Predicate-argument Structure based Textual Entailment Recognition System of KYOTO Team for NTCIR9 RITE

Tomohide Shibata
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku,
Kyoto, 606-8501, Japan
shibata@i.kyoto-u.ac.jp

Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku,
Kyoto, 606-8501, Japan
kuro@i.kyoto-u.ac.jp

ABSTRACT

We participated in Japanese tasks of RITE in NTCIR9 (team id: "KYOTO"). Our proposed method regards predicate-argument structure as a basic unit of handling the meaning of text/hypothesis, and performs the matching between text and hypothesis. Our system first performs predicate-argument structure analysis to both a text and a hypothesis. Then, we perform the matching between text and hypothesis. In matching text and hypothesis, wide-coverage relations between words/phrases such as synonym and is-a are utilized, which are automatically acquired from a dictionary, Web corpus and Wikipedia.

Keywords

RTE, predicate-argument structure, lexical knowledge

Team Name

KYOTO

Subtasks/Languages

Japanese RITE BC, Japanese RITE MC, Japanese RITE EXAM, Japanese RITE RITE4QA

External Resources Used

Japanese Morphological Analyzer JUMAN, Japanese parser KNP, Japanese dictionary, Japanese Wikipedia

1. INTRODUCTION

RTE (Recognizing Textual Entailment) is the task of automatically recognizing textual inference. RTE is important in basic analysis such as parsing and anaphora resolution as well as applications such as Question Answering (QA) and Machine Translation (MT).

Several methods for RTE have been proposed so far. The precise matching between text and hypothesis cannot be achieved only with surface clues, such as the overlap ratio of characters/words between text and hypothesis. Therefore, our proposed method regards predicate-argument structure as a basic unit of handling the meaning of text/hypothesis, and performs the matching between text and hypothesis.

To perform the precise matching, wide coverage relations between words/phrases, such as synonym, is-a and antonym, are indispensable. To recognize the following entailment relation, the synonym between "*genshiryoku-hatsuden*" (atomic power generation) and "*genpatsu*" (the abbr. of "*genshiryoku-hatsuden*"), and the synonym between "*haisyutsu*" (emit) and "*dasu*" (emit) are required.

(1) **T:** *genshiryoku-hatsuden-wa* *nisanka-tanso-wo*
Atomic power generation-tm carbon dioxide-acc
haisyutsu-shinai energy-*da*
does not emit *energy*
(Atomic power generation is energy in which carbon dioxide is not be emitted.)

H: *genpatsu-wa* *nisanka-tanso-wo*
Atomic power generation-tm carbon dioxide-acc
dasa-nai
does not emit
(Atomic power generation does not emit carbon dioxide.)

Our proposed method acquires such relations from a dictionary and Wikipedia, and calculates distributional similarity using a Web corpus. Then, they are utilized when matching text and hypothesis.

The rest of this paper is organized as follows: Section 2 describes resources our system utilizes, and Section 3 presents predicate argument structure analysis. Section 4 proposes PA-matching method, and Section 5 proposes SVM-based Method and two-stage Method. Finally, Section 6 describes our experiments.

2. RESOURCES

This section describes resources utilized for the matching between text and hypothesis.

2.1 Automatic Acquisition of Relations between Words/Phrases

First, synonym, is-a, and antonym relations are automatically extracted from an ordinary dictionary and Wikipedia. Next, by assigning an ID to each synonymous group, synonymy database is constructed. Then, SynGraph data structure, we proposed earlier [5], is introduced to pack expressive divergence.

2.1.1 Synonym/is-a/antonym Extraction from an Ordinary Dictionary

We extract synonym, is-a, and antonym relations from definitions of an ordinary dictionary. The last word of the first definition sentence is usually the hypernym of an entry

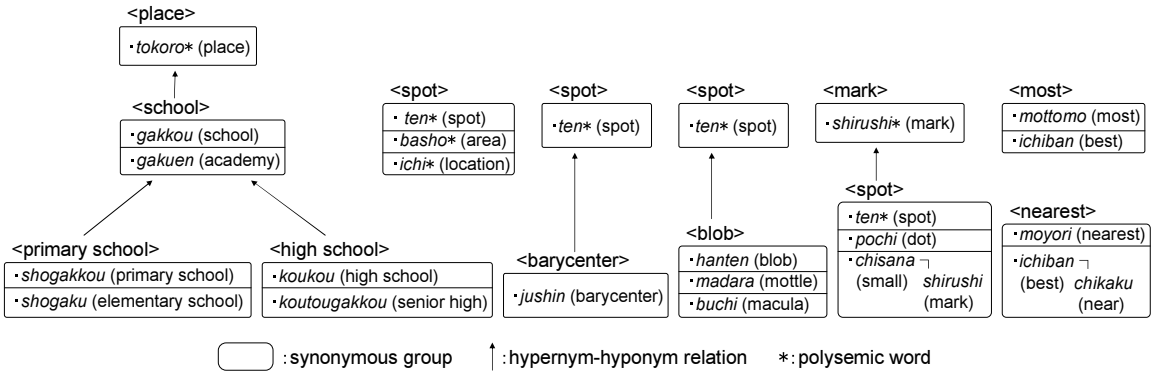

Figure 1: An example of synonymy database.

