IMTKU Question Answering System for World History Exams at NTCIR-13 QA Lab-3

Min-Yuh Day
Tamkang University
myday@mail.tku.edu.tw

Chao-Yu Chen
Tamkang University
susan.cy.chen@gmail.com

Wan-Chu Huang
Tamkang University
k92307@gmail.com

I-Hsuan Huang
Tamkang University
q7983451@gmail.com

Shih-Ya Cheng
Tamkang University
butterfly60698@gmail.com

Tz-Rung Chen
Tamkang University
ivy1000816@gmail.com

Min-Chun Kuo
Tamkang University
whane601@gmail.com

Yue-Da Lin
Tamkang University
simon08074@gmail.com

Yi-Jing Lin
Tamkang University
pkjack9504@gmail.com

ABSTRACT

This paper describes the IMTKU (Information Management at Tamkang University) question answering system for world history exams in Japanese university entrance exams at NTCIR-13 QA Lab-3. The IMTKU team proposed a question answering system that integrates natural language processing with deep learning approach for Japanese university entrance exams at NTCIR-13 QA Lab-3. In QA Lab-3 phase-2, the IMTKU team submitted 3 English End-to-End multiple-choice run results, 2 English End-to-End essay run results, 2 Japanese End-to-End essay run results, 2 English extraction essay run results, 2 Japanese extraction essay run results, 1 English summarization essay run result, and 1 Japanese summarization essay run result for National Center Tests and Second-stage Examinations. The best total score of IMTKU QA system is 40 in English multiple-choice subtask phase-3 and the best score is 0.408 for the complex Japanese essay subtask at NTCIR-13 QA Lab-3.

Team Name

IMTKU

Subtasks

QA Lab-3 (National Center Exams English Version, Secondary Exams English Version, National Center Exams Japanese Version)

Keywords

IMTKU, NTCIR 13, QA Lab-3, Question Answering, University Entrance Examination

1. INTRODUCTION

The IMTKU team participated in NTCIR-13 QA Lab-3 National Center Test for University Admissions and Secondary exams in Japanese and English version from Japanese university entrance exams. NTCIR-13 QA Lab-3 totally has two phases and a research run, the English and Japanese subtask will be done in two phases and a research run. In Phase-2, we submitted 3 English End-to-End multiple-choice run results, 2 English End-to-End essay run results, 2 Japanese End-to-End essay run results, 2 English extraction essay run results, 2 Japanese extraction essay run results, 1 English summarization essay run result, and 1 Japanese summarization essay run result for National Center Tests and Second-stage Examinations. This paper describes the tools and resources used in IMTKU question answering system for world history exams at NTCIR-13 QA Lab-3.

Question Answering (QA) is a CLEF/TREC task of solving and evaluating an answer for a given question, which is widely applied for many languages such as English, French, Japanese, Chinese, etc. QA-Lab provides a module platform for answering real-world entrance exam questions at NTCIR-11. [1, 3]

QA-Lab is designed to solve real-world Japanese university entrance exam questions for world history. The world history questions are used from The National Center Test for University Admissions and Secondary exams, which include Japanese and English translated version. The question types of NTCIR-11 QALab1 are True/False questions, factoid questions, and a number of questions with short answer of Japanese characters. [1, 3]

QA-Lab provided the module structure of the original QA system. The QA-Lab architecture consists of 4 modules, question analysis, document retrieval, answer extraction, and answer generation. [3] Question analysis module analyzes the types of questions and extracts the question format. Document retrieval module focuses on searching related documents. Answer extraction module extracts answer candidates from the output of document retrieval module, which is retrieved documents or passages. Answer generation module ranks the answer candidates based on the ranking score. [3]

NTCIR-12 QA Lab-2 subtask is slightly different from NTCIR-11 QA-Lab1. In NTCIR-12 QA Lab-2, the organizers defined six question types such as Complex Essay (CE), Simple Essay (SE), Factoid (F), Slot-Filling (SF), True-or-False (TF) and Unique (U). [2]

NTCIR-13 QALab-3 is considered the results from NTCIR-11 QA-Lab1 and NTCIR-12 QALab-2, and improve the architecture of basic system. The basic system is divided into three part, such as term questions, multiple-choice questions, and essay questions. Especially the part of essay questions is divided into End-to-End, extraction, summarization, and evaluation-method subtask. [7]

