KitAi: Textual Entailment Recognition System for NTCIR-10 RITE2

Kazutaka Shimada Yasuto Seto Mai Omura Kohei Kurihara

Department of Artificial Intelligence
Kyushu Institute of Technology
{shimada, y_seto, m_omura, k_kurihara}@pluto.ai.kyutech.ac.jp

ABSTRACT

This paper describes Japanese textual entailment recognition systems for NTCIR-10 RITE2. The tasks that we participated in are the Japanese BC subtask and the ExamBC subtask. Our methods are based on some machine learning techniques with surface level, syntax and semantic features. We use two ontologies, the Japanese WordNet and Nihongo-Goi-Taikei, and Hierarchical Directed Acyclic Graph (HDAG) structure as the syntax and semantic information. For the ExamBC task, the confidence value from a classifier is important to judge the correctness as the entrance exams. To predict a suitable confidence value, we apply a weighting method of each output from several classifiers. In formal runs, the best accuracy rates in the methods for the BC and the ExamBC tasks were 77.11 points and 59.84 on the macro F1 measure, respectively. Although the method based on SVMs was better than others in terms of the macro F1 measure, the weighted scoring method produced the best performance for the correct answer ratio (45.4%).

Team Name

KitAi / Kyushu Institute of Technology (Department of Artificial Intelligence)

Subtasks

BC and ExamBC tasks (Japanese)

Keywords

Correspondence, Edit Distance, Ontologies, Hierarchical Directed Acyclic Graph, Weighted Scoring

1. INTRODUCTION

This paper describes Japanese textual entailment recognition systems for NTCIR-10 RITE2 (the Japanese BC subtask and the ExamBC subtask) [8]. Our methods, KitAi[1], are based on some machine learning techniques such as SVM. The basic features are based on surface-based alignment. However, a simple bag of words feature is generally insufficient. Therefore, we introduce semantic information. As the semantic information, we use two ontologies; the Japnaese WordNet [1] and Nihongo-Goi-Taikei [2]. In other words, we apply a surface-based alignment process with the semantic information to our textual entailment recognition systems.

However, the surface-based alignment process can not handle the structural information of each sentence, such as dependency relations. Therefore, we introduce a structure feature with semantic and grammatical information. We use the Hierarchical Directed Acyclic Graph (HDAG) structure [7] to compute a similarity between t_1 and t_2. The HDAG structure can handle the structural information of a sentence and semantic information of each word in the sentence.

For the ExamBC task, a confidence value of each output is one of the most important factors because of the correctness as the answer of an examination question. Therefore we introduce a weighted scoring method to estimate a suitable confidence value for the selection process.

In the next section, we describe features and methods about our textual entailment recognition systems. Next, we discuss our experimental results on the development data and formal run in Section 3. Finally, we conclude our methods in Section 4.

2. SYSTEM DESCRIPTION

First, we explain features for our methods. Then, we describe classifiers for the BC and ExamBC tasks.

2.1 Features

The feature set for our methods consists of three types of linguistic information; (1) word correspondence, (2) HDAG and (3) others. For the feature extraction process, we use JUMAN[2] as a morphological analyzer and KNP as a dependency parser[3]. Finally, we obtain 36 features for our textual entailment recognition systems.

2.1.1 Word Correspondence

The basic features in our method are based on correspondence between t_1 and t_2 in surface level. We compute the rates of words of t_1 containing in t_2 and words in t_2 containing in t_1, respectively. We also compute the WER (word error rate) score based on the edit distance.

For these features, we use three types of input lists for t_1 and t_2: the all word list, the content word list and the simplified word list. The all word list contains all words in t_1 and t_2. The content word list contains nouns, adjectives, verbs in t_1 and t_2. The simplified word list contains words linking directly to the main predicate verb in each sentence.

[1]Short of *K*yushu *I*nstitute of *T*echnology (Department of *A*rtificial *I*ntelligence). The English meaning is "expectation."

[2]http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[3]http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

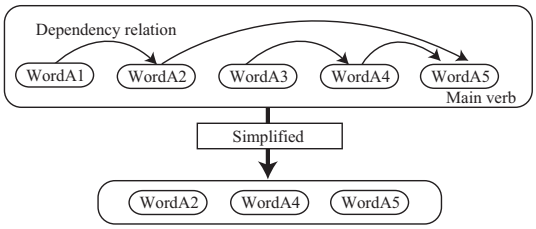

Figure 1: The simplification process.