word. Some definition sentences in a Japanese dictionary are shown below (the left word of ":" is an entry word, the right sentence is a definition, and words in bold font is the extracted words):

yushoku (dinner) : *yugata* (evening) *no* (of) ***shokuji* (meal)**.
jushin (barycenter) : *omosa* (weight) *ga* (is) *tsuriatte* (balance) *tyushin* (center) *tonaru* (become) ***ten* (spot)**.

For example, the last word *shokuji* (meal) can be extracted as the hypernym of *yushoku* (dinner). In some cases, however, a word other than the last word can be a hypernym or synonym. These cases can be detected by sentence-final patterns as follows (the underlined expressions represent the patterns):

Hypernyms
dosei (Saturn) : ***wakusei* (planet)** *no* (of) *hitotsu* (one).
tobi (kite) : ***taka* (hawk)** *no* (of) *issyu* (kind).

Synonyms / Synonymous Phrases
ice : ***ice cream*** *no* (of) *ryaku* (abbreviation).
mottomo (most) : ***ichiban* (best)**. (∗ one word definition)
moyori (nearest) : ***ichiban* (best)** ***chikai* (near)** *tokoro* (place)[1]. (∗ less than three phrases)

Antonyms are extracted using a symbol representing an antonym in a definition. For example, "*atsui*" (hot) is extracted as the antonym of "*samui*" (cold).

2.1.2 Synonym/is-a Extraction from Wikipedia

By applying the same method as the one introduced in the previous section to Wikipedia articles, synonym/is-a relations are extracted from Wikipedia. For example, "*tansaki*" (probe) can be extracted as the hypernym of Genesis.

2.1.3 Synonymy Database Construction

With the extracted binomial relations, a synonymy database can be constructed. Here, polysemic words should be treated carefully[2]. When the relations A=B and B=C are extracted, and B is not polysemic, they can be merged into A=B=C.

[1]If the last word of a sentence is a highly general term such as *koto* (thing) and *tokoro* (place), it is removed from the synonymous expression.
[2]If a word has two or more definition items in the dictionary, the word can be regarded as polysemic.

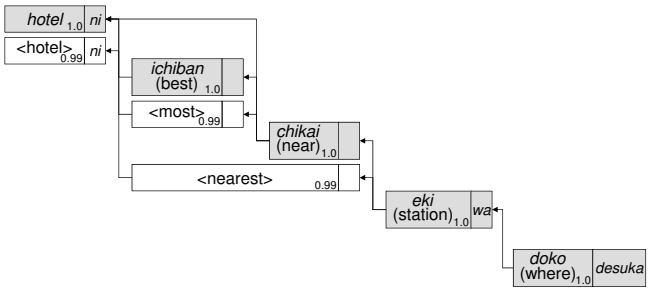

Figure 2: An example of SynGraph.

However, if B is polysemic, the synonym relations are not merged through a polysemic word. In the same way, as for is-a relations, A → B and B → C, and A → B and C → B are not merged if B is polysemic. By merging binomial synonym relations with the exception of polysemic words, synonymous groups are constructed first. They are given IDs, hereafter called SYNID. Then, is-a and antonym relations are established between synonymous groups. We call this resulting data as synonymy database. Figure 1 shows examples of synonymous groups in the synonymy database. SYNID is denoted by using English gloss word, surrounded by " ⟨ ⟩ ".

2.1.4 SynGraph Data Structure

SynGraph data structure is an acyclic directed graph, and the basis of SynGraph is the dependency structure of an original sentence. In the dependency structure, each node consists of one content word and zero or more function words, which is called a *basic node*. If the content word of a basic node belongs to a synonymous group, a new node with the SYNID is attached to it, and it is called a *SYN node*. For example, in Figure 2, the shaded nodes are basic nodes and the other nodes are SYN nodes.

Then, if the expression conjoining two or more nodes corresponds to one synonymous group, a SYN node is added there. In Figure 2, ⟨nearest⟩ is such a SYN node. Furthermore, if one SYN node has a hyper synonymous group in the synonymy database, the SYN node with the hyper SYNID is also added.

In this SynGraph data structure, each node has a score, NS (Node Score), which reflects how much the expression of the node is shifted from the original expression.

