Overview of NTCIR-9 RITE: Recognizing Inference in TExt

Hideki Shima* Hiroshi Kanayama** Cheng-Wei Lee# Chuan-Jie Lin †
Teruko Mitamura* Yusuke Miyao ‡ Shuming Shi+ Koichi Takeda**

*Carnegie Mellon University, USA **IBM Research – Tokyo, Japan #Academia Sinica, Taiwan
† National Taiwan Ocean University, Taiwan ‡ National Institute of Informatics, Japan
+Microsoft Research Asia, P.R. China

hideki@cs.cmu.edu, hkana@jp.ibm.com, aska@iis.sinica.edu.tw, cjlin@mail.ntou.edu.tw
teruko@cs.cmu.edu, yusuke@nii.ac.jp, shumings@microsoft.com, takedasu@jp.ibm.com

ABSTRACT

This paper introduces an overview of the RITE (Recognizing Inference in TExt) task in NTCIR-9. We evaluate systems that automatically recognize entailment, paraphrase, and contradiction between two texts written in Japanese, Simplified Chinese, or Traditional Chinese. The task consists of four subtasks: Binary classification of entailment (BC); Multi-class classification including paraphrase and contradiction (MC); and two extrinsic application-oriented datasets: Entrance Exam and RITE4QA. This paper also describes how we built the test collection, evaluation metrics, and evaluation results of the submitted runs.

Keywords: *test collections, entailment, contradiction, paraphrase, evaluation*

1. INTRODUCTION

We organized the NTCIR-9 RITE (Recognizing Inference in TExt) task which evaluates systems that recognize entailment, paraphrase and contradiction relations between a given text pair. The problem, often called Recognizing Textual Entailment (RTE), can be positioned as a basic research rather than applied one. In the past, NTCIR has been focusing on applied problems where a system can directly help end users to achieve a certain goal in an Information Access (IA) task. Nevertheless, we proposed this task because we thought making an advancement in textual inference research can greatly benefit us since the problem is generic across various Information Access applications, e.g. Question Answering (QA; between question and answer-bearing sentence) [1][2], Information Retrieval [3][4] (IR; for query expansion and exhaustive high-recall retrieval), Information Extraction (for increasing a chance of matching a vocabulary in a pattern) [5], Text Summarization (for measuring the meaning redundancy among summary candidates) [6][7], Intelligent Tutoring [8] for checking whether a student's answer can entail a reference answer, and automatic evaluation for Machine Translation [9] and Text Summarization [10] (for improving meaning similarity model or expanding human references).

Recognizing Textual Entailment is a very active research field in European language communities. PASCAL/TAC RTE has been conducting a series of shared task evaluations [11][12][13][14][15] for 2-way or 3-way entailment relation classification tasks, as well as EVALITA/IRTE [16] for Italian language. As one direction, the task evolves into cross-/multi-lingual entailment [17][18][19]. When it comes to Asian languages, for example in Japanese, there has been previous studies on textual entailment [20][21][22][23] using in-house data. Odani et al [24] built and released an entailment evaluation dataset created by hand considering a balance of linguistic phenomena. The NTCIR-9 RITE task is the first large-scale open evaluation effort for Japanese (JA), Simplified Chinese (CS) and Traditional Chinese (CT). A system has to process long texts which are extracted from actual texts with minimum post-edits, making the task very challenging. For Chinese language, there may be less resources and tools available as compared to Japanese, which results in adding even more challenges. In order to help reduce participants' effort in non-research parts, we provided a framework software called RITE SDK[1], with which one can easily build a participating system and evaluate runs and a resource pool[2] .

The RITE task consists of four subtasks: BC, MC, Entrance Exam and RITE4QA (see Figure 1 for a quick comparison among subtasks). In the BC (Binary Class) subtask, given a pair of texts (t_1, t_2), a system automatically identifies if t_1 *entails* t_2 or not. The premise t_1 entails the hypothesis t_2 if a human reading t_1 would infer that t_2 is most likely true [12]. Note that logical entailment and textual entailment are slightly different. Texts one needs to handle in Information Access applications are almost always with vagueness rather than with clear logicality. Thus, most likely is an important key word to allow us to work on real world texts. Also, keep in mind that we assume a human can utilize common understanding of language and common background knowledge when inferring meaning in texts [15].

In the MC (Multi Class) subtask, a system has to recognize entailment direction (forward, reverse and bidirectional), as well as contradiction and none of the above. The RITE task is the first of a kind to include entailment direction recognition and contradiction into one evaluation challenge.

The Entrance Exam and RITE4QA subtasks are similar to the BC and MC, however, their dataset has natural distribution of linguistic phenomena as they are created from existing real task oriented data, namely, past Japanese National Center Test for University Admissions and Factoid Question Answering datasets, respectively. There are previous works that relate textual entailment with reading comprehension [11][25].

[1] http://code.google.com/p/rite-sdk

[2] http://artigas.lti.cs.cmu.edu/rite/Resources

Figure 1. Overview of NTCIR-9 RITE.

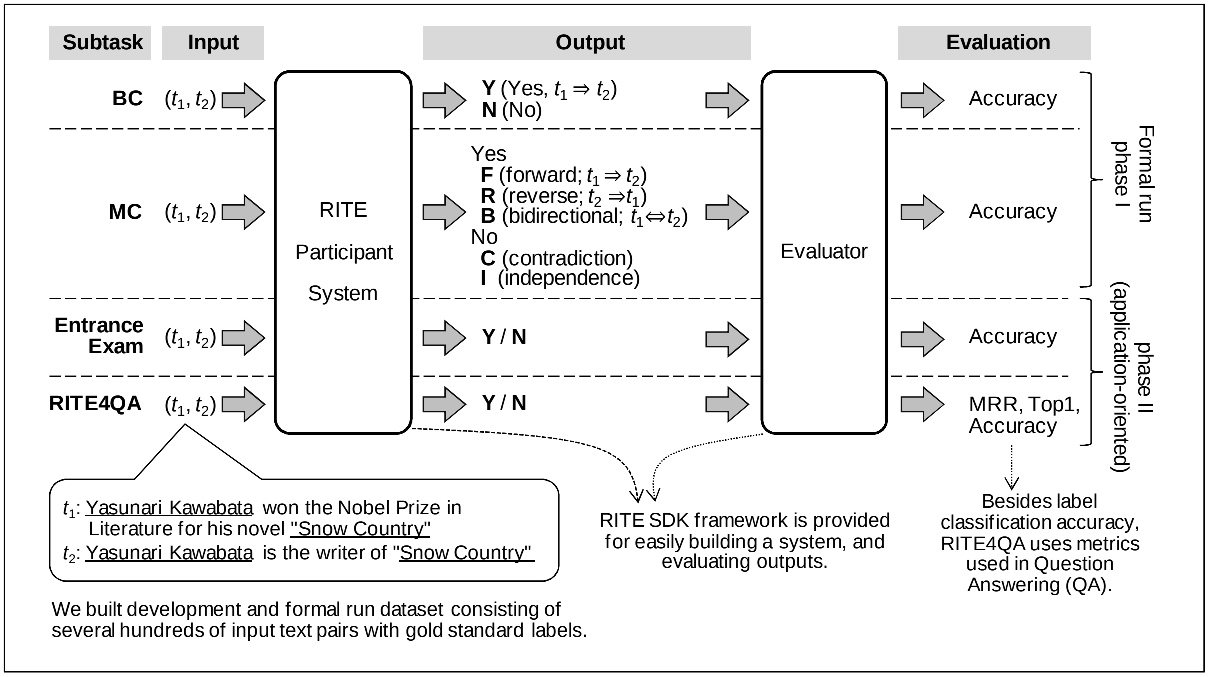

Subtask
Input
Output
Evaluation
BC
(t_1, t_2)
Y (Yes, $t_1 \Rightarrow t_2$)
N (No)
Accuracy
MC
(t_1, t_2)
Yes
F (forward; $t_1 \Rightarrow t_2$)
R (reverse; $t_2 \Rightarrow t_1$)
B (bidirectional; $t_1 \Leftrightarrow t_2$)
No
C (contradiction)
I (independence)
Accuracy
Entrance Exam
(t_1, t_2)
Y / N
Accuracy
RITE4QA
(t_1, t_2)
Y / N
MRR, Top1, Accuracy
RITE Participant System
Evaluator
Formal run phase I
phase II (application-oriented)
t_1: Yasunari Kawabata won the Nobel Prize in Literature for his novel "Snow Country"
t_2: Yasunari Kawabata is the writer of "Snow Country"
We built development and formal run dataset consisting of several hundreds of input text pairs with gold standard labels.
RITE SDK framework is provided for easily building a system, and evaluating outputs.
Besides label classification accuracy, RITE4QA uses metrics used in Question Answering (QA).

