Predicate-argument Structure based Textual Entailment Recognition System of KYOTO Team for NTCIR-10 RITE-2

Tomohide Shibata, Sadao Kurohashi, Shotaro Kohama, Akihiro Yamamoto
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku,
Kyoto, 606-8501, Japan
{shibata,kuro,akihiro}@i.kyoto-u.ac.jp, khmshtr28@iip.ist.i.kyoto-u.ac.jp

ABSTRACT

We participated in Japanese tasks of RITE-2 in NTCIR-10 (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 BC, Japanese MC, Japanese EXAM, Japanese Search-Task

External Resources Used

Japanese Morphological Analyzer JUMAN, Japanese parser KNP, Japanese dictionary, Japanese Wikipedia, TSUBAKI search result

1. INTRODUCTION

RTE (Recognizing Textual Entailment) is the task to detect whether a hypothesis (**H**) can be inferred/entailed by a text (**T**) [1]. RTE is important in basic analysis such as parsing and anaphora resolution as well as applications such as Question Answering (QA) [2], Information Retrieval (IR) and Machine Translation (MT).

Different from the conventional machine learning approaches, we take a structural matching approach to Japanese Recognizing Textual Entailment task. Let us consider the following simple example, whose correct answer is "NO". The conventional machine learning approaches would wrongly judge it as "YES" since all the words in **H** are matched with a word in **T**.

(1) **T:** 花王 は リンゴ 酸 を 使った
Kao (a company name)-tm malic-acid-acc used
「クリアクリーンプラスホワイトニング」を
"clear clean plus whitening"-acc
発売する。
release
(Kao releases "clear clean plus whitening", in which malic acid is used.)

H: 花王 は リンゴ 酸 を 発売する。
Kao-tm malic-acid-acc release

(Kao releases malic acid.)

By structural analysis, the following structure can be obtained: while in **T**, the ヲ (acc) [1] case of the predicate "発売する" (release) is "クリアクリーンプラスホワイトニング" (clear clean plus whitening), in **H**, the ヲ (acc) case of the predicate "発売 する" (release) is "リンゴ 酸" (malic-acid). By performing predicate-argument matching, the system can correctly judge it as "NO", since the ヲ (acc) case of the same predicate is different between **T** and **H**. The utilization of predicate-argument based matching is expected to achieve high precision compared to the conventional machine learning approaches.

Our proposed method regards predicate-argument structure, which consists of a predicate and zero or more arguments, as a basic unit of handling the meaning of text/hypothesis, and performs the matching between a text and hypothesis. Both the text and hypothesis are divided into predicate-argument structures based on predicate-argument structure analysis, and if all the predicate-argument structures in the hypothesis are matched to predicate-argument structures in the text, the hypothesis is judged to be entailed from the text.

To perform precise matching, wide-coverage lexical knowledge between words/phrases, such as synonym, is-a and antonym, is indispensable. To recognize the following entailment relation, the synonym between "原子力発電" (atomic power generation) and "原発" (the abbr. of "原子力 発電"), and the synonym between "排出" (emit) and "出す" (emit) are required.

(2) **T:** 原子力 発電 は 二酸化 炭素 を
Atomic power generation-tm carbon dioxide-acc
排出 しない エネルギー だ。
does not emit energy

[1] This paper uses the following abbreviations: nom (nominative), acc (accusative), dat (dative), ins (instrumental), loc (locative), abl (ablative), cmi (comitative), quo (quotative), all (allative), del (delimitative), and cmp (comparative).

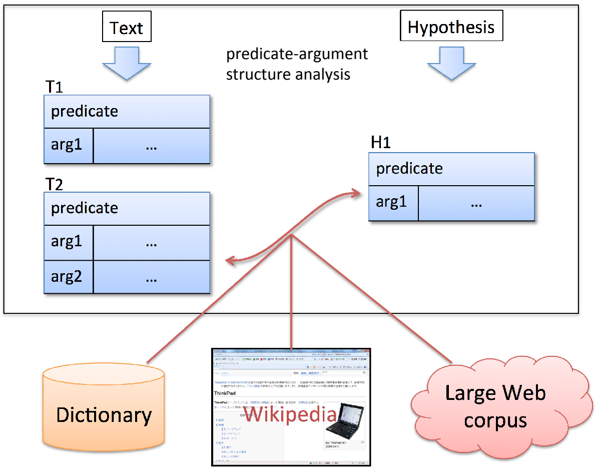

Figure 1: An overview of our proposed method.

