KitAi-VAL: Textual Entailment Recognition System for NTCIR-11 RITE-VAL

Ayaka Morimoto Kenta Kurashima Yo Tokunaga Kazutaka Shimada

Department of Artificial Intelligence
Kyushu Institute of Technology
{a_moimoto, k_kurashima,, y_tokunaga, shimada}@pluto.ai.kyutech.ac.jp

ABSTRACT

This paper describes Japanese textual entailment recognition systems for NTCIR-11 RITE-VAL. The tasks that we participated in are the system validation subtask and the fact validation subtask for Japanese. Our methods for the system validation are based on our previous method KitAi for RITE2. We add new features to the previous method. In addition, we construct a combined classifier for the unit-test, which is a sentence pair, t_1 and t_2, about a single linguistic phenomenon. For the fact validation task, we propose two approaches; search log based and summarization based methods. The search log based method generates a classifier using logs from Apache Solr. It does not contain any linguistic features for the classifier. The summarization based method generates t_1 from outputs of Apache Solr. It is a kind of multi-document summarization. We apply the generated t_1 to KitAi, namely a classifier for the binary class problem of textual entailment recognition. In formal runs, the best accuracy rates in the methods for the system validation and the fact validation tasks were 68.02 points and 57.98, respectively.

Team Name

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

Subtasks

System validation (SV) and fact validation (FV) tasks (Japanese)

Keywords

Correspondence, Edit Distance, Ontologies, Weighted Scoring, Search log, Summarization, Combination

1. INTRODUCTION

This paper describes Japanese textual entailment recognition systems for NTCIR-11 RITE-VAL (System validation (SV) and fact validation (FV) tasks) [4]. Our methods, KitAi-VAL[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 Japanese WordNet [1] and Nihongo-Goi-Taikei [3]. In other words, we apply a surface-based alignment process with the semantic information to our textual entailment recognition systems.

For the system validation, one of the main topics is the unit-test task. It is single linguistic phenomena in recognizing textual entailment. For the task, we focus on some pattern-based features and a combined method consisting of single classifiers to several linguistic phenomena in the unit-test. We utilize a weighted vote strategy to determine the final class for the unit-test.

In fact validation, t_1 is not given. Therefore, we need to estimate t_1 from textbooks. We regard it as a summarization task. We propose two approaches for the generation of t_1 from textbooks. The first approach is to extract one sentence with the highest confidence from candidate documents. The second approach is to generate a sentence from candidate documents on the basis of some rules. We also propose another method for the fact validation. It is based on search log information.

In the next section, we describe features and methods for the system validation. Next, we describe methods for the fact validation; a search log method and two summarization based methods. Then, we discuss our experimental results on formal run in Section 4. Finally, we conclude our methods in Section 5.

2. SYSTEM VALIDATION

2.1 Basic classifier

The basic classifier for RITE-VAL is the classifier used in RITE-2, namely KitAi. It is based on a machine learning approach. In this section, we explain the features of KitAi.

First, we describe features of the original KitAi. The feature set for the method consists of several linguistic information such as word correspondence. For the feature extraction process, we use JUMAN[2] as a morphological analyzer and KNP as a dependency parser[3].

The basic features in the 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. Furthermore, we extend each word by using two ontologies: the Japanese WordNet [1] and Nihongo-Goi-Taikei [3]. For the Japanese WordNet, we use the synonyms database, which is

[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

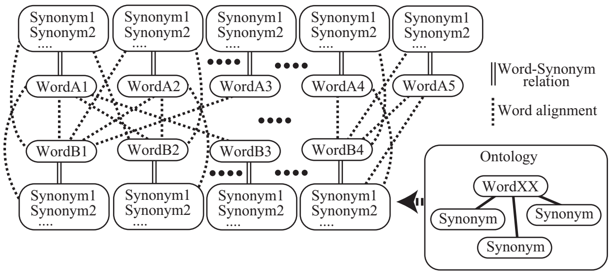

Figure 1: The word correspondence with ontologies.

created by synsets and manually annotated. For Nihongo-Goi-Taikei, we use words that belong to the same semantic class for each word. Figure 1 shows the process of calculation of word correspondence degrees. We compute the edit distance by using the DP matching.

