

Setting	MacroF1	Acc.
SVM + mono	75.14	75.57
SVM + mono + bi	75.56	76.23
Bagging + mono	76.12	76.56
Bagging + mono + bi	**76.47**	**76.89**

Table 7: Extra experiments on MC subtask (JA)

Setting	MacroF1	Acc.
SVM + mono	**52.60**	66.97
SVM + mono + bi	52.50	**67.34**
Bagging + mono	51.48	65.51
Bagging + mono + bi	52.49	66.61

Table 8: Extra experiments on ExamBC subtask (JA)

Setting	MacroF1	Acc.
SVM + mono	58.65	64.96
SVM + mono + bi	57.55	63.17
Bagging + mono	58.53	61.61
Bagging + mono + bi	**60.84**	63.84

Table 9: Extra experiments on UnitTest subtask (JA)

Setting	MacroF1	Acc.
SVM + mono	76.57	87.55
SVM + mono + bi	67.36	79.67
Bagging + mono	**77.77**	**90.87**
Bagging + mono + bi	74.52	89.21

tures only achieved improvements when we applied Bagging method.

In the Exam BC subtask, we achieved the highest macro F1 when we apply Bagging method and extract all features. In the UnitTest subtask, using bilingual features did not show benefits while Bagging method consistently achieves higher macro F1 and accuracy than those obtained by SVM method.

5. CONCLUSIONS

We have presented our participated system at NTCIR-10 RITE-2. This year, we modified the system used at NTCIR-9 RITE-2. In the system used this year, we applied Bagging learning method with Ripper rule learners as base models. Experimental results show that Bagging method consistently obtained better or competitive results on four subtasks we participated. The effectiveness of using bilingual features has been shown on the BC subtask and some settings of other subtasks. The experimental results also indicated that using the machine learning-based system with current feature set is not enough for MC subtask and we need a specific treatment for contradiction pairs.

6. REFERENCES

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[2] G. Brown. Ensemble learning. *In Encyclopedia of Machine Learning, C. Sammut and G. Webb, Eds*, 2009.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] W. W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[5] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. *MLCW, LNAI*, 3944:177–190, 2006.

[6] T. G. Dietterich. Ensemble methods in machine learning. In *In Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[7] M. Hall, E. Frank, H. Geoffrey, B. Pfahringer, P. Reutermann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

[8] M. Q. N. Pham, M. L. Nguyen, and A. Shimazu. A machine learning based textual entailment recognition system of jaist team for ntcir9 rite. In *In NTCIR9 Proceedings*, 2011.

[9] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of ntcir9 rite: Recognizing inference in text. In *In NTCIR9 Proceedings*, 2011.

[10] Y. M. Taku Kudo. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[11] V. N. Vapnik. *Statistical learning theory*. John Wiley, 1998.

[12] 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.