Table 1. Pairs in the BC dataset.

	Y	N	Total
JA (dev)	250	250	500
JA (test)	250	250	500
CS (dev)	265	142	407
CS (test)	263	144	407
CT (dev)	266	155	421
CT (test)	450	450	900

Table 2. Pairs in the MC dataset.

	F	R	B	C	I	Total
JA (dev)	110	110	75	80	65	440
JA (test)	110	110	75	80	65	440
CS (dev)	92	85	88	72	70	407
CS (test)	101	91	71	74	70	407
CT (dev)	87	97	82	74	81	421
CT (test)	180	180	180	180	180	900

Table 3. Pairs in the Entrance Exam dataset.

	Y	N	Total
JA (dev)	204	295	499
JA (test)	181	261	442

Table 4. Pairs in the RITE4QA dataset.

	Y	N	Total
JA (test)	106	858	964
CS (test)	130	552	682
CT (test)	130	552	682

Unlike these works, Entrance Exam subtask covers wide range subjects including Domestic and World History, Politics, Economy, and Modern Society. The RITE4QA subtask is inspired by a series of Answer Validation tasks at CLEF [26][27][28]. Ours is unique in a sense that we evaluate using actual QA evaluation metrics to make the outcome comparable to QA systems.

As a first attempt in NTCIR, the goal of RITE is to establish the baseline for this new evaluation challenge. We also aim to contribute in continuing growth of the related problem domain. To this end, we will present some efforts including resource pool, ablation study, and a discussion on what left to be done to make advancement in the community.

2. TASK OVERVIEW

We constructed datasets consisting of hundreds of labeled pairs for each subtask, which numbers are summarized in the Table 1 through Table 4 above. In the rest of this section, we will describe how we built these datasets.

2.1 BC Subtask

As described in the previous section, the binary-class (BC) subtask is about analyzing text pairs and assigning binary labels on the pairs. For instance, Figure 1 shows an example where a system needs to infer that someone winning a Nobel Prize in Literature means that the same person is a writer.

The RITE datasets were created in the following way. First, the RITE organizers proposed a small set of *sample* dataset on an online collaborative spreadsheet, and presented them to participants. Then, participants either posted some feedbacks to organizers, or even put additional samples with comments, if any. The sample data created through this hands-on exercise was a very useful material to be discussed among the RITE community. The BC dataset for system development and formal run (hereafter called *dev* and *test* respectively) have been created using this sample as a reference.

For building the Japanese dev and test datasets, about ten college students (belonging to different undergrad/graduate program with different majors) were hired as annotators. They initially studied general trends from the sample data, and then collected pairs from a newswire corpus (Mainichi newspaper 2002-2005 with 410k articles) and assigned labels following a minimum guideline[3]. The guideline contains a brief introduction of the task, steps to use the online spreadsheet and corpus search tool[4], and some tips and common mistakes. In order to cover a wide range of topics in the dataset, we recommended the students to visit the random page redirection URL in Wikipedia[5], and to try using terms in the randomly chosen page as query terms to retrieve documents in the corpus. The students collected a sentence or a series of sentences, and asked to do post-edits only when needed (e.g. solving coreference with another sentence, fixing particles, removing information from t_2 that are not inferable from t_1 when creating a positive example, etc). They were also told to select sentences so that simple surface term overlap does not result in Y or N label easily. Four students were then independently asked to annotate additional labels just by looking at the text pair. We discarded pairs where less than three agreed on the same label. As a result, we obtained 1000 pairs where the inter-annotator agreement measured in Fleiss' Kappa among the four was 0.829. Finally, we randomly split the dataset into dev and test, taking care of the balanced label distribution.

Due to limited resources and time constraints, the Traditional Chinese dev data was mostly created from the NTCIR-7 CCLQA [29] dataset based on the gold standard answer nuggets of complex questions. The idea is to create entailment pairs that may be useful for answering complex questions. There were 1137 answer nuggets from 100 NTCIR-7 CCLQA complex questions. For each answer nugget, we asked the annotator to search for similar sentences by sending queries to a web search engine with proper keywords based on the content of the answer nugget. These similar sentences were collected, tidied (if there were improper words, sentence structures, or other noise) and then paired with their source answer nuggets and categorized into the five different entailment labels (We shared the pairs with the MC subtask. The details about this subtask are available in the next subsection). The meaning of some collected sentences as well as the answer nuggets may be modified to create extra pairs if the annotator thought it has interesting entailment issues that are worth exploring. Three annotators were involved in the creation of Traditional Chinese development set. Each pair was created by one of the annotators and reviewed by all of them. In the end, 421 out of 485 created pairs were agreed by the three annotators and became the development set for both CT-BC and CT-MC subtasks.

The Traditional Chinese test dataset was created from two different sources. The first source was the answer nuggets from NTCIR-8 CCLQA [30]. The way we used to create pairs from this source was the same as the way we used to create the development set from the NTCIR-7 data. 677 pairs were created by this way.

The second source was relevant documents retrieved in the past NTCIR CLIR tasks. Passages were two consecutive clauses separated by punctuation marks suggesting an end of a sentence or a clause. Top similar passage pairs (each passage selected from different documents) were collected, and then they were filtered or revised by an annotator to make it more relevant and inferential. Each pair was labeled in MC classes by three annotators. Only the pairs agreed by all the annotators could be selected in the test set. Among the top 2200 similar passage pairs, only 785 pairs remained and were revised, owing to the replication of the articles in the document collection. After the voting stage, 483 of them were agreed by all the annotators.

In order to make a label-balanced CT-MC test set, 180 pairs were randomly selected from each set of pairs labeled as ‘F’, ‘B’ and ‘I’. The two classes, R and C, with insufficient number of pairs were expanded by the following method: some unselected F-pairs (pairs labeled as ‘F’) were randomly selected and swapped into R-pairs; more unselected F- and B-pairs labeled were selected and revised into contradictions. The steps were repeated until both classes contained 180 pairs, respectively. In the end, we have 900 MC pairs in the test set with 180 pairs in each class. The pairs in the first part (ID=422~1092) came from the CCLQA data and those in the second part (ID=1093~1321) came from the CLIR data.

The CT-BC test set was converted from the CT-MC test set. Pair IDs were re-assigned since the ID of MC-subtask starts from 422 and the ID of BC-subtask from 1. Many pairs were swapped so that t_1 is longer than t_2. Note that in such a case, those swapped F-pairs became R-pairs and the swapped R-pairs became F-pairs. Moreover, some of the un-swapped R-pairs were randomly swapped into F-pairs in order to make the test set binary-label-balanced, despite the length requirement. All ‘F’ and ‘B’ labels were then converted into ‘Y’, while ‘R’, ‘C’, and ‘I’ were converted into ‘N’ pairs. In the end, we have 900 BC pairs in the test set with 450 pairs in each class. Similarly, the first group (ID=1~671) was created from the CCLQA data and the second group (ID=672~900) was created from the CLIR data.