We apply these features a machine learning approach, such as SVM. For more details, see our RITE-2 paper [8].

2.2 Added features

Next, we add some features to the original KitAi to improve the accuracy. After RITE-2, we analyzed errors. On the basis of the error analysis, we add five features [5]. They are based on linguistic expressions.

- The pattern “X_1 to yobu” exists in a sentence t_1 or t_2 and the pattern “X_1 toha ... koto wo iu” or “X_1 toha ... de aru” exists in the other sentece.
- The pattern “X_1 ha X_2 de ari X_3 ha X_4 de aru” exists in t_1 and the pattern “X_3 ha $X_1 X_2$ de aru” exists in t_2.
- The pattern “X_1 ha X_2 wo X_3 shita” exists in t_1 and the pattern “X_2 ha X_1 ga X_2 shita” exists in t_2.
- Location names exist in t_1 and the location names appear in t_2 by the specific order. For example, X_1 prefecture and X_2 city exist in t_1 and “X_2 located in $X1$” exists in t_2.
- A parallel noun phrase exists in t_1 and one of them appears in t_2.

In the pattern, each X is a word or a phrase. These are binary features $\{0, 1\}$.

The main topic of RITE-VAL is addition of unit-tests. In other words, it is a textual entailment recognition task focusing on single linguistic phenomena between each sentence pair. In RITE-VAL, 30 categories are defined as the linguistic phenomena. To solve the unit-test task, we add five features.

- The suffix expression “reru” or “rareru” appears in t_1 or t_2.
- Case-ga of t_1 (or t_2) matches Case-wo t_2 (or t_1).
- An expression about limitation, such as “nomi” appears in only t_2.
- A proper noun appears in only t_2.
- A numeral appears in only t_2.

These are also binary features $\{0, 1\}$.

Table 1: The experimental result for the RITE-2 unit test data.

Method	Accuracy
KitAi	35.67
Combined method	38.01

2.3 Classifier for unit test

For the unit tests, we propose a combined method for the classification of 30 categories. We focus on nine categories containing many instances in the training data. The nine categories are (1) case_alternation, (2) scrambling, (3) synonymy phrase, (4) modifier, (5) entailment phrase, (6) coreference, (7) clause, (8) relative_clause and (9) implicit_relation. We generate specific classifiers for each category. We integrate them with the improved KitAi, namely the classifier with features described in Section 2.2.

First, we generate specific classifiers. We apply all features in Section 2.1 and 2.2 to each specific classifier. Then, we select effective features from them experimentally. For example, the classifier for the case_alternation category selects some word correspondence features in Section 2.1 and the suffix expression feature in Section 2.2.

Next, we combine the nine specific classifiers and the improved KitAi. We adopt a weighted vote approach with a threshold to the combination. Each specific classifier has a threshold. If the output of a specific classifier is more than the threshold, the specific classifier can vote with a weight. Our combined method selects one output from weighted votes as the category of an input sentence pair. The threshold values are 0.5 for category (1), (3), (4), (5), (7), (8), 0.72 for (6), 0.75 for (2) and 0.85 for (9). The weights for each vote are 1.0 for (1), (3), (4), (5) and (7) and 1.5 for others. These values were determined experimentally.

We evaluated the combined method with the RITE-2 data. We compared it with the improved KitAi, namely a non-combined method. We use SVMs for the improved KitAi and C4.5 for the combined method as the machine learning method. The experimental result shows Table 1[4]. The combined method outperformed the non-combined method.

3. FACT VALIDATION

3.1 Document retrieval

First, we need to retrieve related documents or passages for an input, namely t_2, from textbooks. We use Apache Solr as the document retrieval task.

For the searching process, we need keywords as inputs for Apache Solr. We construct a word dictionary from textbooks and RITE2 datta automatically. First, we divide each sentence to words by using a morphological analysis JUMAN, and extract nouns from the divided words. Here we handle successive nouns as one word. In addition, we eliminate noise words by using some rules, such as words with one Kanji character. We obtained 51,165 keywords from this process.

We search related documents by using the keyword list for each t_2. First, we divide each sentence to words. Here

[4] This accuracy was the multi-class problem. The accuracy of the binary class task was 89.28.

Table 2: The score table of each category.