By converting both of a text and a hypothesis to SynGraph data structure, the synonym/is-a relation between words/phrases in a text and a hypothesis can be effectively

Table 1: Examples of contexts of the word "*haishi*" (abolish).

context	frequency
seido-wo (system-acc)	4,348
jorei-wo (regulation-acc)	2,968
jigyou-wo (business-acc)	1,597
...	
gyousha-ga (trader-nom)	24
...	

handled.

2.2 Distributional Similarity between Verbs

Although synonym/is-a relation between verbs can be acquired from a dictionary in the way introduced in Section 2.1.1, some near-synonymous relations such as "*haishi*" (abolish) and "*chuushi*" (stop) cannot be acquired. Therefore, distributional similarity between verbs is calculated using a Web corpus, and the pair whose distributional similarity is high is utilized when matching predicates in text and hypothesis. First, a context for each verb is extracted from a corpus. Then, distributional similarity between verbs is calculated.

2.2.1 Context Extraction

A context relation is defined as a tuple (v, r, n) where a noun n modifies a verb v with a case marker r. The tuple (r, n') is defined as a context of the verb v.

Context relations are extracted from a corpus, and then each verb is represented by a feature vector of the contexts. Table 1 shows examples of contexts of the verb "*haishi*" (abolish).

2.2.2 Distributional Similarity Calculation

Curran decomposes the distributional similarity calculation into *weight* and *measure* function [2]. The *weight* function transforms the raw counts for each feature instance into more comparable values. The *measure* function calculates the similarity between the two weighted vectors.

Several weight/measure functions have been utilized thus far. In this research, the most suitable combination of weight and measure functions was determined using the test evaluation set [1], in which the system decides whether or not a noun pair is similar. In this research, the following weight function is adopted:

$$weight = \begin{cases} 1(\mathsf{MI} > 0) \\ 0(otherwise) \end{cases} \quad (1)$$

where $\mathsf{MI} = \log \frac{P(u,f)}{P(u)P(f)}$. ($u$ and f represent a unit and feature, respectively.)

The following measure function is adopted:

$$measure = \frac{1}{2}(\mathsf{JACCARD} + \mathsf{SIMPSON}), \quad (2)$$

where

$$\mathsf{JACCARD} = \frac{|(u_1, *) \cap (u_2, *)|}{|(u_1, *) \cup (u_2, *)|} \quad (3)$$

$$\mathsf{SIMPSON} = \frac{|(u_1, *) \cap (u_2, *)|}{min(|(u_1, *)|, |(u_2, *)|)}. \quad (4)$$

The JACCARD-SIMPSON measure, which is the average of JACCARD and SIMPSON, is utilized to alleviate the problem arising when the numbers of features are totally different.

3. PREDICATE ARGUMENT STRUCTURE ANALYSIS

In both a text and a hypothesis, we perform morphological analysis using the Japanese Morphological Analyzer JUMAN[3] and syntactic/case analysis using the Japanese parser KNP[4][4]. Then, they are converted to SynGraph data structure. Based on the syntactic/case analysis, a text and a hypothesis are divided into predicate-argument structures.

3.1 Predicate Argument Structure

An example of predicate-argument structure is shown in Figure 3. Each predicate-argument structure consists of a predicate and zero or more arguments. For example, predicate-argument structure (1-1) consists of the predicate "*tanoshimu*" (enjoy) and the two arguments: case component "*ga*" and *de*".

Predicate-argument structure is able to have another predicate-argument structure that has the same meaning but has the different case structure. For example, predicate-argument structure (2-1) is an original one, and predicate-argument structure (2-2) is another predicate-argument structure that has the same meaning as (2-1). (In this case, while (2-1) has a passive form, (2-2) has an active form.)

Basically, a verb, adjective, and noun+copula are regarded as a predicate, and case components whose case marker is *ga* (nom), *wo* (acc), and *ni* (dat), *kara* (abl), *to* (cmi/quo), *he* (all), *made* (del), *yori* (cmp) are regarded as an argument.

KNP makes a case analysis using case frames, which are automatically acquired from a large Web corpus [3]. In the case frames, the case alignment of two case frames, such as active and passive, is performed. For example, the *ga* case of the case frame "*hiraka-reru*" (the passive voice of "*hiraku*" (hold)) and the *wo* of the case frame "*hiraku*" (hold) is aligned. By using this alignment, predicate-argument structure as (2-2) in Figure 3 can be generated.

In addition, the followings are regarded as a predicate-argument structure:

- deverbative noun

  (3) David Kelly *shi-no* *jisatsu-wa*
  Mr. David Kelly-gen suicide-tm
  eikoku-syakai-wo *yusaburi-tsuzukete-iru*
  British society-acc has shaken
  (The suicide of Mr. David Kelly has shaken the British society.)