(Atomic power generation is energy in which carbon dioxide is not be emitted.)

H: 原発 は　二酸化 炭素 を
Atomic power generation-tm carbon dioxide-acc
出さ ない。
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 a text and hypothesis. The synonym between "原子力 発電" (atomic power generation) and "原発" (the abbr. of "原子力 発電") is acquired from a dictionary, and the synonym between "排出" (emit) and "出す" (emit) can be recognized based on the distributional similarity using a large Web corpus. Figure 1 describes our proposed method.

We also take a machine learning approach to consider relatively shallow clues such as the overlap ratio of characters and morphemes as well as the result of predicate-argument matching method.

We participated in Japanese BC, MC, EXAM, and SearchTask subtasks of RITE-2 in NTCIR-10 [12].

2. RESOURCES

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

2.1 Automatic Acquisition of Relations between Words/Phrases

Synonym, is-a, and antonym relations are automatically extracted from an ordinary dictionary and Wikipedia using some manually-prepared patterns[9]. Examples of extracted relations are shown below.

synonym
アイス (ice) = アイスクリーム (ice cream)
タッチスクリーン (touch screen) = タッチパネル (touch panel)

is-a
夕食 (dinner) → 食事 (meal)
Genesis → 探査機 (probe)

antonym
暑い (hot) ⇔ 寒い (cold)

2.2 Distributional Similarity Calculation

Although synonym/is-a relation can be acquired from a dictionary/Wikipedia in the way introduced in Section 2.1, some near-synonymous relations cannot be acquired. For example, near-synonymous predicate relations such as "廃止" (abolish) and "中止" (stop) cannot be acquired.

Therefore, distributional similarity [5, 3], which is calculated based on the notion that "words that occur in similar contexts tend to be semantically similar", is calculated using a Web corpus. Then, those pairs whose distributional similarities are high are utilized when matching predicates/arguments in a text and hypothesis.

In this paper, the following types of distributional similarity are calculated:

1. between predicates
   - e.g.) "廃止" (abolish) and "中止" (stop)
2. between predicate-arguments where the argument is identical
   - e.g.) "人に うつる" (person dat move/reflect/catch) and "人 に 感染" (person dat catch)
3. a predicate and an idiom consisting of a predicate and an argument
   - e.g.) "魅了" (charm) and "心 を とらえる" (mind acc catch)
     Idioms consisting of a predicate and an argument are collected from [4].
4. noun and noun that has a redundant suffix
   - e.g.) "PET" and "PET 検査" (PET examination)

The distributional similarities for each type are calculated in the same framework, where only the feature (context) for each unit is different. First, a feature for each type is extracted from a corpus. Then, distributional similarity between units is calculated. The detail of the distributional similarity calculation is described in [9].

3. PREDICATE ARGUMENT STRUCTURE ANALYSIS

In both a text and a hypothesis, we perform morphological analysis using the Japanese Morphological Analyzer JUMAN[2] and syntactic/case analysis and zero anaphora resolution [8] using the Japanese parser KNP[3][7]. Then, they are converted to SynGraph data structure, we proposed earlier [10]. Based on the syntactic/case analysis, a text and a hypothesis are divided into predicate-argument structures.