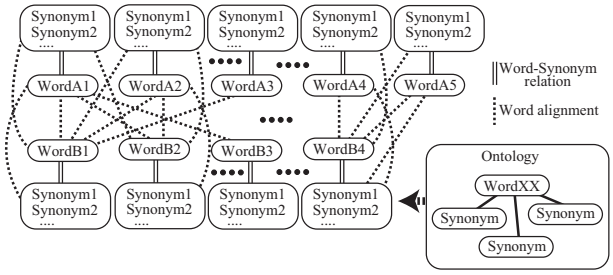

Figure 2: The word correspondence with ontologies.

Figure 1 shows the simplification process of a sentence. By using the simplified word list, we can compute a similarity of compressed meanings between sentences.

Furthermore, we extend each word by using two ontologies: the Japnaese WordNet [1] and Nihongo-Goi-Taikei [2]. For the Japnaese WordNet, we use the synonyms database, which is created by synsets and manually annotated. For Nihongo-Goi-Taikei, we use words that belong to the same semantic class for each word. Figure 2 shows the process of calculation of word correspondence degrees. We compute the edit distance by using the DP matching.

2.1.2 HDAG

To compute a similarity between two sentences, bag-of-words representation is the most general way to express features of them. However, it is insufficient to represent the features of each sentence because of lack of relations between words.

To solve the problems, Suzuki et al. [7] have reported a new graph-based approach, called Hierarchical Directed Acyclic Graph kernels (HDAG). The method can handle many linguistic features in a sentence and includes characteristics of sequence and tree kernels. The HDAG is a hierarchized graph-in-graph structure. It represents semantic and grammatical information in a sentence. We have used the HDAG for a sentiment sentence extraction task [6].

In this task, we introduce this structure for the similarity calculation. We apply three layers to the structure: pos layer, semantic layer and word layer. Figure 3 shows an example of the structure with the three layers. Our method computes the similarity of the HDAGs of two sentences.

2.1.3 Others

We add some features to our method. First, we use the number of negation words in each sentence. The inconsistency of the number of negation words between two sentences has the potential of the disagreement of entailment between them.

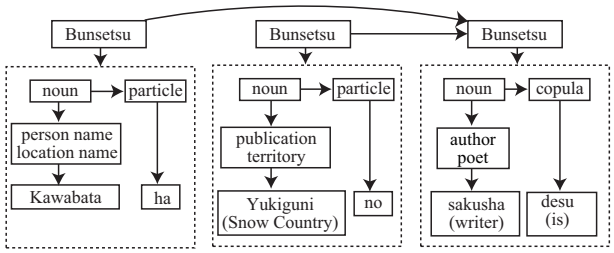

Figure 3: An example of HDAG structures.

Next, we compute the length values of the all word list and content word list. If the length values of two sentences are extremely-different, it indicates that the two sentences disagree with high probability.

2.2 Methods

For the BC task, we use three classifiers with the open source software Weka[4]. They are SMO, Logistic and Rotation Forest. SMO (MethodBC1) is a support vector classifier with John Platt's sequential minimal optimization algorithm [3]. Logistic (MethodBC2) is a multinomial logistic regression model with a ridge estimator. Rotation Forest (MethodBC3) is a regression model with a base learner [5]. The base learner is the C4.5 algorithm [4]. These methods are determined heuristically.

For the ExamBC task, we use AdaBoost with SMO (MethodEX1) and Logistic (MethodEX2) in a similar way. We focus on a weighted scoring approach as the 3rd method (MethodEX3) for the ExamBC task. The reason that we apply the weighted score to the ExamBC task is that the task evaluates the correct answer ratio for entrance exams. In the ExamBC task, the output value of each instance is used as a confidence score for tie-breaking multiple Y labels on series of pairs on a certain topic. Therefore, estimation of the output of the method is one of the most important factors for the correct answer ratio.

The MethodEX3 computes a confidence from three classifiers; AdaBoost with SMO, Logistic and Rotation Forest. First, the method obtains three output values from the classifiers. Then, it computes a weighted score by

$$Score = \frac{\alpha \times SMO + \beta \times Logistic + \gamma \times RotationForest}{3} \quad (1)$$

where $\alpha = 0.5$, $\beta = 1.0$ and $\gamma = 0.5$. These values are determined heuristically from the development data set.

3. EXPERIMENTS

In this section, we describe the results of our methods on the development data first. Next, we discuss the formal run results.

3.1 Development data

First, we evaluated our method with the development data set by using the leave-one-out cross validation. In this experiment, we focused on the effectiveness of each feature, such as word correspondence and HDAG. The methods in

[4]http://www.cs.waikato.ac.nz/ml/weka/

Table 1: The experimental result for BC task.