The Simplified Chinese dev data contains 407 pairs, which includes samples by the organizers (ID=1~5), samples by the participants (ID=6~8), data by annotators (ID=9~50), and those transliterated from CT (ID=51~407 where 3 removed for label disagreements). Most sentence pairs in the Simplified Chinese test were created by manually transliterating the CT test data. Others were created by the annotators. For both the dev data and the test data, pairs were selected so that all the annotators agreed on the same label. That is, pairs without agreement were discarded.

Evaluation is done based on the label classification accuracy, or the ratio of correctly returned labels. Using the Iverson bracket notation, the accuracy can be formalized as follows:

$$\mathrm{Accuracy} = \frac{1}{\#\,\mathrm{pairs}} \sum \left[\mathrm{output\ label\ is\ correct}\right].$$

This metric is reported for all the subtasks.

2.2 MC Subtask

The multi-class classification (MC) subtask is different from the other three in a sense that a system needs to classify a pair into one of five categories considering entailment direction, paraphrase and contradiction. The output labels in the MC subtasks are the following:

- F: forward entailment (t_1 entails t_2 AND t_2 does not entail t_1).
- R: reverse entailment (t_2 entails t_1 AND t_1 does not entail t_2).
- B: bidirectional entailment (t_1 entails t_2 AND t_2 entails t_1).
- C: contradiction (t_1 and t_2 contradict, or cannot be true at the same time).
- I: independence (otherwise)

[3] http://artigas.lti.cs.cmu.edu/rite/Annotator_manual

[4] http://code.google.com/p/sepia/

[5] http://ja.wikipedia.org/wiki/Special:Randompage

One of motivations being that, for example in Text Summarization, knowing textual entailment direction helps to choose one from multiple summary candidate sentences. Contradiction detection is also meaningful since it's a "fundamental task in text understanding" [31] which is applicable to practical research fields such as conflicting position detection in political candidate debates [32], conflicting opinion analysis in user reviews [33], Question Answering and by Multi-Document Summarization [34]. The RITE MC subtask is novel in terms of addressing various classes, which are not traditionally evaluated in one problem.

One specific point we paid attention to when building the MC dataset is to shorten the length of sentences so that the length does not become the strong indicator of entailment direction. In the BC dataset, it is often the case that t_1 is long, and t_2 is much shorter.

The inter-annotator agreement for the Japanese MC dataset among the four annotators was 0.759 in Fleiss' Kappa.

2.3 Entrance Exam Subtask

The Entrance Exam subtask runs in the same setting as the BC subtask; a participating system is asked to determine Yes or No for each text pair. All the pairs in the datasets provided in this subtask are created based on actual entrance exams for university admission in Japan, which is called the National Center Test for University Admission (Center Test; Daigaku Nyushi Center Shiken). All Japanese national universities and many private ones adopt Center Test for their admission or as their first-stage examination, and all students who are going to enter those universities/colleges must pass this test. Center Test provides multi-choice questions such as the following:

> Choose the most appropriate statement about the Ottoman Empire from 1 to 4.
>
> 1. The peak of this country was during the reign of Suleyman.
>
> 2. The official religion of this country was Shiah Muslim.
>
> : : :

Examinees are required to answer such questions using their knowledge.

To create text pairs for the Entrance Exam subtask from Center Test, we assume that correct answers are supported by evidential texts in a knowledge source such as textbooks and Wikipedia. Therefore, we can create "Yes" pairs from correct statements while "No" pairs from wrong statements, by extracting t_2 from Center Test choices and t_1 from supporting texts in a knowledge source. For example, we could find the following texts from Wikipedia.

> ... Suleyman set 13 times of military expedition with great success, and led the Ottoman Empire to its peak. ...
>
> ... While Sunna constituted the majority in the south part of the Ottoman Empire, many Shiah Muslims lived in the south Iraq. ...

From these texts we can judge the statement 1 is true, while the statement 2 is false. We can create a "Yes" pair from the first sentence and the statement 1, and a "No" pair from the second sentence and the statement 2.

Following this observation, we created text pairs and their Yes/No labels from statement-style choices of Center Test. Annotators are asked to find relevant texts from Wikipedia for each statement. For correct statements, annotators found a text that supports the statement. For wrong statements, annotators found a contradictory text, or, if such a text is not found, an irrelevant text that includes some keywords in the statement. Since raw texts extracted from Wikipedia involve Wikipedia-specific writings and might lack contextual information (e.g. coreferences), annotators post-edited extracted texts so that they form natural sentences that can be read without the contexts. In some cases multiple sentences are necessary to support one statement. In this case, we also edit them to form a single sentence.

It should be noted that the process of data creation is different from the BC/MC subtasks where text pairs are first created and labels are assigned afterwards. In the Entrance Exam subtask, t_2 and labels are determined by the Center Test questions, while annotators are asked to find t_1 from Wikipedia.

2.4 RITE4QA Subtask

The RITE4QA subtask is also same as the BC subtask in terms of input and output. We assume a real application scenario where a RITE system works as if it’s an embedded answer validation module plugged-in to a QA system. Suppose there is a QA system which has answer extraction capability (without ranking or filtering final answer candidates), and it expects a certain module to score and rank final answers to be returned. This way, the impact of RITE to an end-to-end application can be measured.

Figure 2 illustrates how pairs are created for the RITE4QA evaluation. As the source data, we use the dataset from the past NTCIR-6 CLQA task (in the JA & CT monolingual QA track) [35], as well as answers (system responses) from one of the best QA runs. There are 200 JA questions and 150 CT questions available[6]. For each question, a system returned up to 5 answers each with a source document ID. Since we would like to simulate a realistic scenario where a RITE system is used in QA framework, we tried to automate the process of creating the dataset except for improving template generation algorithm with a slight feedback. The almost fully-automatic mechanism is described as follows.

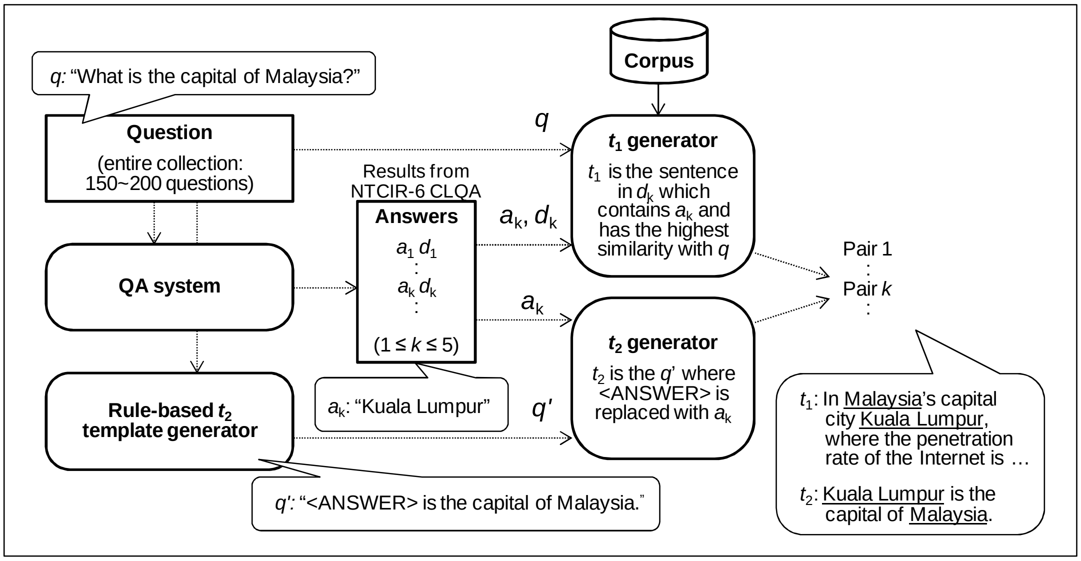
- t_1 is an answer-bearing sentence, or a sentence that contains an answer. If there are multiple answer-bearing-sentences in a document, we automatically selected the sentence with the highest lexical overlap with the question.
- t_2 is basically a question transformed from interrogative to affirmative form. The question’s WH-word part has been replaced with an answer.
- The expected label is Y for a pair created from a correct answer (which must be supported with a valid supporting document) and N otherwise.
- CS data has been transliterated using the Google Translate MT service[7].