  The deverbative noun "*jisatsu* (suicide)" is regarded as a predicate, and the following predicate-argument structure is generated.

jisatsu (suicide)	
〈*ga*〉	David Kelly *shi*(Mr)

- apposition

  (4) *motomoto* St. Valentine's Day-*wa*
  originally St. Valentine's Day-tm
  san-seiki-no Roma-*no* *shisai*
  third century-gen Roma-gen priest

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

(2) Tokyo-*to*-*Nishi*-*Tama*-*chiku*-*de*-*wa* *kakuchi*-*de* *kisetsu*-*wo* *tanoshimu* *ibento*-*ga* *hiraka*-*reru*
Tokyo-Metropolis-West-Tama-area-loc-tm each place-loc season-acc enjoy event-nom be hold

(In Tokyo West Tama area, the event, where people enjoy the season every place, is hold.)

⇓

1-1

tanoshimu (enjoy)	
[syn] ⟨enjoy⟩	
⟨*wo*⟩	*kisetsu* (season) [syn] ⟨season⟩
⟨*de*⟩	*kakuchi* (each place) [is-a] ⟨place⟩

2-1

hiraka-*reru* (be held)	
⟨*ga*⟩	*ibento* (event)
⟨*de*⟩	*chiku* (area) ⟨mod⟩ Tokyo-Metropolis-West-Tama

2-2

hiraku (hold)	
⟨*wo*⟩	*ibento* (event)
⟨*de*⟩	*chiku* (area) ⟨mod⟩ Tokyo-Metropolis-West-Tama

Figure 3: An example of predicate-argument structure. (In the example sentence, the underlined phrases represent a predicate.)

St. Valentine-*no* *densetsu*-*ni* *yurai*-*suru*
St. Valentine-gen legend-dat originate

(St. Valentine's Day originates in the legend of St. Valentine, the Roman priest in the third century.)

By KNP, "*shisai*" (priest) and "St. Valentine" are recognized as an apposition relation, and the following predicate-argument structure is generated.

shisai (priest)	
⟨*ga*⟩	St. Valentine

The representation for both predicate and argument is handled by a surface form. If a word has a synonym, the attribute [syn] whose value is its SYNID is added. For example, the word "*kisetsu* (season)" in the *wo* case component of predicate-argument structure (1-1) has the attribute [syn] whose value is ⟨season⟩. Similarly, if a word has a hypernym, the attribute [is-a] whose value is its SYNID is added.

If a verb has negation expression, the negation flag is attached to the verb. For example, the verb "*kaka*-*nai*" (don't write) has the negation flag.

As for verbs which represent affirmation/negation for the depending clause, no predicate-argument structure is generated for the verb. Verbs which represent affirmation/negation are manually prepared as follows:

- affirmation
  jijitsu (fact), *hontou* (true), *shinjitsu* (true), *makoto* (true)
- negation
  uso (lie), *ayamari* (mistake), *kangaechigai* (misunderstanding), *machigai* (mistake), *itsuwari* (deceit), *soragoto* (falsehood), *tsukurigoto* (fiction), *kyogi* (falsehood), *omoichigai* (misapprehension), *kanchigai* (misunderstanding), *sakkaku* (have an illusion), *gokai* (misconception)

As for verbs representing negation, the negation flag of the verb in the depending clause is reversed. In the following example, since "*kanchigai*" (misunderstanding) corresponds to a verb representing negation, the negation flag is attached to the verb "*korobu*" (fall down).

(5) *kare*-*ga* *koronda*-*to* *kanchigai*-*shita*
he-nom fell down misunderstood

((I) misunderstood that he fell down.)

korobu (fall down) negation	
⟨*ga*⟩	*kare* (he)

3.2 Numerical Expression

Numerical expressions are handled in a different way. A predicate is represented by a tuple: number, counter, and class, and a class corresponds to *exact*, *over*, or *less*. An argument takes only the *ga* case component.

(6) *uchuu*-*no* *nenrei*-*wa* *hyakuoku*-*sai*-*da*
space-gen age-tm 10 billion years

(The age of space is 10 billion.)

In the above sentence, the following predicate-argument structure is generated.

number: 10,000,000,000 counter: *sai* (years) class: exact	
⟨*ga*⟩	*nenrei* (age) ⟨*no*⟩ *uchuu* (space)

(7) *uchuu*-*no* *nenrei*-*wa* *hyakusanjuuoku*-*sai*-*ijo*-*da*
space-gen age-tm 13 billion years or more

(The age of space is 13 billion or more.)