Category	Current	Prev/Next
Personal name	1.0	0.2
Location name	0.5	0.1
Sahen-noun	0.3	0.1
General noun	0.7	0.2
Compound noun	0.8	0.2

we classify the nouns into personal/location name, word in the keyword list and unknown word. We generate all combinations of the extracted words as queries, and search documents by them.

For the searching process, we have two strategies. The first strategy is to extract documents as a result in the case that the number of words in a query is maximum and the query obtains a result consists of at least one document (Search-1). The second strategy is to extract documents as a result for all combinations (Search-2).

3.2 Search log method

Next, we explain the search log method. It handles only search log information for the fact validation task. For this method, we use the Search-2 logs in Section 3.1 as the input. We extract 47 features from each search log. The outline of them is as follows:

Document information: it consists of the number of documents in each result, the minimum and maximum number of words in the retrieved document, the number of documents retrieved with n-queries and more than n-queries and so on.

Query information: it consists of the size of query words, the minimum and maximum number of words in the queries, the number of query words in the document title and so on.

Weight information: it consists of the minimum, maximum and average values of tfidf in the retrieved documents.

We apply these features to a machine learning method.

3.3 Summarization method

For the summarization approach, we apply two methods. For these methods, we use the Search-1 logs in Section 3.1 as the input.

The first method extracts the most important sentence from the retrieved documents. For the extraction process, we apply a weighting method about each sentence and the previous and next sentences. For the weighting, we identify the following information of each word by using a morphological analyzer; personal name, location name, sahen-noun, general noun, compound noun and others. Table 2 shows the score table. The "Current" and "Prev/Next" in the table denote the score of a current sentence and the scores of the previous and next sentences of the current sentence. These values are determined experimentally. Finally, we select one sentence with the highest value as t_1.

The second method is a combined approach of two phrases. The basic idea is to extract phrases which include many query words in a short range and predicates in t_2. First, we divide each sentence, which contains at least one query word, into phrases by using punctuation marks. For all extracted phrases, we compute a score as follows:

$$Score = PW \times Length \tag{1}$$

where $Length$ is the length of the phrase. The PW is a weighted value of a phrase and is computed as follows:

$$PW = \sum_{q \in P} W_q \tag{2}$$

where q are a query word in a phrase P. W_q is 4 for personal names, 3 for location names, 2 for words in textbooks and 1 for others. These values are determined experimentally. We extract the phrase with the highest score as a main phrase first. Then, we compute a similarity measure between the main phrase and other phrases. We select the phrase with the lowest similarity as an additional phrase. Finally we combine them as t_1[5].

Table 3: The experimental result for the system validation.

Method	Accuracy	Macro-F1
MethodSV1	68.02	62.02
MethodSV2	65.41	59.93
MethodSV3	33.14	32.12

4. FORMAL RUN

In this section, we describe our methods for the formal run and the results. For all methods, we use the open source software Weka[6] for each machine learning method.

4.1 Methods and results for system validation

The first method (MethodSV1) is a straightforward approach by SMO, which is a support vector classifier with John Platt's sequential minimal optimization algorithm [6]. The basic idea and features of MethodSV1 is described in Section 2.1 and 2.2.

The second method (MethodSV2) is also a simple extended version of RITE2. In the RITE2 formal run, a combined method with three classifiers generated the high accuracy rate in terms of the correct answer ratio [8, 9]. Therefore, we apply a similar approach to the second method. We combine SMO, Logistic and J48 in Weka as the single classifiers. Logistic is a multinomial logistic regression model with a ridge estimator [2]. J48 is C4.5 algorithm [7]. First, the method obtains three output values from the classifiers. Then, it computes a weighted score by

$$Score = \frac{\alpha \times SMO + \beta \times Logistic + \gamma \times J48}{3} \tag{3}$$

where $\alpha = 1.0$, $\beta = 1.8$ and $\gamma = 0.8$.

The third method (MethodSV3) is described in Section 2.3. Although it is a method for the unit-test, we use the third method for the system validation[7].

The result on the formal run is shown in Table 3. The accuracy and macro-F1 were not good as a whole. The reason

[5] The result often become an ungrammatical sentence.

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

[7] Unfortunately, the unit-test data was not provided on the formal run.

Table 4: The experimental result for the fact validation.