There could be minor errors generated through this automatic process. Also, note that a Y label do not necessarily represent an entailment between t_1 and t_2 (e.g. sometimes t_1 lacks coreferential information from previous sentences; t_2 has additional information that cannot be inferred from t_1).

[6] One invalid question in each dataset (CLQA2-JA-T1087-00 in JA and CLQA2-ZH-T3069-00 in CS/CT) was removed.

[7] http://translate.google.com

Figure 2. The mechanism for automatically creating the RITE4QA pairs from the past QA test collection.


q: "What is the capital of Malaysia?"
Question
(entire collection: 150~200 questions)
QA system
Rule-based t2 template generator
q': "<ANSWER> is the capital of Malaysia."
Results from NTCIR-6 CLQA
Answers
a1 d1
ak dk
(1 ≤ k ≤ 5)
ak: "Kuala Lumpur"
Corpus
q
ak, dk
ak
q'
t1 generator
t1 is the sentence in dk which contains ak and has the highest similarity with q
t2 generator
t2 is the q' where <ANSWER> is replaced with ak
Pair 1
Pair k
t1: In Malaysia's capital city Kuala Lumpur, where the penetration rate of the Internet is …
t2: Kuala Lumpur is the capital of Malaysia.

Participants were allowed to use both dev and test data from the BC and MC subtasks, in order to develop a system as long as it's clearly described in the system paper. On the other hand, participants were not allowed to utilize past NTCIR QA data.

As for the evaluation, the key metric we used is Mean Reciprocal Rank (MRR), instead of the label classification accuracy score used in the previous three subtasks. The first reason is that we would like to use a metric which is comparable with an extrinsic QA task performance. The second reason is that, because of the skewed label distribution, it is easy to cheat the metric in this subtask (one can simply return the major label only).

MRR is the mean of Reciprocal Ranks (RR) over the entire questions:

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}.$$

In order to get a ranked list to evaluate, we used confidence scores that were submitted together with each label from the RITE systems. The ranking criterion from the highest rank to the lowest is as follows: Y with high confidence < Y with low conf < N with low conf < N with high conf. Given this ranked list, we can subsequently calculate the highest rank of the correct label. If it is at the k-th rank, $rank_i$=k and so RR = 1/k. In order to handle tied labels, we used the rank averaging mechanism as seen in the Spearman's rank correlation coefficient. For instance, if there are three outputs ranked 2nd in the list, we will obtain (2+3+4)/3 = 3, and therefore RR is 1/3. If we don't use this mechanism, one can easily cheat the metric by returning five Y labels with 1.0 confidence to get the RR of 1.

We additionally reported the Top1 score also used in the NTCIR-6 CLQA task:

$$\mathrm{Top1} = \frac{1}{|Q|}\sum_{i=1}^{||Q|}\left[\text{highest ranked is correct}\right].$$

2.5 Formal Run Settings

We had two phases in the formal runs. The BC and MC subtasks were conducted in the phase I in one week, and the Entrance Exam and RITE4QA subtasks were conducted in the phase II in a separate week. The formal run period was relatively close to the NTCIR workshop, in order to allow participants to spend time on a system development as much as possible. Before opening the formal run test data, participants were required to freeze the system development. Once the system is frozen, no system updates were allowed except for bugs that are trivial and non-essential (e.g. output formatting bug).

The dev (training) and test (formal run) data were provided in the following xml format[8] in all the four subtasks. Note that the label field was not available in the test data. T

```
<dataset>
  <pair id="1" label="Y">
    <t1>パルテノン神殿は、古代ギリシア時代にアテナイのアクロポリスの上に建設された、アテナイの守護神であるアテーナーを祀る神殿である。/The Parthenon, built on the Acropolis of Athens in the ancient Greece period, is a temple dedicated to Athena, the protector of Athens.</t1>
    <t2>パルテノン神殿の建つ丘は、アクロポリスと呼ばれている。/The hill where the Parthenon template is located, is called Acropolis.</t2>
  </pair>
  <pair id="2" label="N">
    <t1>パルテノン神殿は、ドーリア式神殿の最高傑作と言える作品である。/The Parthenon temple is a masterpiece of Doric order temples.</t1>
    <t2>パルテノン神殿は，ヘレニズム文化の影響下で建設された。/The Parthenon temple was built under the influence of the Hellenism culture.</t2>
  </pair>
      : : :
</dataset>
```

[8] The example was taken from the Entrance Exam subtask dev dataset. English translations in italics are attached for the reader's convenience.

For each subtask, a participating team was allowed to submit up to 3 runs. Each line in the submission file was in the following format.

```
ID [SPACE] LABEL [SPACE] CONFIDENCE [CR]
```

The confidence score column could take a real number between 0 and 1. In the BC and MC subtasks, the confidence column was optional (but recommended). In the Entrance Exam (for future evaluation purpose) and RITE4QA (as explained in the previous subsection) subtasks, the confidence column was strongly recommended for tie-breaking multiple labels.

3. TASK ORGANIZATION EFFORTS

We tried to address the following important aspects when designing and organizing the task.

Abstract laboratory experiments. Major evaluation conferences such as TREC, CLEF and NTCIR are modern examples of the "Cranfield evaluation paradigm" [36] where abstraction of a real task is done with a system-centric (rather than user-centric) evaluation to avoid affects from uncontrollable variables [37]. Even with this paradigm, complicated IA systems such as QA are hard to evaluate in component level, due to high inter-component dependencies, and accumulation of errors through multiple steps (e.g. question analysis, document retrieval, named entity extraction, answer candidate reranking etc in QA). RITE abstracts away complexities and focuses on a key semantic processing need commonly exist in various IA systems. With a capability to conduct fully-automatic evaluation, we believe the laboratory evaluation infrastructure of RITE should enable a quick research iteration (from analysis, hypothesis design, implementation to evaluation) which will results in advancements of the textual entailment research. As a result of establishing an abstract experiment paradigm, one can expect a potentially synergy on improving many IA applications.

Lowering barrier to entry. We provided the RITE-SDK and the resource pool to help participants to quickly build a system. With a common framework, experiment reproducibility/repeatability can be improved because one can detach the core component of a system and share with others.

Generalizability / Domain Portability. Although being in an abstract setting, the Entrance Exam and RITE4QA subtasks capture salient aspects of real tasks. These two subtasks serve at least two important purposes: testing knowledge domain portability and testing application domain portability. Due to a common framework and standardized input and output format, participants are able to reuse components across domain.

Community-driven. Even though RITE is the first-of-a-kind task in NTCIR, we envisioned a community-driven task which reflects needs among participants better, as well as keeping the task sustainable. To this end, we set up some environments where participants can be involved in task design, such as a mailing list for discussion and online synchronous spreadsheet for proposing sample data among organizers and participants.

Accountability. We encouraged participants to do an ablation study which is done by removing one resource, tool, or algorithm at a time, and see its impact to the overall system (lower performance indicates higher importance). Remember that participants can take advantage of automatic evaluation in the RITE task, and quickly try out multiple different experiments. In that way, participants can avoid a system from being a complicated *black box*, but can instead see it as a collection of *building blocks.*

Social impact. The Entrance Exam subtask's long-term ultimate goal is to develop a system that can result in a competitive score in a college-level entrance exam. It can be a good grand-challenge showcase for a scientific outreach because of its clearness, familiarity and toughness. The progress toward the goal is easily measurable, which is a nice property to have in a grand challenge.