An example of predicate-argument structure is shown in Figure 2. Each predicate-argument structure consists of a

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

(3) 東京 都 西 多摩 地区 で は 各地 で 季節 を 楽しむ イベント が 開か れる。
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

楽しむ (enjoy)
[syn] ⟨ 楽しむ ⟩

〈ヲ〉 (acc)	季節 (season) [syn] ⟨ 季節 ⟩ (season)
〈デ〉 (loc)	各地 (each place) [is-a] ⟨ 場所 ⟩ (place)

2-1

開か れる (be held)

〈ガ〉 (nom)	イベント (event)
〈デ〉 (loc)	地区 (area) [mod] 東京 都 西 多摩 (Tokyo Metropolis West Tama)

2-2

開く (hold)

〈ヲ〉 (acc)	イベント (event)
〈デ〉 (loc)	地区 (area) [mod] 東京 都 西 多摩 (Tokyo Metropolis West Tama)

Figure 2: An example of predicate-argument structure. (In the example sentence, the underlined phrases represent a predicate.)

predicate and zero or more arguments. For example, the sentence (3) in Figure 2 is decomposed to two predicate-argument structures (1-1) and (2-1) based on predicate-argument structure analysis, and predicate-argument structure (1-1) consists of the predicate "楽しむ" (enjoy) and the two arguments: case component "ヲ" (acc) and "デ" (loc).

Basically, a verb, adjective, and noun+copula are regarded as a predicate, and case components whose case marker is ガ (nom), ヲ (acc), ニ (dat), デ (ins/loc), カラ (abl), ト (cmi/quo), ヘ (all), マデ (del), and ヨリ (cmp) are regarded as an argument. If an argument has modified phrase, the attribute [mod] is assigned.

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.)

KNP makes a case analysis using case frames, which are automatically acquired from a large Web corpus [6]. In the case frames, the case alignment of two case frames, such as active and passive, is performed. For example, the "ガ" (nom) case of the case frame "開か れる" (the passive voice of "開く" (hold)) and the "ヲ" (acc) of the case frame "開く" (hold) is aligned. By using this alignment, predicate-argument structure as (2-2) in Figure 2 can be generated.

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

- deverbative noun

(4) デビッド・ケリー 氏 の 自殺 は
Mr. David Kelly-gen suicide-tm
英国 社会 を 揺さぶり 続けて いる。
British society-acc has shaken
(The suicide of Mr. David Kelly has shaken the British society.)

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

自殺 (suicide)

〈ガ〉 (nom)	デビッド・ケリー 氏 (Mr. David Kelly)

- apposition

(5) もともと バレンタインデー は
originally St. Valentine's Day-tm
3 世紀 の ローマ の 司祭
third century-gen Roma-gen priest
聖バレンタイン の 伝説 に 由来 する。
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, "司祭" (priest) and "聖バレンタイン" (St. Valentine) are recognized as an apposition relation, and the following predicate-argument structure is generated.

司祭 (priest)

〈ガ〉 (nom)	聖バレンタイン (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[4] is added. For example, the word "季節" (season) in the "ヲ" (acc) case component of predicate-argument structure (1-1) has the attribute [syn] whose value is ⟨ 季節 ⟩ (season). Similarly, if a word has a hypernym, the attribute [is-a] whose value is its SYNID is added.

[4] SYNID is an ID assigned to a synonymous group.

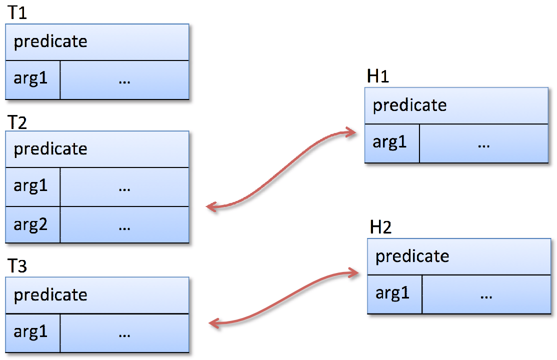

Figure 3: An overview of PA-matching method.