Features	Accuracy	MacroF1
WC_{base}	65.6	58.8
WC_{cont}	77.1	76.7
WC_{sum}	59.1	47.0
WC_{base}+WC_{cont}	78.1	77.8
WC_{base}+WC_{cont}+WC_{sum}	77.9	77.5
WC_{all}+HDAG	77.9	77.5
WC_{all}+OTHER	**79.5**	**79.3**
ALL	79.4	79.2
ALL without Ontologies	78.4	78.2

Table 2: The experimental result for ExamBC task.

Features	Accuracy	MacroF1
WC_{base}	58.6	43.8
WC_{cont}	65.9	64.6
WC_{sum}	58.8	43.6
WC_{base}+WC_{cont}	65.3	64.2
WC_{base}+WC_{cont}+WC_{sum}	65.4	64.6
WC_{all}+HDAG	**66.7**	**65.8**
WC_{all}+OTHER	65.5	64.4
ALL	66.5	65.5
ALL without Ontologies	64.1	63.3

the experiment were MethodBC1 for the BC task and MethodEX1 for the ExamBC task, respectively. In other words, the methods were based on SVMs.

Table 1 and Table 2 show the experimental results for the BC task and the ExamBC task, respectively. In the tables, WC denotes word correspondence features. WC_{base}, WC_{cont} and WC_{sum} denote the features using all word, content word and simplified word lists, respectively. "+" denotes the combination of features. WC_{all} denotes all word correspondence features, namely WC_{base}+WC_{cont}+WC_{sum}. HDAG and OTHER denote the features described in Section 2.1.2 and Section 2.1.3. "ALL without Ontologies" denotes the features that combined WC_{all} without two ontologies and HDAG+OTHER.

The WC_{all}+OTHER feature set for the BC task and the WC_{all}+HDAG feature set for the ExamBC task produced the best performance. The features about the content word list were effective for both the tasks. For the ExamBC task, the method with HDAG generated a slight positive effect on the both criteria. Although the method with HDAG correctly classified some instances that were incorrect by the method without HDAG, some instances were classified incorrectly by using HDAG, namely structure information. Figure 4 shows an example of the mistake. In this instance, some word correspondence features for t_2 to t_1 generated high values. As a result, the method without HDAG classified this instance correctly. However, the relations between the subject word and the main predicate verb, [Stockholm Olympic → Olympic Summer Games] and [Stockholm → city], mismatched. Therefore, the method with HDAG predicted the incorrect label for this instance.

The method with ontologies outperformed that without

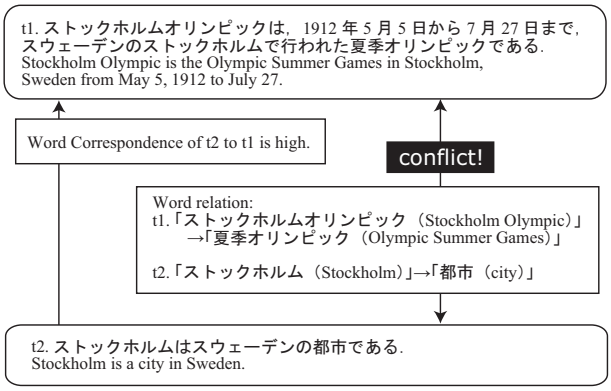

Figure 4: An incorrect instance.

Table 3: The results on the formal run.

Data	Method	HDAG	MacroF1
BC	MethodBC1	With	77.1
		Out	76.9
	MethodBC2	With	72.4
		Out	75.5
	MethodBC3	With	76.2
		Out	76.0
EX	MethodEX1	With	59.8
		Out	61.3
	MethodEX2	With	57.2
		Out	57.4
	MethodEX3	With	59.1
		Out	60.8

ontologies for both the tasks (79.2 vs. 78.2 on MacroF1 for the BC and 65.3 vs. 63.3 on MacroF1 for the ExamBC). This result shows the effectiveness of expansions using ontologies, such as synonyms of each word.

3.2 Formal run

Next, we discuss the formal run results. Table 3 shows the results. In the table, "With" and "Out" denote a method with HDAG and a method without HDAG, respectively. All submitted runs included HDAG information as the features, i.e., the method with "With" in the table are our formal run results for the BC and ExamBC tasks.