Difficulty level. We understand that it may be too early for a relatively new community to tackle real texts with a lot of challenges, and the difficulty of the task should be in an adequate level to encourage participation. However, we avoided arbitrary modifications to original texts as much as possible, so as not to make the task too easy. As a result, the BC and MC datasets, especially Japanese, are very difficult. See also Table 5 where lower BC JA baseline score indicates its relative difficulty as compared to CS and CT. The scores also indicate that numbers between different subtasks or languages are not comparable.

3.1 Baselines

We provided baseline runs, which are useful for measuring relative performance against a certain standard. We can also use them for comparing difficulties among different subtasks and/or languages.

The *character overlap* baseline is based on a very simple algorithm, but known to work reasonably good in the past English RTE challenges [12]. As a unit of overlap comparison, we used characters rather than words because of its straightforwardness and error-free nature. Some studies show that the character may be the better unit to be used in certain tasks in Japanese [38] and Chinese [39]. The algorithm works in the following way: The percentage of characters in t_2 existing in t_1 is calculated with clipped-counting which truncates each character's count, if necessary, to not exceed the largest count observed in t_1 [40]. If the percentage is over a certain threshold θ, the algorithm returns the Y label, and otherwise the N label. The thresholds were trained in 0.05-scale parameter sweep using the development data.

For the MC subtask, the same approach is used to determine the entailment direction for the F, R, and B labels. If there is no entailment exists according to the algorithm, we randomly assigned either C or I label.

See Table 5 (also shown in Table 7 through Table 17) for the summary of this baseline's results. In the RITE4QA subtask, we provide three additional baselines and one oracle score (Table 15 and Table 17).

Table 5. Evaluation results for the character overlap baseline.

Subtask	Lang	θ	Dev	Test
BC	JA	0.60	0.5280	0.5160
	CS	0.55	0.7543	0.7617
	CT	0.55	0.7553	0.6667
MC	JA	0.60	0.4742	0.4682
	CS	0.70	0.5356	0.5315
	CT	0.65	0.5091	0.4885
Entrance Exam	JA	0.80	0.6673	0.6516
RITE4QA	JA	0.60	-	0.4180
	CS	0.55	-	0.2317
	CT	0.55	-	0.2317

The *all-yes baseline* simply returns Y for all pairs. This baseline constantly returns a confidence of 1.

The *random baseline* outputs a label at random. The Accuracy shows a theoretical value of 0.5, whereas MRR and Top1 are based on an average over 10 trials.

The *QA system baseline* shows original scores from the QA systems. This baseline is very strong to beat since it's from one of the best runs submitted to the NTCIR-6 CLQA task. A caveat being that the QA systems used much richer information and techniques to rank final answers, e.g. redundancy of extracted answers, retrieval scores, extraction confidence scores, answer candidate type checking confidence, joint learning-to-rank scores (rather than independent confidence scores) etc.

Additionally, the *oracle* score indicates the upper bound by the perfect system (which is simulated by hand). In order to achieve this score, a system has to be able to rank one of correct answer candidates, if exists, to be the first without ties.

4. FORMAL RUN RESULTS

Table 6. Number of submissions.

Subtask	Language			Total
	JA	CS	CT	
BC	24	33	32	89
MC	10	27	22	59
Entrance Exam	18	-	-	18
RITE4QA	13	17	16	46
Total	65	77	70	212

Table 7. Active participants.

	Team ID	Organization	Country/ Region	Language		
				JA	CS	CT
1	FudanNLP	Fudan University	China		✔	
2	FX	Fuji Xerox	Japan	✔		
3	IASLD	Academia Sinica	Taiwan		✔	✔
4	IBM	IBM Research – Tokyo / Preferred Infrastructure	Japan	✔		
5	ICL	Key Laboratory of Computational linguistics, Peking University / Ministry of Education	China		✔	
6	ICRC_HITSZ	Intelligence Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School	China		✔	✔
7	III_CYUT_ NTHU	Institute for Information Industry / Chaoyang University of Technology / National Tsing Hua University	Taiwan			✔
8	IMTKU	Information Management, Tamkang University	Taiwan			✔
9	JAIST	Japan Advanced Institute of Science and Technology	Japan	✔		
10	JUCS	Jadavpur University, Computer Sc. & Engineering	India	✔		
11	KYOTO	Kyoto University (Kurohashi Laboratory)	Japan	✔		
12	LTI	Language Technologies Institute, Carnegie Mellon University	USA	✔		
13	MCU	Ming-Chuan University	Taiwan			✔
14	NSNG	Northeastern University, USA / Wuhan University	USA / China		✔	
15	NTOU	National Taiwan Ocean University	Taiwan			✔
16	NTU	National Taiwan University	Taiwan		✔	✔
17	NTTCS	Nippon Telegraph and Telephone Corporation	Japan	✔		
18	SITLP	Shibaura Institute of Technology LP lab	Japan	✔		
19	TU	Tohoku University	Japan	✔		
20	UIOWA	University of Iowa	USA		✔	✔
21	WHUTE	Wuhan University	China		✔	
22	WUST	Wuhan University of Science and Technolog	China		✔	
23	Yuntech	National Yunlin University of Science and Technology	Taiwan		✔	✔
24	ZSWSL	Beijing University of Posts and Telecommunications	China		✔	

Table 8. Evaluation result on BC subtask (JA).

Run	Accuracy
JAIST-JA-BC-01	0.5800
JAIST-JA-BC-02	0.5660
JAIST-JA-BC-03	0.5520
NTTCS-JA-BC-03	0.5480
LTI-JA-BC-03*	0.5460
LTI-JA-BC-02*	0.5420
LTI-JA-BC-01*	0.5340
NTTCS-JA-BC-01	0.5320
IBM-JA-BC-02*	0.5260
FX-JA-BC-02	0.5240
FX-JA-BC-03	0.5200
NTTCS-JA-BC-02	0.5200
IBM-JA-BC-01*	0.5160
KYOTO-JA-BC-02	0.5160
KYOTO-JA-BC-03	0.5160
SITLP-JA-BC-01	0.5160
SITLP-JA-BC-02	0.5120
FX-JA-BC-01	0.5100
JUCS-JA-BC-03	0.5080
IBM-JA-BC-03*	0.5000
JUCS-JA-BC-02	0.5000
SITLP-JA-BC-03	0.4940
KYOTO-JA-BC-01	0.4920
JUCS-JA-BC-01	0.4900
Baseline (char overlap)	*0.5160*

Table 9. Evaluation result on BC subtask (CS).

Run	Accuracy
ICRC_HITSZ-CS-BC-03	0.7764
FudanNLP-CS-BC-02	0.7617
ICRC_HITSZ-CS-BC-02	0.7568
FudanNLP-CS-BC-01	0.7469
WHUTE-CS-BC-03	0.7371
NTU-CS-BC-01	0.7346
WHUTE-CS-BC-02	0.7322
WUST-CS-BC-01	0.7248
NTU-CS-BC-02	0.7224
NTU-CS-BC-03	0.7199
ZSWSL-CS-BC-01	0.7199
IASLD-CS-BC-01*	0.7150
ICL-CS-BC-01	0.7150
WHUTE-CS-BC-01	0.7125
ICL-CS-BC-02	0.7101
ICRC_HITSZ-CS-BC-01	0.7076
IASLD-CS-BC-02*	0.7052
IASLD-CS-BC-03*	0.6880
III_CYUT_NTHU-CS-BC-02	0.6830
NSNG-CS-BC-02	0.6683
ZSWSL-CS-BC-02	0.6658
NSNG-CS-BC-01	0.6536
Yuntech-CS-BC-01	0.6364
NSNG-CS-BC-03	0.5897
ZSWSL-CS-BC-03	0.5897
Yuntech-CS-BC-02	0.5602
III_CYUT_NTHU-CS-BC-01	0.5577
III_CYUT_NTHU-CS-BC-03	0.5577
Baseline (char overlap)	*0.7617*
UIOWA-CS-BC-01 ‡	*0.9705*
UIOWA-CS-BC-03 ‡	*0.9631*
UIOWA-CS-BC-02 ‡	*0.9361*

Table 10. Evaluation result on BC subtask (CT).