In the above sentence, the following predicate-argument structure is generated.

number: 13,000,000,000 counter: *sai* (years) class: over	
⟨*ga*⟩	*nenrei* (age) ⟨*no*⟩ *uchuu* (space)

As for predicates that take numerical expressions as an argument, this is treated in the same way as the above example.

(8) *nyuuen-sha-suu-ga* *sanokunin-wo*
number of visitors-nom three hundred million-acc
koeta
exceed
(The number of visitors has exceeded three hundred million.)

In the above sentence, the following predicate-argument structure is generated.

number: 3,000,000,000	
counter: *nin* (person)	
class: over	
⟨*ga*⟩	*nyuuen-sha* (visitor)

Numerical expressions are made some normalizations as follows:

- 2 *man* (ten thousand) 5000 ⇒ 25000
- 5 *wari* (ten percent) ⇒ 50%.

4. PA-MATCHING METHOD

Based on predicate-argument structures of **T** and **H**, the entailment judgement is performed by matching **T** and **H** considering a predicate-argument structure as a basic unit. If all the predicate-argument structures in an **H** are matched to a predicate-argument structures in a **T**, **H** is judged to be entailed from **T**.

The entailment of predicate-argument structures is defined as follows. First, the predicate-argument structure in **H** is totally the same as the predicate-argument structure in **T**, i.e. the predicate and all the arguments in **H** are matched to those in **T**, where this match includes the correspondence of surface form, the correspondence of SYNID (which means synonymous relation), and the distributional similarity between predicates is greater than a threshold[5].

When arguments or predicates in **H** have is-a relation compared to those in **T**, the entailment relation is identified. The is-a relation of predicates/arguments is defined as follows:

is-a relation of predicates

- is-a relation of predicates:
  hirune (nap) → *neru* (sleep)
- lack of argument:
  kinou (yesterday) *umareta* (be born) → *umareta* (be born)

is-a relation of arguments

- is-a relation of nouns:
  katsuo (bonito) → *sakana* (fish),
  inko (parakeet) → *tori* (bird)
- lack of modification expression:
  jidousha-koujou (car factory) → *koujou* (factory)

If all the arguments are identical and the negation flag in a predicate is not identical, this is judged as "C" (Contradiction).

Table 2: Entailment relation between numerical expressions.

	T	**H**	magnitude relation between $num_{\mathbf{T}}$ and $num_{\mathbf{H}}$	judgement
(1)	exact	exact	$num_{\mathbf{T}} = num_{\mathbf{H}}$	Y
(2)			$num_{\mathbf{T}} \neq num_{\mathbf{H}}$	C
(3)	exact	over	$num_{\mathbf{T}} \geqq num_{\mathbf{H}}$	Y
(4)			$num_{\mathbf{T}} < num_{\mathbf{H}}$	C
(5)	exact	less	$num_{\mathbf{T}} \leqq num_{\mathbf{H}}$	Y
(6)			$num_{\mathbf{T}} > num_{\mathbf{H}}$	C
(7)	over	exact	–	N
(8)	over	over	$num_{\mathbf{T}} \geqq num_{\mathbf{H}}$	Y
(9)			$num_{\mathbf{T}} < num_{\mathbf{H}}$	N
(10)	over	less	$num_{\mathbf{T}} > num_{\mathbf{H}}$	C
(11)			$num_{\mathbf{T}} \leqq num_{\mathbf{H}}$	N
(12)	less	exact	–	N
(13)	less	over	$num_{\mathbf{T}} < num_{\mathbf{H}}$	C
(14)			$num_{\mathbf{T}} \geqq num_{\mathbf{H}}$	N
(15)	less	less	$num_{\mathbf{T}} \leqq num_{\mathbf{H}}$	Y
(16)			$num_{\mathbf{T}} > num_{\mathbf{H}}$	N

Entailment relation between numerical expressions.

Entailment relation between numerical expressions is handled in a different way. The entailment relation is defined as shown in Table 2, which is classified in terms of the class of number (*exact*, *over*, *less*) in **T** and **H**, and the magnitude relation between number of **T** ($num_{\mathbf{T}}$) and number of h ($num_{\mathbf{H}}$).

For example, when the sentence (6) is a text and the sentence (7) is a hypothesis, the relation is judged as "C" (corresponds to (4) in Table 2).