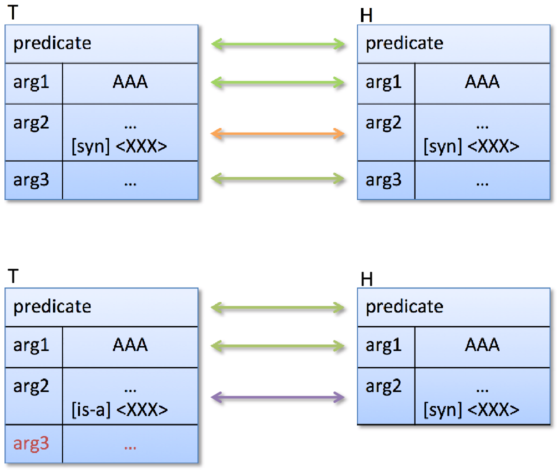

Figure 4: Matching between predicate-argument structure. (Green line, orange line, purple line represent the correspondence of surface form, the correspondence of SYNID, is-a relation, respectively.)

If a verb has negation expression, the negation flag is attached to the verb. For example, the verb "書か ない" (don't write) has the negation flag.

Some entailment relations are caused by numerical expressions. As for the handling for numerical expressions, refer to [9].

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 predicate-argument structures in a **T**, **H** is judged to be entailed from **T**, as shown in Figure 3.

If a predicate-argument in **H** is equal or "general" compared to a predicate-argument **T**, the entailment relation is identified. If there is something referred only in **H**, the entailment relation is not identified. The entailment of predicate-argument structures is defined as follows (Figure 4). First, the predicate-argument structure in **H** is totally the same as the predicate-argument structure in **T** (as shown in the top of Figure 4), 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 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 also identified (as shown in the bottom of Figure 4). The is-a relation of predicates/arguments is defined as follows:

is-a relation of predicates

- is-a relation of predicates:
  昼寝 (nap) → 寝る (sleep)
- lack of argument:
  昨日 (yesterday) 産まれた (be born) → 産まれた (be born)

is-a relation of arguments

- is-a relation of nouns:
  カツオ (bonito) → 魚 (fish),
  インコ (parakeet) → 鳥 (bird)
- lack of modification expression:
  自動車工場 (car factory) → 工場 (factory)

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

5. SVM-BASED METHOD

Although the method of entailment judgement introduced in the previous section is aimed at precisely matching between a text and hypothesis, it is often the case that precise matching cannot be achieved due to the gaps in the structure, parsing errors, shortages 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 Machine) 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
- if the result of PA-matching method is "Y", the value is set to be 1, otherwise 0
- if a predicate in **H** is matched to a predicate in **T**, the value is set to be 1, otherwise 0
- if a predicate in **H** is matched to a predicate in **T** and has a different negation flag from the predicate in **T**, the value is set to be 1, otherwise 0

In the 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 **T** and **H** and **H** and **T**, and then the result is classified into four classes (B, F, C, I) based on the SVM results, as shown in Table 1.

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

Table 2: Time expression classification.

type	examples	cue phrases
range	10 世紀 (10th century) 1930 年代 (1930s)	世紀 (century), 時代 (era), 年代 (s), ... から (from)... まで (to), · · ·
starting point only	10 世紀以後 (from 10th century) ビスマルクの失脚後 (after Bismarck fell)	以後 (from), 後 (after), · · ·
ending point only	1923 年より前 (before 1923) 第二次世界大戦前 (before World War II)	前 (before), より前 (before), 以前 (to), · · ·
year	2002 年 (year)	年 (year)

Table 1: System output judgement in the MC task.

SVM output against **T** and **H**	SVM output against **H** and **T**	system output
*	C	C
C	*	C
Y	Y	B
Y	N	F
N	Y	I
N	N	I

6. SEARCH TASK

In the SearchTask subtask, given a statement, the system judges the truth or falsity of the statement based on text knowledge base (KB), such as Wikipedia and textbook.

As we examined the development set of SearchTask, the task can be classified into two types: one is whether a time expression and the time when an event occurred are consistent or not (as shown in (6)), and the other is whether an event is true or not (as shown in (7)). As described in Section 7, approximately 30% of tasks in the development set contain a time expression.