As in the case of the formal runs, incorporating HDAG features to our method did not lead to the improvement of the macro F1 measure. In some situations, such as the MethodBC2 and the MethodEx1, the HDAG feature decreased the macro F1 measure. One reason that the accuracy decreased is that the layer of the HDAG structure. Since we used the POS layer as the upper layer of the semantic layer, the similarity based on the HDAG was sensitive to structure information of sentences rather than semantic relations. Figure 5 shows an example of the problem. In this example, the HDAG generates a high similarity value because the POS layer is completely matched although the meanings of them are entirely-different. We need to consider the layers of the HDAG for computing more suitable similarity values.

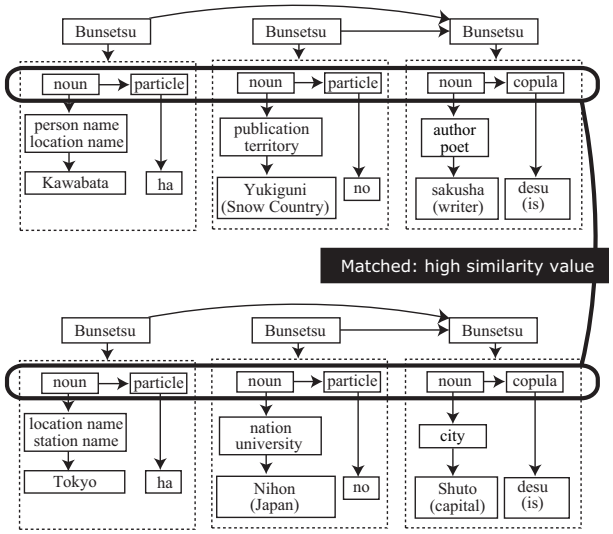

Figure 5: An example of the layer problem.

Table 4: The correct answer ratio on the formal run.

Method	MacroF1	CorrectAR
MethodEX1	**59.8**	36.1
MethodEX2	57.2	39.8
MethodEX3	59.1	**45.4**

Then, we compared the correct answer ratio for the ExamBC task. For the ExamBC, important evaluation criteria are not only the classification accuracy of textual entailment, namely Yes or No, but also the accuracy on the entrance exam. Table 4 shows the correct answer ratios of three methods. The mirco F1 values are the same as Table 3. The CorrectAR in the table denotes the correct answer ratio.

On the F1 measure, MethodEX1 (AdaBoost with SMO) produced the best performance in our methods. However, the MethodEX3 (weighted scoring) outperformed the two methods on the correct answer ratio even though the MacroF1 of MethodEX1 was slightly better than thet of MethodEx3. For the selection of the correct answer, the confidence value from each method is important. Therefore, the MethodEX3 based on a weighting function with several classifiers' outputs was suitable for the answer selection on the entrance exam task. Parameter tuning on the weighting approach and integration of other classifiers are our important future work.

4. CONCLUSIONS

This paper described a Japanese textual entailment recognition system, which is named KitAi. We used machine learning techniques, such as SVM, and a weighted scoring method for the BC and ExamBC tasks. We applied not only simple bag of words features but also semantic and grammatical information, namely ontologies and the HDAG structure.

The best accuracy rate for the BC task was 77.11 points on the macro F1 measure (10th among 42 methods). The best accuracy rate for the ExamBC task was 59.84 on the macro F1 measure (13th among 32 methods)[5]. All our methods outperformed the baseline system [8]. For the correct answer ratio on the ExamBC task, the weighted scoring method predicted the suitable confidence value even though the textual entailment accuracy was lower than other methods.

In the development data and formal run, there was no significant difference between the methods with HDAG and without HDAG. To improve the accuracy, we need to consider the layers in the HDAG. For the correct answer ratio on the ExamBC task, estimation of the confidence score from the method is one of the most important points. To improve the criterion, we need to add more suitable classifiers to the weighting method. In addition, to determine appropriate parameters, namely weights in Equation (1), is important future work.

5. REFERENCES

[1] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. Enhancing the japanese wordnet. In The 7th Workshop on Asian Language Resources, in conjunction with ACL-IJCNLP 2009, 2009.

[2] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. Goi-Taikei - A Japanese Lexicon. Iwanami Shoten, 1999.

[3] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning, 1998.

[4] R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[5] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10):1619–1630, 2006.

[6] K. Shimada, D. Hashimoto, and T. Endo. Sentiment sentence extraction using a hierarchical directed acyclic graph structure and a bootstrap approach. In Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation (PACLIC22), pages 341–349, 2008.

[7] J. Suzuki, Y. Sasaki, and E. Maeda. Hierarchical directed acyclic graph kernel. Systems and Computers in Japan, 37(10):58–68, 2006.

[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.

[5]In terms of the correct answer ratio, the best accuracy was 45.37 and 11th among 32 methods.