5. SVM-BASED METHOD AND TWO-STAGE METHOD

Although the method of entailment judgement introduced in the previous section is aimed at precisely matching between text and hypothesis, it is often the case that precise matching cannot be achieved due to the gap of structure, parsing error, the shortage of lexical knowledge and world knowledge, and others. To consider relatively shallow clues such as the overlap ratio of characters and morphemes, we take a machine learning approach where these clues as well as the result of PA-matching method are considered as a feature. SVM (Support Vector Machines) is adopted as a machine learning method, and the following features are considered:

- the overlap ratio of morphemes between **T** and **H**
- the overlap ratio of characters between **T** and **H**
  - 1-gram, 2-gram, 3-gram, 4-gram
- the result of PA-matching method
- there is a correspondence of predicate
- there is an inconsistent of negation flag in a predicate of **T** and **H**

In MC task, the SVM handles the 3-class classification problem (Y,C,N), and the one-vs-rest method is adopted. In the classification step, the trained SVM model is applied to the pair of **T** and **H** and the pair of **H** and **T**, and then the result is classified into five class (B, F, R, C, I) based on the SVM results.

[5]In this paper, the threshold is set to be 0.2.

The two-stage method is also adopted: since we believe the precision of the PA-matching method is high, first, the PA-matching method is applied, then if "Y" for BC task or "Y"/"C" for MC task is obtained, the result is adopted; otherwise, the SVM-based model is applied.

6. EXPERIMENTS

We participated in Japanese BC, MC, EXAM, and RITE4QA subtasks of RITE in NTCIR9 [6]. While the development set was prepared for BC, MC, and EXAM, it was not prepared for RITE4QA, and thus our method for BC was applied to RITE4QA.

6.1 Settings

For the acquisition of relations between words/phrases described in Section 2.1, REIKAI-SHOGAKU dictionary (a dictionary for children), which consists of about 30,000 entries, and Japanese Wikipedia were utilized. In the distributional similarity calculation between verbs described in Section 2.2, approximately 100 million Japanese Web pages were used.

For the implementation of SVM, svm_light[6] was adopted, and the linear kernel was used, where the default parameters were chosen. For the development set, the methods using SVM were evaluated on the 5 cross validation fold, and for the test set, SVM models were trained using all the development set data, and were applied to the test set.

We submitted the following three methods with RUNID (SUBTASK = {BC, MC, EXAM, RITE4QA}):

1. PA-matching method (introduced in Section 4)
   - RITE1-KYOTO-JA-SUBTASK-01.txt
2. SVM-based method (introduced in Section 5)
   - RITE1-KYOTO-JA-SUBTASK-02.txt
3. Two-stage method (introduced in Section 5)
   - RITE1-KYOTO-JA-SUBTASK-03.txt

In EXAM and RITE4QA, the confidence score for each text pair was required. In PA-matching method, it is 1.0 in the case of exact matching; otherwise 0.8. In SVM-based method and two-stage method, it is obtained by transforming SVM score x with the sigmoid function as follows:

$$|\frac{1}{1+e^{-x}} - 0.5| \times 2. \quad (5)$$

6.2 Result and Discussion

Table 3 shows an accuracy of BC dev, BC test, MC dev, MC test, EXAM dev, EXAM test, and RITE4QA. Table 4, 5, 6, 7, 8, 9, 10 shows a confusion matrix of BC dev, BC test, MC dev, and MC test, EXAM dev, EXAM test, RITE4QA, respectively.

In BC, while PA-matching method performed best in the development set, SVM-based method and two-stage method performed best in the test set. In MC, SVM-based method and two-stage method performed better than PA-matching method both in development set and test set. In Exam, SVM-based method and two-stage method performed best in both development and test set. In RITE4QA, PA-matching method performed best.

[6] svmlight.joachims.org/.

Table 4: Confusion matrix. (BC dev)

PA-matching Method
Accuracy: 0.550 (275 / 500)

		correct		
		Y	N	all
system	Y	36	11	47
	N	214	239	453
	all	250	250	500

SVM-based Method
Accuracy: 0.536 (268 / 500)

		correct		
		Y	N	all
system	Y	101	83	184
	N	149	167	316
	all	250	250	500

Two-stage Method
Accuracy: 0.536 (268 / 500)

		correct		
		Y	N	all
system	Y	101	83	184
	N	149	167	316
	all	250	250	500

Table 5: Confusion matrix. (BC test)

PA-matching Method
Accuracy: 0.492 (246 / 500)

		correct		
		Y	N	all
system	Y	13	17	30
	N	237	233	470
	all	250	250	500

SVM-based Method
Accuracy: 0.516 (258 / 500)

		correct		
		Y	N	all
system	Y	52	44	96
	N	198	206	404
	all	250	250	500

Two-stage Method
Accuracy: 0.516 (258 / 500)

		correct		
		Y	N	all
system	Y	52	44	96
	N	198	206	404
	all	250	250	500

Table 11 shows the result of one leave out experiments. For PA-matching method, the method that does not use SynGraph (which means this method does not utilize synonyms and is-a relations) and the method that does not use distributional similarity were tested. While for the development set in BC and MC, these two resources were effective, for the others, they were not effective.