(6) 16 世紀 から 18 世紀 の 間 に、
16th-century-abl 18th-century-gen between
ドイツ では, 関税 同盟が 発足 した。
Germany-loc-tm custom-union-nom started

(In Germany, custom union started between the 16th and 18th centuries.)

(7) イェニチェリ は、 オスマン 帝国 の
Yeniceri-tm Osman-Empire-gen
常備 軍 であった。
standing army

(Yeniceri was a standing army of Osman Empire.)

The identification of these two types is performed by checking whether a statement contains a time expression. As shown in Table 2, time expressions are classified into four types, and each type is recognized by using some cue phrases. The system judges the truth or falsity of the statement for these two types as follows:

a statement contains a time expression

The part after the removal of a time expression is regarded as an event. In (6), since "16 世紀から 18 世紀の間に" (between the 16th and 18th centuries) represents a time expression, "ドイツでは関税同盟が発足した" (In Germany, custom union started) is regarded as an event. Then, the year when the event occurred is estimated using the text knowledge base. Relevant sentences with the event are retrieved using TSUBAKI [11], an open search engine, regarding the event as a query to the text knowledge, and by matching between the event and the sentences using the PA-matching method, the year when the event occurred is estimated. For example, the year when the event "ドイツでは関税同盟が発足した" occurred can be estimated as 1834 by referring to the sentence like "ドイツでは，1834 年に関税同盟が発足した。" (In Germany, custom union started in 1834.) in knowledge base. The time expression is normalized, for example, "16 世紀から 18 世紀の間に" is normalized to "1501-1800". Finally, the truth judgement is determined by checking the consistency of the year the event occurred with the time expression. In (6), the system judges it as false.

a statement does not contain a time expression

Relevant sentences with the statement are retrieved regarding it as a query to the text knowledge, and if there is a sentence that entails it, the system judges it as true: otherwise false.

7. EXPERIMENTS

We participated in Japanese BC, MC, EXAM, and SearchTask subtasks of RITE-2 in NTCIR-10 [12].

7.1 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 two methods to the BC, MC, EXAM subtasks (SUBTASK = {BC, MC, EXAM}):

1. PA-matching method (introduced in Section 4)
   - RITE2-KYOTO-JA-SUBTASK-01.txt
2. SVM-based method (introduced in Section 5)
   - RITE2-KYOTO-JA-SUBTASK-02.txt,

and, in the BC, EXAM subtasks, PA-matching-loose method, where the partial string match was allowed in the argument matching and some manually-prepared matching rules were applied, was submitted (RITE2-KYOTO-JA-SUBTASK-03.txt).

[6] svmlight.joachims.org/.

Table 3: Experimental results of BC/MC/EXAM. (The numbers represent a macro F1.)

	BC dev	BC test	MC dev	MC test	EXAM dev	EXAM test
PA-matching Method	42.84	41.97	16.95	17.04	39.50	37.86
PA-matching-loose Method	46.75	46.42	–	–	43.01	38.57
SVM-based Method	78.18	74.50	49.13	50.12	62.06	56.82
Baseline	–	62.53	–	26.61	–	54.77

Table 4: Confusion matrix. (BC dev)

PA-matching Method

Macro F1: 42.84

		correct		
		Y	N	all
system	Y	12	2	14
	N	228	369	597
	all	240	371	611

PA-matching-loose Method

Macro F1: 46.75

		correct		
		Y	N	all
system	Y	23	7	30
	N	217	364	581
	all	240	371	611

SVM-based Method

Macro F1: 78.18

		correct		
		Y	N	all
system	Y	159	42	201
	N	81	329	410
	all	240	371	611

Table 5: Confusion matrix. (BC test)

PA-matching Method

Macro F1: 41.97

		correct		
		Y	N	all
system	Y	13	1	14
	N	243	353	596
	all	256	354	610

PA-matching-loose Method

Macro F1: 46.42

		correct		
		Y	N	all
system	Y	27	9	36
	N	229	345	574
	all	256	354	610

SVM-based Method

Macro F1: 74.50

		correct		
		Y	N	all
system	Y	168	61	229
	N	88	293	381
	all	256	354	610

Table 6: Confusion matrix. (MC dev)