Method	Accuracy	Macro-F1	CorrectAR
MethodFV1	57.98	50.30	30.27
MethodFV2	57.20	55.91	19.02
MethodFV3	56.61	54.16	28.23

was that we focused on only single linguistic phenomena, namely the unit-test, in the system development for RITE-VAL.

4.2 Methods and results for fact validation

For the fact validation, we apply one search log method and two summarization methods. The numbers of retrieved documents in Section 3.1 were 1,920 documents (Search-1) and 517,039 documents (Search-2) for 514 sentences on the formal run, respectively.

The first method (MethodFV1) is the search log method with the Search-2 result. This method used only the search log. We apply the features mentioned in Section 3.2 to SMO on Weka.

The second method (MethodFV2) is the one sentence extraction method from the Search-1 result. We apply extracted t_1 to the basic classifier explained in Section 2.1.

The third method (MethodFV3) is based on the phrase combination method described in Section 3.3. The MethodFV3 is also a combination method of the search log method and the summarization method. First, we generated t_1 by using the the phrase combination method. Second, we generated a feature set from the t_1 for the basic classifier explained in Section 2.1. Next, we combined the feature set from the estimated t_1 and the feature set of the search log method. Then, we apply the combined feature set to three machine learning methods; J48, Logistic and MultilayerPerceptron (MP) on Weka. MultilayerPerceptron is a classifier that uses back-propagation to classify instances. Finally, we computed a score with weighted voting as follows:

$$Score = \frac{\alpha \times MP + \beta \times Logistic + \gamma \times J48}{3} \quad (4)$$

where $\alpha = 1.0$, $\beta = 1.8$ and $\gamma = 0.8$.

The result on the formal run is shown in Table 3. "CorrectAR" in the table denotes the correct answer ratio as the National Center Test[8]. For the accuracy and CorrectAR, the MethodFV1 (search log) produced the best performance. For the macro-F1, the MethodFV2 (one sentence extraction) was the best. However, the CorrectAR of the MethodFV2 was extremely low as compared with other two methods. On the other hand, the evaluation criteria of the MethodSV3 were better on average. This result might show the effectiveness of the combination of the log and summarization methods and the estimation of the confidence value from the scoring method (Eq. 4) for the final answer selection as the National Center Test.

5. CONCLUSIONS

This paper described a Japanese textual entailment recognition system, which is named KitAi-VAL. The tasks that we participated in were the system validation subtask and the fact validation subtask for Japanese. Our methods in NTCIR-11 RITE-VAL were based on our previous method KitAi for RITE2.

For the system validation, we added new features on the basis of an error analysis of RITE2. In addition, we proposed a combined method for the unit-test. For the experiment with the RITE2 unit-test data, the combined method outperformed an extended version of KitAi. We need to evaluate the combined method, MethodSV3, with another data set.

For the fact validation task, we proposed two types of approaches; search log based and summarization based methods. The search log based method generated a classifier using logs from Apache Solr. The summarization based methods, namely one sentence extraction and phrase combination, generated t_1 from outputs of Apache Solr. We applied the generated t_1 and t_2 pairs to KitAi. The search log method (MethodFV1) was the best in terms of the accuracy rate. The one sentence extraction method (MethodFV2) was not suitable in terms of the National Center Test because the correct answer ratio was the lowest in them. The MethodFV3, which is a combined method wiht the phrase combination and search log methods, was better on average.

[8] It is the average value of several subjects

6. 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. L. Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. Applied Statistic, 41(1):191–201, 1992.

[3] 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.

[4] S. Matsuyoshi, Y. Miyao, T. Shibata, C.-J. Lin, C.-W. Shih, Y. Watanabe, and T. Mitamura. Overview of the ntcir-11 recognizing inference in text and validation (rite-val) task. In Proceedings of the 11th NTCIR Conference, 2014.

[5] A. Morimoto and K. Shimada. Improving a textual entailment recognition system with error analysis (in japanese). In The 22nd IEICE Kyushu section Gakuseikaikouenkai, D-37, 2014.

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

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

[8] K. Shimada, Y. Seto, M. Omura, and K. Kurihara. Kitai: Textual entailment recognition system for ntcir-10 rite2. In Proceedings of the 10th NTCIR Conference, 2013.

[9] 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.