Run	Accuracy
IASLD-CT-BC-03	0.6611
IASLD-CT-BC-02	0.6533
III_CYUT_NTHU-CT-BC-02	0.6500
IASLD-CT-BC-01	0.6478
NTOUA-CT-BC-02*	0.6422
ICRC_HITSZ-CT-BC-01	0.6133
NTOUA-CT-BC-01*	0.6133
NTU-CT-BC-01	0.6078
NTU-CT-BC-03	0.6067
NTOUA-CT-BC-03*	0.6022
ICRC_HITSZ-CT-BC-02	0.5967
NTU-CT-BC-02	0.5956
III_CYUT_NTHU-CT-BC-01	0.5733
III_CYUT_NTHU-CT-BC-03	0.5733
IMTKU-CT-BC-02	0.5556
MCU-CT-BC-01	0.5544
IMTKU-CT-BC-01	0.5500
Yuntech-CT-BC-01	0.5278
IMTKU-CT-BC-03	0.5244
Yuntech-CT-BC-02	0.5244
Baseline (char overlap)	*0.6667*
UIOWA-CT-BC-01 ‡	*0.9078*
UIOWA-CT-BC-02 ‡	*0.8844*

Table 11. Evaluation result on MC subtask (JA).

Run	Accuracy
IBM-JA-MC-02*	0.5114
KYOTO-JA-MC-03	0.4841
KYOTO-JA-MC-02	0.4795
IBM-JA-MC-01*	0.4545
NTTCS-JA-MC-03	0.4523
NTTCS-JA-MC-01	0.4477
IBM-JA-MC-03*	0.4455
NTTCS-JA-MC-02	0.4045
KYOTO-JA-MC-01	0.2136
JUCS-JA-MC-01	0.1750
Baseline (char overlap)	*0.4682*

Table 12. Evaluation result on MC subtask (CS).

Run	Accuracy
ICRC_HITSZ-CS-MC-03	0.6413
ICRC_HITSZ-CS-MC-02	0.6241
ZSWSL-CS-MC-02	0.6192
WHUTE-CS-MC-02	0.6093
III_CYUT_NTHU-CS-MC-02	0.5897
FudanNLP-CS-MC-02	0.5848
WHUTE-CS-MC-01	0.5823
WUST-CS-MC-01	0.5823
FudanNLP-CS-MC-01	0.5799
ICRC_HITSZ-CS-MC-01	0.5749
NTU-CS-MC-02	0.5749
NTU-CS-MC-03	0.5700
IASLD-CS-MC-01*	0.5651
NTU-CS-MC-01	0.5651
ZSWSL-CS-MC-03	0.5627
IASLD-CS-MC-03*	0.5553
ZSWSL-CS-MC-01	0.5455
IASLD-CS-MC-02*	0.5430
III_CYUT_NTHU-CS-MC-01	0.5332
III_CYUT_NTHU-CS-MC-03	0.5307
Yuntech-CS-MC-01	0.5283
ICL-CS-MC-01	0.5061
ICL-CS-MC-02	0.4840
Yuntech-CS-MC-02	0.3980
Baseline (char overlap)	*0.5315*
UIOWA-CS-MC-01 ‡	*0.8919*
UIOWA-CS-MC-02 ‡	*0.8919*
UIOWA-CS-MC-03 ‡	*0.8870*

Table 13. Evaluation result on MC subtask (CT).

Run	Accuracy
MCU-CT-MC-01	0.5356
IMTKU-CT-MC-01	0.5222
IMTKU-CT-MC-02	0.5067
IASLD-CT-MC-03	0.5011
IASLD-CT-MC-01	0.4989
ICRC_HITSZ-CT-MC-01	0.4967
III_CYUT_NTHU-CT-MC-02	0.4911
IASLD-CT-MC-02	0.4867
NTU-CT-MC-03	0.4833
Yuntech-CT-MC-01	0.4767
NTOUA-CT-MC-02*	0.4611
NTU-CT-MC-01	0.4589
NTU-CT-MC-02	0.4578
NTOUA-CT-MC-01*	0.4400
III_CYUT_NTHU-CT-MC-03	0.4333
III_CYUT_NTHU-CT-MC-01	0.4300
NTOUA-CT-MC-03*	0.4211
Yuntech-CT-MC-02	0.3878
IMTKU-CT-MC-03	0.2678
Baseline (char overlap)	*0.4885*
UIOWA-CT-MC-01 ‡	*0.7867*
UIOWA-CT-MC-02 ‡	*0.7744*
UIOWA-CT-MC-03 ‡	*0.7244*

**** IASLD, IBM, LTI, and NTOUA include RITE organizer(s) in a team. They paid full attention to fairly participate in the formal run.***

***** Evaluated only on pairs where a label is returned.***

‡ Manual runs, in which the synonym list used by the system is manually enhanced based on BC and MC training and test sets.

Table 14. Evaluation result on Entrance Exam subtask (JA).

Run	Accuracy
IBM-JA-EXAM-01	0.7217
TU-JA-EXAM-02**	0.7183
TU-JA-EXAM-03**	0.7042
IBM-JA-EXAM-02	0.6742
LTI-JA-EXAM-03	0.6674
KYOTO-JA-EXAM-02	0.6561
KYOTO-JA-EXAM-03	0.6561
LTI-JA-EXAM-02	0.6538
JAIST-JA-EXAM-02	0.6516
JAIST-JA-EXAM-03	0.6516
TU-JA-EXAM-01	0.6493
JAIST-JA-EXAM-01	0.6222
LTI-JA-EXAM-01	0.6018
KYOTO-JA-EXAM-01	0.5928
IBM-JA-EXAM-03	0.5837
JUCS-JA-EXAM-01	0.5204
TU-JA-EXAM-02	0.1154
TU-JA-EXAM-03	0.1131
Baseline (char overlap)	*0.6516*

Table 15. Evaluation result on RITE4QA subtask (JA).

Run	Accuracy	Top1	MRR
LTI-JA-RITE4QA-03*	0.6753	0.2136	0.2982
JAIST-JA-RITE4QA-01	0.5602	0.1802	0.2765
JAIST-JA-RITE4QA-03	0.6940	0.1658	0.2731
JAIST-JA-RITE4QA-02	0.6763	0.1508	0.2604
LTI-JA-RITE4QA-02*	0.6411	0.1743	0.2563
JUCS-JA-RITE4QA-01	0.5954	0.1315	0.2490
KYOTO-JA-RITE4QA-02	0.6836	0.1206	0.2344
KYOTO-JA-RITE4QA-03	0.6836	0.1206	0.2344
IBM-JA-RITE4QA-01*	0.3330	0.1131	0.2327
IBM-JA-RITE4QA-03*	0.4015	0.0871	0.2221
LTI-JA-RITE4QA-01*	0.8434	0.1265	0.2220
IBM-JA-RITE4QA-02*	0.3164	0.0905	0.2168
KYOTO-JA-RITE4QA-01	0.8890	0.1168	0.1752
Baseline1 (char overlap)	*0.4180*	*0.2337*	*0.3192*
Baseline2 (all yes)	*0.1100*	*0.1077*	*0.1657*
Baseline3 (random)	*0.5000*	*0.1025*	*0.2320*
Baseline4 (QA system)	*0.1100*	*0.3350*	*0.3917*
Oracle	*1.0000*	*0.5326*	*0.5326*

Table 16. Evaluation result on RITE4QA subtask (CS). See the Table 17 for the baseline scores.