PA-matching Method

Macro F1: 16.95

		correct				
		B	F	C	I	all
system	B	0	1	0	0	1
	F	1	13	1	1	16
	C	0	0	1	0	1
	I	82	193	63	192	530
	all	83	207	65	193	548

SVM-based Method

Macro F1: 49.13

		correct				
		B	F	C	I	all
system	B	63	33	17	17	130
	F	7	135	25	30	197
	C	0	0	0	0	0
	I	13	39	23	146	221
	all	83	207	65	193	548

Furthermore, the following two methods to the Search-Task subtask:

1. KB: Wikipedia
   - RITE2-KYOTO-JA-SearchTask-01.txt
2. KB: textbook
   - RITE2-KYOTO-JA-SearchTask-02.txt.

In EXAM, 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, it is obtained by transforming SVM score x with the sigmoid function as follows:

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

7.2 Result and Discussion

Table 3 shows macro F1 of PA-matching method, PA-matching-loose method, SVM-based method, and the baseline method (the RITE-2 organizer provided) in BC dev, BC test, MC dev, MC test, EXAM dev, and EXAM test. In all the subtasks, the SVM-based method performed better than PA-matching method in terms of macro F1, and SVM-based method performed better than the baseline method.

Tables 4, 5, 6, 7, 8, 9, show a confusion matrix of BC dev, BC test, MC dev, and MC test, EXAM dev, EXAM test, respectively.

The following example can be correctly judged as "Y" by recognizing the apposition relation between "首都" (capital) and "アシガバート" (Ashgabat) in **T**.

(8) **T:** アシガバート 空港 は、 トルクメニスタン の
Ashgabat Airport-tm Turkmenistan-gen
首都 アシガバート に ある 空港 である。
capital Ashgabat-dat is located in airport
(Ashgabat Airport is an airport that is located in Ashgabat, which is a capital of Turkmenistan.)

H: アシガバート は、 トルクメニスタン の
Ashgabat-tm Turkmenistan-gen
首都 である。
capital
(Ashgabat is a capital of Turkmenistan.)

Although the precision of PA-matching method was high, the recall was very low. The followings are examples where

Table 7: Confusion matrix. (MC test)

PA-matching Method

Macro F1: 17.04

		correct				
		B	F	C	I	all
system	B	0	0	1	1	2
	F	1	9	2	0	12
	C	0	0	1	0	1
	I	69	196	57	211	533
	all	70	205	61	212	548

SVM-based Method

Macro F1: 49.13

		correct				
		B	F	C	I	all
system	B	53	29	10	16	108
	F	5	144	20	38	207
	C	0	0	0	0	1
	I	12	32	31	158	233
	all	70	205	61	212	548

Table 8: Confusion matrix. (EXAM dev)

PA-matching Method

Macro F1: 39.50

		correct		
		Y	N	all
system	Y	5	1	6
	N	205	299	504
	all	210	300	510

PA-matching-loose Method

Macro F1: 43.01

		correct		
		Y	N	all
system	Y	16	17	33
	N	194	283	477
	all	210	300	510

SVM-based Method

Macro F1: 62.06

		correct		
		Y	N	all
system	Y	92	59	151
	N	118	241	359
	all	210	300	510

our system could not recognize the synonymous/entailment that cause false negatives:

- 連隊長となった (became a regimental commander) = 連隊長に着任した (became a regimental commander) [synonym in predicate-argument]
- 張り付けにされた (crucified) → 罰を受けた (take the rap) [entailment in predicate-argument]
- 唐は高句麗を滅ぼした (Tang destroyed Kokuryo) → 高句麗は唐と戦った (Kokuryo made war with Tang) [presupposition in predicate-argument]

Table 10 shows macro F1 of SearchTask dev and SearchTask test. The macro F1 when Wikipedia is used for KB is almost the same when the textbook is used for KB. Tables 11, 12 show a confusion matrix of SearchTask dev, SearchTask test, respectively.