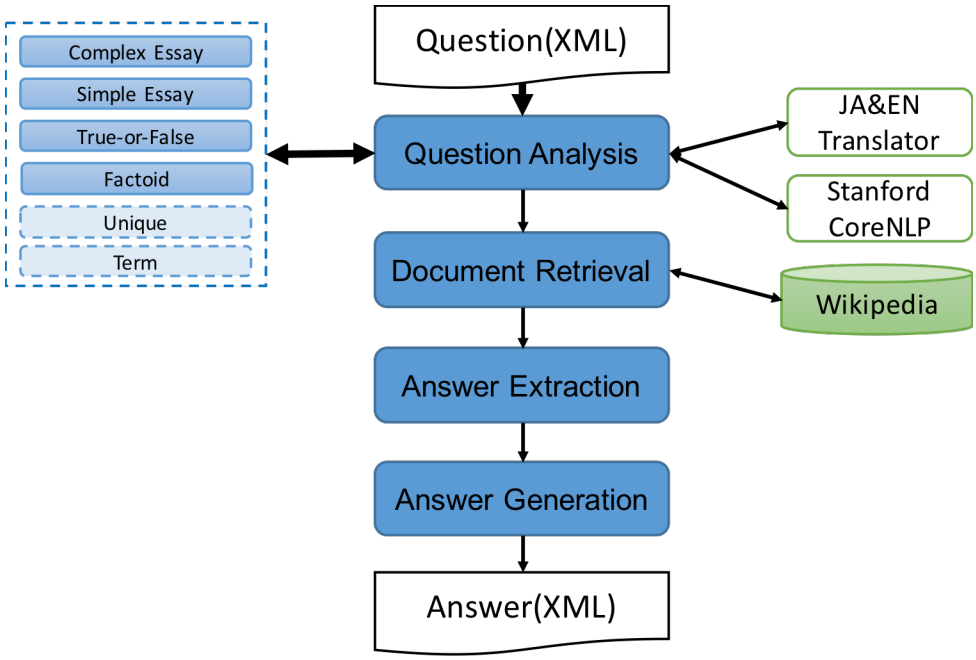

Figure 1. System architecture of IMTKU question answering system for world history exams at NTCIR-13 QA Lab-3

The evaluation uses scores from National Center for University Admissions and other universities with multiple-choice questions. For term questions, the evaluation is using the accuracy by exact matching with the gold standard data. Complex Essay and Simple Essay evaluation uses various of human expert marks, ROUGE method, pyramid method, and the quality questions [7].

The remainder of the paper is organized as follows. Section 2 describes the system architecture of IMTKU question answering system. Section 3 details the experimental results and analysis. Finally, we present discussions in Section 4 and conclude our work in Section 5.

2. SYSTEM ARCHITECTURE

The system architecture of IMTKU question answering system for NTCIR-13 QA Lab-3 is presented in Figure 1. There are four main processes which are question analysis, document retrieval, answer extraction, and answer generation. In question analysis, JA&EN translator and Stanford CoreNLP are used for recognizing the type of each question and extracting the keywords in question. After the system generate the keyword list, the list is put in the Document Retrieval module and receive the content list as the output. In the next step, we extract the passages from the content list, and then use some methods to generate answers.

2.1 Question Analysis

2.1.1 XML dataset extraction

In essay subtask, the XML files from Phase-2 are analyzed for understanding the type of questions and answering more correctly. Table 1 shows the analysis of essay from XML files. It is divided into two types of essay. One is simple essay which should be answered with a short summary in a range from 5 to 60 words in English subtask and in a range from 10 to 120 words in Japanese subtask. The relevant keywords could be obtained in the tag <introduction> of simple essay. The other type is complex essay including special keywords that should be answered with a longer summary in a range from 255 to 300 words in English subtask and in a range from 450 to 600 words in Japanese subtask. The set of special keywords could be gotten in the tag <keyword_set> of complex essay. The total numbers of questions of each type in Phase-2 are also analyzed and showed in Table 1.

Table 1. The analysis of essay from XML

Type of Essay	Simple Essay	Complex Essay
Content length by word (EN)	5, 15, 30, 45, 60	255, 270, 300
Content length by word (JA)	10, 30, 60, 90, 120	450, 510, 540, 600
Tag of exacting keywords	<instruction>	<keyword_set>
# of questions	22	5

To analyze every question in multiple-choice and the introduction of each essay question for extracting keywords, the system used NLTK to tokenize the questions and get the POS tag for each token. NLTK is a leading platform for building Python programs to work with human language data. Figure 2 shows an example how the questions could be presented after NLTK was applied on the sentence. Then stop-words were removed from the question to generate the most relevant words as keywords. Using these keywords from the question were taken as search terms for document retrieval from Wikipedia.

```
The Uighurs destroyed the Kyrgyz.
tokens>>> [ 'The' , 'Uighurs' , 'destroyed' , 'the' , 'Kyrgyz' , '.' ]
tagged>>> [('The' , 'DT') , ('Uighurs' , 'NNP') , ('destroyed','VBD') ,
           ('the' , 'DT') ,('Kyrgyz' , 'NNP') , ('.', '.')]
```

Figure 2. An example for using NLTK and POS tagger on the question

2.1.2 POS Tagger

Stanford POS Tagger is a part-of-speech tagging tool, the function is giving part-of-speech of different words (e.g., verb, noun, adjective) after it reads articles or sentences. Stanford POS Tagger supports other computer languages. [5, 6]

IMTKU system used Stanford POS Tagger to analyze the part-of-speech of words in a question. Stanford POS Tagger could divide the part-of-speech of words into thirty-six kinds. After analyze questions, mark the consequence as a label behind each word.