Run	Accuracy	Top1	MRR
IMTKU-CS-RITE4QA-02	0.4090	0.2953	0.3998
WHUTE-CS-RITE4QA-02	0.4876	0.2852	0.3979
WHUTE-CS-RITE4QA-01	0.3886	0.2651	0.3773
IMTKU-CS-RITE4QA-03	0.4716	0.2550	0.3768
IMTKU-CS-RITE4QA-01	0.3319	0.2450	0.3744
ICL-CS-RITE4QA-01	0.3231	0.2931	0.3545
ICRC_HITSZ-CS-RITE4QA-01	0.6390	0.2479	0.3520
WHUTE-CS-RITE4QA-03	0.3275	0.2248	0.3494
ICRC_HITSZ-CS-RITE4QA-03	0.7293	0.2262	0.3398
IASLD-CS-RITE4QA-01*	0.4833	0.2274	0.3028
IASLD-CS-RITE4QA-02*	0.4803	0.2274	0.3028
III_CYUT_NTHU-CS-RITE4QA-01	0.7525	0.2585	0.2944
III_CYUT_NTHU-CS-RITE4QA-02	0.7162	0.2408	0.2908
ICRC_HITSZ-CS-RITE4QA-02	0.6128	0.2234	0.2705
IASLD-CS-RITE4QA-03*	0.4352	0.2310	0.2608
III_CYUT_NTHU-CS-RITE4QA-03	0.3377	0.2320	0.2527
UIOWA-CS-RITE4QA-01 ‡	*0.9010*	*0.4559*	*0.4272*

Table 17. Evaluation result on RITE4QA subtask (CT).

Run	Accuracy	Top1	MRR
IMTKU-CT-RITE4QA-03	0.4003	0.2953	0.3992
NTOUA-CT-RITE4QA-03*	0.6346	0.2813	0.3824
NTOUA-CT-RITE4QA-01*	0.5459	0.2746	0.3803
IMTKU-CT-RITE4QA-01	0.3246	0.2517	0.3772
IMTKU-CT-RITE4QA-02	0.3392	0.2517	0.3736
NTOUA-CT-RITE4QA-02*	0.5124	0.2282	0.3572
ICRC_HITSZ-CT-RITE4QA-01	0.6390	0.2479	0.3520
ICRC_HITSZ-CT-RITE4QA-03	0.7293	0.2262	0.3398
IASLD-CT-RITE4QA-01*	0.4760	0.2274	0.3016
IASLD-CT-RITE4QA-02*	0.4731	0.2274	0.3016
III_CYUT_NTHU-CT-RITE4QA-01	0.7525	0.2598	0.2947
III_CYUT_NTHU-CT-RITE4QA-02	0.7147	0.2408	0.2908
ICRC_HITSZ-CT-RITE4QA-02	0.6128	0.2234	0.2705
IASLD-CT-RITE4QA-03*	0.4279	0.2290	0.2619
III_CYUT_NTHU-CT-RITE4QA-03	0.3392	0.2320	0.2527
Baseline1 (char overlap)	*0.2317*	*0.2317*	*0.3844*
Baseline2 (all yes)	*0.1906*	*0.2243*	*0.2378*
Baseline3 (random)	*0.5000*	*0.2109*	*0.3454*
Baseline4 (QA system)	*0.1906*	*0.4200*	*0.4852*
Oracle	*1.0000*	*0.5906*	*0.5906*
UIOWA-CT-RITE4QA-01 ‡	*0.9010*	*0.4559*	*0.4272*

* ***IASLD, IBM, LTI, and NTOUA include RITE organizer(s) in a team. They paid full attention to fairly participate in the formal run.***

** ***Evaluated only on pairs where a label is returned.***

‡ ***Manual runs, in which the synonym list used by the system is manually enhanced based on BC and MC training and test sets.***

5. DISCUSSION

Because the Traditional Chinese test set was created from two different sources and the first source was the same as the CT-MC development set, we would like to see the impact of the genre of the text. The CT-*-Set1 contains the first 671 pairs (in CT-BC and MC test sets) which came from the previous QA tasks, and the CT-*-Set2 contains the last 229 pairs coming from the previous IR data. Table 18 and Table 19 illustrate the evaluation results in these two subsets. Note that only results produced by fully-automatic systems are listed in the tables.

As expected, most systems perform better in CT-*-Set1 than in CT-*-Set2, indicating that most systems were built by using the development set. However, some systems achieve better performance in CT-*-Set2. It would be interesting to see what strategies have made these systems more robust.

6. CONCLUSION

This paper described an overview of the NTCIR-9 RITE task. We built large-scale reusable evaluation datasets for four kinds of subtasks.

Considering this was the first attempt to conduct a set of new challenge problems, we think the RITE task was successful, with 24 participating teams from 5 different countries, who submitted 212 runs in total. Although the evaluation results may not show good enough scores to indicate that the community is ready to declare a victory in textual entailment problem, we did make the very meaningful first step in establishing a state-of-the-art.

Table 18. More Evaluation on BC subtask (CT).

Run	CT-BC-Set1	CT-BC-Set2
IASLD-CT-BC-03	0.645	0.707
IASLD-CT-BC-02	0.666	0.616
III_CYUT_NTHU-CT-BC-02	0.668	0.598
IASLD-CT-BC-01	0.650	0.642
NTOUA-CT-BC-02*	0.653	0.611
ICRC_HITSZ-CT-BC-01	0.633	0.555
NTOUA-CT-BC-01*	0.645	0.520
NTU-CT-BC-01	0.644	0.502
NTU-CT-BC-03	0.633	0.528
NTOUA-CT-BC-03*	0.629	0.524
ICRC_HITSZ-CT-BC-02	0.644	0.459
NTU-CT-BC-02	0.641	0.463
III_CYUT_NTHU-CT-BC-01	0.577	0.563
III_CYUT_NTHU-CT-BC-03	0.577	0.563
IMTKU-CT-BC-02	0.574	0.502
MCU-CT-BC-01	0.586	0.463
IMTKU-CT-BC-01	0.571	0.489
Yuntech-CT-BC-01	0.534	0.511
IMTKU-CT-BC-03	0.565	0.406
Yuntech-CT-BC-02	0.519	0.541

Table 19. More Evaluation on MC subtask (CT).

Run	CT-MC-Set1	CT-MC-Set2
MCU-CT-MC-01	0.586	0.389
IMTKU-CT-MC-01	0.559	0.415
IMTKU-CT-MC-02	0.534	0.428
IASLD-CT-MC-03	0.520	0.445
IASLD-CT-MC-01	0.522	0.432
ICRC_HITSZ-CT-MC-01	0.542	0.362
III_CYUT_NTHU-CT-MC-02	0.537	0.358
IASLD-CT-MC-02	0.505	0.432
NTU-CT-MC-03	0.519	0.380
Yuntech-CT-MC-01	0.519	0.354
NTOUA-CT-MC-02*	0.475	0.419
NTU-CT-MC-01	0.510	0.310
NTU-CT-MC-02	0.511	0.301
NTOUA-CT-MC-01*	0.478	0.328
III_CYUT_NTHU-CT-MC-03	0.469	0.328
III_CYUT_NTHU-CT-MC-01	0.463	0.332
NTOUA-CT-MC-03*	0.455	0.323
Yuntech-CT-MC-02	0.377	0.419
IMTKU-CT-MC-03	0.301	0.170

7. REFERENCES

[1] Harabagiu, Sanda, and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL.

[2] Bogdan, Sacaleanu, Constantin Orasan, Christian Spurk, Shiyan Ou, Óscar Ferrández, Milen Kouylekov, Matteo Negri. 2008. Entailment-based Question Answering for Structured Data. In Proceedings of COLING '08 22nd International Conference on on Computational Linguistics: Demonstration Papers. Manchester, UK.

[3] Parapar, David, Alvaro Barreiro, David E. Losada. 2005. Query expansion using wordnet with a logical model of information retrieval. IADIS AC 2005: 487-494

[4] Clinchant, Stéphane, Cyril Goutte and Eric Gaussier. 2006. Lexical Entailment for Information Retrieval. Lecture Notes in Computer Science, 2006, Volume 3936/2006, 217-228.