The following example can be correctly judged as "Y" by using the synonymous relation "*hiraki*" (difference) and "*sa*" (difference), which was acquired from a dictionary.

(9) **T:** *chuugakusei-no* *gakuryoku-ni*
junior high school student-gen achievement-dat
ookina *hiraki-ga* *aru*
great difference-nom there is

(There is a great difference in the achievement of junior high school students.)

H: *chuugakusei-no* *gakuryoku-ni*
junior high school student-gen achievement-dat

Table 3: Experimental results. (The numbers represent an accuracy.)

	BC dev	BC test	MC dev	MC test	EXAM dev	EXAM test	RITE4QA
PA-matching Method	**0.550**	0.492	0.216	0.214	0.593	0.593	**0.889**
SVM-based Method	0.536	**0.516**	**0.498**	0.480	**0.655**	**0.656**	0.684
Two-stage Method	0.536	**0.516**	0.495	**0.484**	**0.655**	**0.656**	0.684

Table 11: Result of one leave out experiments. (The numbers represent an accuracy.)

	BC dev	BC test	MC dev	MC test	EXAM dev	EXAM test	RITE4QA
PA-matching Method	0.550	0.492	0.216	0.214	0.593	0.593	0.889
w/o SynGraph	0.542	0.496	0.214	0.216	0.589	0.593	0.890
w/o distributional similarity	0.532	0.494	0.205	0.207	0.593	0.588	0.889
SVM-based Method	0.536	0.516	0.498	0.480	0.655	0.656	0.684
w/o PA-matching Method feature	0.512	0.570	0.493	0.482	0.651	0.665	0.362

Table 6: Confusion matrix. (MC dev)

PA-matching Method
Accuracy: 0.216 (95 / 440)

		correct					
		F	R	B	C	I	all
system	F	6	2	2	5	0	15
	R	0	5	3	0	0	8
	B	1	1	3	0	1	6
	C	0	1	0	2	0	3
	I	103	101	67	58	79	408
	all	110	110	75	65	80	440

SVM-based Method
Accuracy: 0.498 (219 / 440)

		correct					
		F	R	B	C	I	all
system	F	51	0	3	19	18	91
	R	2	56	4	6	3	71
	B	19	25	58	12	5	119
	C	0	0	0	0	0	0
	I	38	29	10	28	54	159
	all	110	110	75	65	80	440

Two-stage Method
Accuracy: 0.495 (218 / 440)

		correct					
		F	R	B	C	I	all
system	F	51	2	5	20	17	95
	R	2	55	6	5	2	72
	B	20	22	55	9	7	113
	C	0	1	0	2	0	3
	I	37	30	7	29	54	157
	all	110	110	75	65	80	440

Table 7: Confusion matrix. (MC test)

PA-matching Method
Accuracy: 0.214 (94 / 440)

		correct					
		F	R	B	C	I	all
system	F	3	0	0	2	0	5
	R	1	7	1	0	1	10
	B	0	1	3	0	0	4
	C	0	1	0	2	0	3
	I	106	101	71	61	79	418
	all	110	110	75	65	80	440

SVM-based Method
Accuracy: 0.480 (211 / 440)

		correct					
		F	R	B	C	I	all
system	F	52	0	5	14	16	87
	R	1	54	5	5	1	66
	B	28	23	59	20	17	147
	C	0	0	0	0	0	0
	I	29	33	6	26	46	140
	all	110	110	75	65	80	440

Two-stage Method
Accuracy: 0.484 (213 / 440)

		correct					
		F	R	B	C	I	all
system	F	54	0	5	15	16	90
	R	2	54	6	5	2	69
	B	25	23	58	18	17	141
	C	0	1	0	2	0	3
	I	29	32	6	25	45	137
	all	110	110	75	65	80	440

ookina *sa*-*ga* *aru*
great difference-nom there is

(There is a great difference in the achievement of junior high school students.)

The following example can be correctly judged as "Y" by using the fact that the distributional similarity between "*kaikan*" (open) and open is high.

(10) **T:** Kanazawa-*shi-no* *kankou-meisyo*
Kanazawa City-gen tourist spot

Kenroku-en-ni *chikai* *shi-chuushin-bu-ni*
Kenroku-en-dat near the center of the city-dat

Kanazawa-21-*seiki-bijyutsukan-ga* *kaikan*-*shita*
Kanazawa 21st century museum-nom was open

(Kanazawa 21st century museum was open in the center of the city, near Kenroku-en, which is a tourist spot in Kanazawa City.)

H: Kanazawa-21-*seiki-bijyutsukan-ga* *open*-*shita*
Kanazawa 21st century museum-nom was open

(Kanazawa 21st century museum was open.)