2.1.3 Name Entity Recognition

Stanford CoreNLP also includes Stanford NER. Stanford NER was used to analyze the named-entity of words in a question. There is a classifier used in NER and are also many kinds of classifiers for user to choose. One version of the classifier, which can separate into seven kinds of category are location, organization, date, money, person, percent, and time, was applied in IMTKU question answering system.

2.1.4 JA&EN Translator

Figure 3 shows an example of JA&EN translator. Because our system is designed to dealing with English questions only, the preprocess is translating all contents into English before analyze the Japanese questions.

Japanese:

古代メソポタミアと古代エジプトにおける暦とその発達の背景について，３行以内で説明しなさい。

English (JA & EN Translator by Google Translate):

Explain the calendar in ancient Mesopotamia and ancient Egypt and the background of its development within 3 lines.

Figure 3. An Example of JA & EN Translator

Google translate is practiced in the module of JA& EN translator for Japanese center exam. The translation function was used by Python package googletrans to get the translated result and return.

2.2 Document Retrieval

We developed a document retrieval module that is supported for each subtask. Figure 4 shows the process of document retrieval. At first, created the ten-keywords list of each question whose keywords were obtained from question analysis.

The number of keywords for default is limited because we want to avoid the divergence between keywords and articles. The next step, put the keyword list into Wikipedia to search the title which is matched the keyword completely.

There are two situations would be happened after we search in Wikipedia. One is find the title successfully, and then crabs the articles from Wikipedia to make a content list which is returned as the output of this module. The other is got the error of disambiguation which in Wikipedia is a process to resolve the reflected error when the article is ambiguous. Our solution is to get the ambiguous word and to repeat the process of searching in Wikipedia that would extract the article to be a part of the content list.

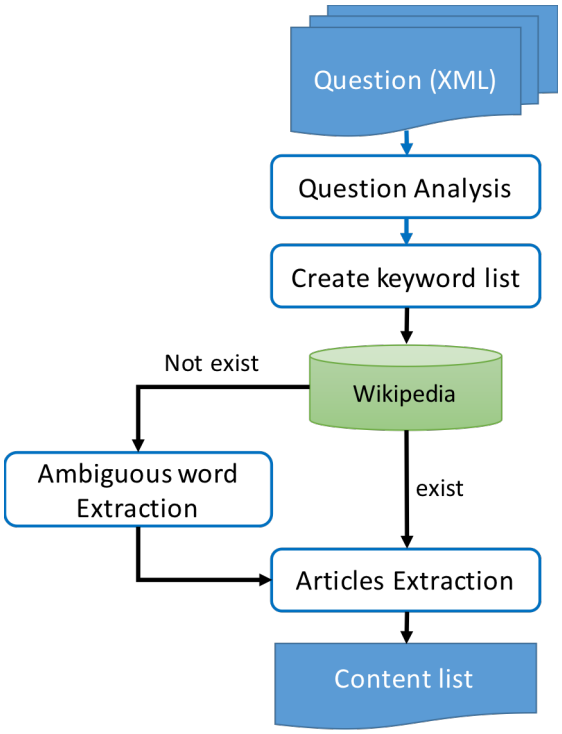

Figure 4. The process of document retrieval module

2.3 Answer Extraction

Answer Extraction is the important step in our system. Keywords would be produced while finished using NLTK and POS tagger. In essay subtask, a list by these keywords is created, and then get the article list from the Document Retrieval module. In multiple-choice subtask, TF-IDF is used to find out matrix to generate the value about each question and its choices. TF-IDF (term frequency–inverse document frequency) is the commonly method that used to weighted technology for information retrieval and information exploration. It also can evaluate the importance of a word for one of the document.

2.3.1 TF-IDF

The value of TF is a frequency that the word appears in the article.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

The value of IDF points the frequency that the word appears in all the articles.

$$idf_i = \log \frac{|D|}{|\{j: t_i \in d_j\}|}$$

When TF multiply IDF, it can remove common words and keep important words. When TF-IDF produce word frequency matrix, the result should be presented which words are the keywords in the article.