[5] Kouylekov, Milen O. 2006. Recognizing Textual Entailment with Tree Edit Distance: Application to Question Answering and Information Extraction. PhD Thesis at University of Trento, Computer Science, Electronics and Telecomunication.

[6] Lloret, Elena, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. 2008. A text summarization approach under the influence of textual entailment. In Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2008).

[7] Tatar, Doina, Andreea Diana Mihis, Dana Lupsa, Emma Tamaianu-Morita. 2009. Entailment-Based Linear Segmentation in Summarization. International Journal of Software Engineering and Knowledge Engineering 19(8): 1023-1038 (2009)

[8] Nielsen, Rodney D., Wayne Ward and James H. Martin. 2009. Recognizing entailment in intelligent tutoring systems. Natural Language Engineering, Volume 15 Issue 4, October 2009.

[9] Padó, Sebastian, Michel Galley, Dan Jurafsky, and Christopher D. Manning. 2009. Robust Machine Translation Evaluation with Entailment Features. In Proceedings of ACL-IJCNLP ’09.

[10] Zhou, Liang, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. ParaEval: Using Paraphrases to Evaluate Summaries Automatically. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics.

[11] Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. “The pascal recognising textual entailment challenge”, In Quiñonero-Candela, J.; Dagan, I.; Magnini, B.;d'Alché-Buc, F. (Eds.), Machine Learning Challenges. Lecture Notes in Computer Science, Vol. 3944, pp. 177-190, Springer, 2006.

[12] Bar-Haim, Roy, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, and Bernardo Magnini. 2006. The second pascal recognising textual entailment challenge. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment.

[13] Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing.

[14] Giampiccolo, Danilo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2009. The Fourth PASCAL Recognizing Textual Entailment Challenge. In Proceedings of TAC 2008 Workshop.

[15] Bentivogli, Luisa, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In Proceedings of TAC 2009 Workshop.

[16] Bos, Johan, Fabio Massimo Zanzotto, and Marco Pennacchiotti. 2009. Textual Entailment at EVALITA 2009. In Proceedings of EVALITA 2009.

[17] Mehdad, Yashar, Matteo Negri, and Marcello Federico. 2010. Towards Cross-Lingual Textual entailment. In

Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2010. LA, USA.

[18] Negri, Matteo, and Yashar Mehdad. 2010. Creating a Bi-lingual Entailment Corpus through Translations with Mechanical Turk: $100 for a 10-day Rush. In Proceedings of the NAACL-HLT 2010, Creating Speech and Text Language Data With Amazon's Mechanical Turk Workshop. LA, USA.

[19] Mehdad, Yashar, Matteo Negri, Marcello Federico. 2011. Using Bilingual Parallel Corpora for Cross-Lingual Textual Entailment. In Proceedings of ACL 2011.

[20] Inui, Kentaro, Koichi Takeuchi, and Atsushi Fujita. 2007. Building an Event Ontology for Textual Entailment Computation. IEIC Technical Report (Institute of Electronics, Information and Communication Engineers) VOL.106; NO. 518(NLC2006 87-92); PAGE. 13-18; (20070124); In Japanese.

[21] Matsuyoshi, Suguru, Koji Murakami, Yuji Matsumoto, Kentaro Inui. 2008. A Database of Relations between Predicate Argument Structures for Recognizing Textual Entailment and Contradiction. In Proceedings of ISUC 2008. pp.366~373. (In Japanese)

[22] Umemoto, Hiroshi, Sugihara Daigo, Ohkuma Tomoko, Masuichi Hiroshi. 2008. Detecting Japanese Textual Entailment Using LFG Analysis and Lexical Resources. IPSJ SIG Notes 2008(113), 57-64, 2008-11-19. (In Japanese)

[23] Muramatsu, Yuki, Kunihiro Udaka, and Kazuhide Yamamoto. 2010. Textual Entailment Recognition using Word Overlap, Mutual Information and Subpath Set. In Proceedings of the Second Workshop on Cognitive Aspects of the Lexicon: Enhancing the Structure and Lookup Mechanisms of Electronic Dictionaries (COGALEX-II), pp.18-27 (2010.8). (In Japanese)

[24] Odani, Michitaka, Tomohide Shibata, Sadao Kurohashi, Takayuki Nakata. 2008. Building data of japanese Text Entailment and recognition of inferencing relation based on automatic achieved similar expression. In Proceeding of 14th Annual Meeting of the Association for "Natural Language Processing, pp. 1140-1143, 2008 (in Japanese)

[25] Kasahara, Kaname, Hirotoshi Taira and Masaaki Nagata. 2010. Consider of the possibility Textual Entailment applied to Reading Comprehension Task consisted of multi documents. In Proceeding of 14th Annual Meeting of the Association for Natural Language Processing, pp. 780-783, 2010 (in Japanese)

[26] Peñas, Anselmo, Álvaro Rodrigo, Valentin Sama, and Felisa Verdejo. 2006. Overview of the answer validation exercise 2006. In Proceedings of the 7th Cross-language evaluation forum conference on evaluating systems for multilingual and multimodal information access.

[27] Peñas, Anselmo, Álvaro Rodrigo, and Felisa Verdejo. 2008. Overview of the Answer Validation Exercise 2007. CLEF 2007, Lecture Notes in Computer Science LNCS 5152. Springer, Berlin.

[28] Rodrigo, Álvaro, Anselmo Peñas, and Felisa Verdejo. 2009. Overview of the answer validation exercise 2008. In Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access.

[29] Mitamura, Teruko, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji and Noriko Kando. 2008. *Overview of the NTCIR-7 ACLIA: Advanced Cross-Lingual Information Access.* In Proceedings of NTCIR-7 Workshop, Japan.

[30] Mitamura, Teruko, Hideki Shima, Tetsuya Sakai, Noriko Kando, Tatsunori Mori, Koichi Takeda, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, and Cheng-Wei Lee. 2010. Overview of the NTCIR-8 ACLIA Tasks: Advanced Cross-Lingual Information Access. In Proceedings of NTCIR-8 Workshop, Japan.

[31] Condoravdi, Cleo, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel G Bobrow. 2003. Entailment, Intensionality and Text Understanding. In Proceedings of the Workshop on Text Meaning.

[32] Marneffe, Marie-catherine De, Anna N Rafferty, and Christopher D Manning. 2008. Finding Contradictions in Text. Computational Linguistics, no. June: 1039-1047.

[33] Chen, Chaomei, Fidelia Ibekwe-SanJuan, Eric SanJuan, and Chris Weaver. 2006. Visual Analysis of Conflicting Opinions. 2006 IEEE Symposium On Visual Analytics And Technology: 59-66.

[34] Harabagiu, Sanda, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast and contradiction in text processing. Proceedings of the 21st national conference on Artificial intelligence: 755-762.

[35] Sasaki, Yutaka, Chuan-Jie Lin, Kuang-hua Chen, Hsin-His Chen. 2007. Overview of the NTCIR-6 Cross-Lingual Question Answering (CLQA) Task. In Proceedings of NTCIR-6 Workshop, Japan.

[36] Cleverdon, Cyril William. The significance of the Cranfield tests on index languages. In Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Researchand Development in Information Retrieval, pages 3–12, 1991.

[37] Voorhees, Ellen. 2002, The Philosophy of Information Retrieval Evaluation. In Proceedings of the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems, Springer-Verlag, Berlin, Heidelberg , pp. 355-370 .

[38] Asahara, Masayuki, and Yuji. Matsumoto. 2003. Japanese Named Entity Extraction with Redundant Morphological Analysis. In Proceedings of NAACL/HLT 2003, 2003.

[39] Li, Maoxi, Chengqing Zong and Hwee Tou Ng. 2011. Automatic Evaluation of Chinese Translation Output: Word-level or character-level? In Proceedings of ACL 2011.

[40] Papineni, Kishore, Salim Roukos, Todd Ward, Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. ACL 2002: 311-318