For SVM-based method, the method that does not use PA-matching result feature was tested. While for the development set in BC, MC and EXAM, and RITE4QA, it is effective, for the others, it was not effective. This is because the accuracy of PA-matching method is not so good.

The following is an example in which PA-matching method incorrectly judged as "Y". Although with the is-a relation between "6-*kai-ijyou-no shinchiku-manshon*" and "*shinchiku-manshon*", the system judged "Y", "6-*kai-ijyou-no*" works as a condition for "*shinchiku-manshon*", and thus the correct answer was "N". To recognize such conditions is our future work.

(11) **T:** *kokudo-koutsuushou-wa* *koureika-ni*
The MLIT-tm graying-dat

Table 8: Confusion matrix. (EXAM dev)

PA-matching Method
Accuracy: 0.593 (296 / 499)

		correct		
		Y	N	all
system	Y	6	5	11
	N	198	290	488
	all	204	295	499

SVM-based Method
Accuracy: 0.655 (327 / 499)

		correct		
		Y	N	all
system	Y	90	58	148
	N	114	237	351
	all	148	351	499

Two-stage Method
Accuracy: 0.655 (327 / 499)

		correct		
		Y	N	all
system	Y	90	58	148
	N	114	237	351
	all	148	351	499

Table 9: Confusion matrix. (EXAM test)

PA-matching Method
Accuracy: 0.593 (262 / 442)

		correct		
		Y	N	all
system	Y	3	2	5
	N	178	259	437
	all	181	261	442

SVM-based Method
Accuracy: 0.656 (290 / 442)

		correct		
		Y	N	all
system	Y	75	46	121
	N	106	215	321
	all	181	261	442

Two-stage Method
Accuracy: 0.656 (290 / 442)

		correct		
		Y	N	all
system	Y	75	46	121
	N	106	215	321
	all	181	261	442

taiou-suru-tameni 98-*nen-ni*
cope with 98-dat

kouei-jyuutaku-nado-seibi-kijyun-de
standards for public housing-ins

6-*kai-ijyou-no* *shinchiku-manshon-ni*
six-and-over-story-gen newly built condominium-dat

elevator-*no* *secchi-wo* *gimuzuketa*
elevator-gen establishment-acc mandated

(The MLIT mandated the establishment of elevator in six-and-over-story newly built condominium by standards for public housing in 98 to cope with graying.)

H: *kokudo-koutsuushou-wa* *koureika-ni*
The MLIT-tm graying-dat

taiou-suru-tameni *shinchiku-manshon-ni*
cope with newly built condominium-dat

elevator-*no* *secchi-wo* *gimuzuketa*
elevator-gen establishment-acc mandated

Table 10: Confusion matrix. (RITE4QA)

PA-matching Method
Accuracy: 0.889 (857 / 964)

		correct		
		Y	N	all
system	Y	3	4	7
	N	103	854	957
	all	106	858	964

SVM-based Method
Accuracy: 0.684 (659 / 964)

		correct		
		Y	N	all
system	Y	28	227	255
	N	78	631	709
	all	106	858	964

Two-stage Method
Accuracy: 0.684 (659 / 964)

		correct		
		Y	N	all
system	Y	28	227	255
	N	78	631	709
	all	106	858	964

(The MLIT mandated the establishment of elevator in newly built condominium to cope with graying.)

7. CONCLUSION

This paper described our RTE system (team id: "KYOTO"). Our system regarded predicate-argument structure as a basic unit of handling the meaning of text and hypothesis, and performed the matching between text and hypothesis. A wide-coverage relations between words/phrases such as synonym and is-a were automatically acquired from a dictionary, Web corpus and Wikipedia, and were utilized when matching text and hypothesis.

Our future work includes the further acquisition of linguistic knowledge and the flexible matching between structures of text and hypothesis.

8. REFERENCES

[1] A. Aizawa. On calculating word similarity using large text corpora. *IPSJ Jounal*, 49(3):1426–1436, 2008. (in Japanese).

[2] J. R. Curran. *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh. College of Science, 2004.

[3] D. Kawahara and S. Kurohashi. Case frame compilation from the web using high-performance computing. In *Proceedings of LREC-06*, 2006.

[4] D. Kawahara and S. Kurohashi. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the HLT-NAACL2006*, pages 176–183, 2006.

[5] T. Shibata, M. Odani, J. Harashima, T. Oonishi, and S. Kurohashi. SYNGRAPH: A flexible matching method based on synonymous expression extraction from an ordinary dictionary and a web corpus. In *Proceedings of Third International Joint Conference on Natural Language Processing (IJCNLP2008)*, pages 787–792, 2008.

[6] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In *NTCIR-9 Proceedings*, 2011.