The following example can be correctly judged as "Y". The system could correctly estimate the year when the event "クックがオセアニアを探検した" (Cook explored Oceania) occurred as 1774, and judge this statement as true.

Table 9: Confusion matrix. (EXAM test)

PA-matching Method

Macro F1: 37.86

		correct		
		Y	N	all
system	Y	0	2	2
	N	173	273	446
	all	173	275	448

PA-matching-loose Method

Macro F1: 38.57

		correct		
		Y	N	all
system	Y	1	1	2
	N	172	274	446
	all	173	275	448

SVM-based Method

Macro F1: 56.82

		correct		
		Y	N	all
system	Y	61	58	119
	N	112	217	329
	all	173	275	448

Table 10: Experimental results of SearchTask. (The numbers represent a macro F1.)

	dev	test
KB:Wikipedia	45.74	46.57
KB:textbook	45.46	45.41

Table 11: Confusion matrix. (SearchTask dev)

KB: Wikipedia

Macro F1: 45.74

		correct		
		Y	N	all
system	Y	20	8	28
	N	190	292	482
	all	210	300	510

KB: textbook

Macro F1: 45.46

		correct		
		Y	N	all
system	Y	19	7	28
	N	191	293	482
	all	210	300	510

(9) クック が、 18 世紀に オセアニア を
Cook-nom 18th-century-dat Oceania-acc
探検 した。
explored
(Cook explored Oceania in the 18th century.)

Table 13 shows a confusion matrix of SearchTask dev containing a time expression. Compared Table 13 with Table 11, the accuracy regarding statements containing a time expression is relatively high, and thus the improvement of the accuracy regarding statements do not contain a time expression is our future work.

8. 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

Table 12: Confusion matrix. (SearchTask test)

KB: Wikipedia

Macro F1: 46.57

		correct		
		Y	N	all
system	Y	17	10	27
	N	156	265	421
	all	173	275	448

KB: textbook

Macro F1: 45.41

		correct		
		Y	N	all
system	Y	15	11	26
	N	158	264	422
	all	173	275	448

Table 13: Confusion matrix. (SearchTask dev: containing a time expression)

KB: Wikipedia

Macro F1: 59.38

		correct		
		Y	N	all
system	Y	16	5	21
	N	43	89	132
	all	59	94	153

KB: textbook

Macro F1: 59.38

		correct		
		Y	N	all
system	Y	16	5	21
	N	43	89	132
	all	59	94	153

as synonym and is-a were automatically acquired from a dictionary, Web corpus and Wikipedia, and were utilized when matching text and hypothesis. We also participated in the SearchTask subtask.

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

9. REFERENCES

[1] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. *Machine Learning Challenges*, pages 177–190, 2006.

[2] S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912, 2006.

[3] Z. Harris. *Mathematical Structures of Language.* Wiley, 1968.

[4] C. Hashimoto, S. Sato, and T. Utsuro. Detecting Japanese idioms with a linguistically rich dictionary. *Language Resources and Evaluation: Special Issue on Asian Language Technology*, 40(3-4):243–252, 2006.

[5] J.R.Firth. *Studies in Linguistic Analysis*, chapter A synopsis of linguistic theory. Oxford, 1957.

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

[7] 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.

[8] R. Sasano and S. Kurohashi. A discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP2011)*, pages 758–766, 2011.

[9] T. Shibata and S. Kurohashi. Predicate-argument structure based textual entailment recognition system exploiting wide-coverage lexical knowledge. *Special Issue of ACM TALIP on RITE (Recognizing Inference in TExt)*, 11(4), 12 2012.

[10] 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.

[11] K. Shinzato, T. Shibata, D. Kawahara, C. Hashimoto, and S. Kurohashi. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Proceedings of Third International Joint Conference on Natural Language Processing (IJCNLP2008)*, pages 189–196, 2008.

[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.