$$tfidf_{if} = tf_{ij} \times idf_i$$

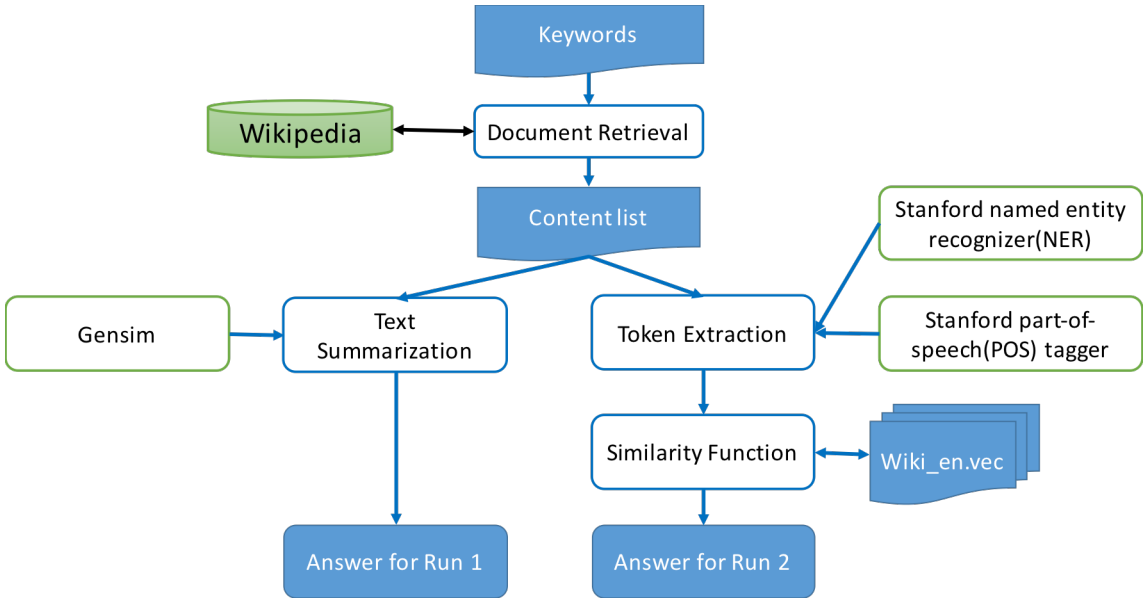

Figure 5. The process of essay subtask

2.4 Answer Generation

In essay subtask, there are two methods to generate answers. Figure 5 shows the process of essay subtask. Run 1 is using the function in Gensim which is a package of Python. The function called text summarization could generate summaries with limited words automatically. Run 2 is generating the set of the co-occurred tokens which is compared between the tokens from the article list and the vectors of English words in Wikipedia as the answer.

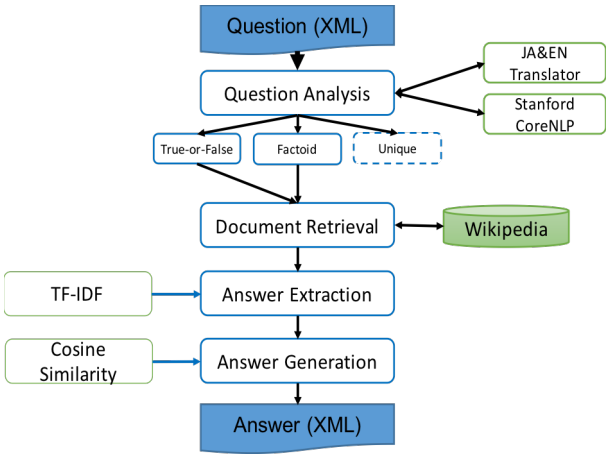

Figure 6. The process of multiple-choice subtask

In multiple-choice subtask, a machine learning method is applied by using cosine similarity for training our answer generation model. The topic and options are given, and then their cosine similarity weighted by inverse document frequency could be computed.

Figure 6 shows our multiple-choice process. At first, NLTK was used to analyze all question and delete pleonasm. Second, we used keywords searched in Document Retrieval to find the suitable article. Finally, we used both TF-IDF and cosine similarity to confirm final answer.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1 Phase-2

This section presents the results of Phase-2. In essay subtask, the results, which evaluated by ROUGE method from NTCIR13 QALab-3, would be released after submitted several runs. Table 5 shows the average ROUGE score of each run in English subtask. There are three features, case, stem, and stopword, which are used to evaluate the similarity between English summaries. In Run 1, the best scores in the simple English essay and the complex English essay are both found in the feature of stem, namely 0.077 and 0.329 respectively. Table 6 and Table 7 show the average ROUGE score of each run in Japanese subtask. There are three features, content, text, shortest unit (stem) and shortest unit (root), which are used to evaluate the similarity between Japanese summaries. The best marks in Run 1 are found with the text feature in both the simple Japanese essay and the complex Japanese essay, that is, 0.185 and 0.408 respectively.

In multiple-choice subtask, we conduct several experiments using various datasets (national Center Test for University Admissions and Secondary exams) to train and test models, as well as generate different results. The estimated correct rates of fixed choice as the answer for all questions with the dataset from 1997 to 2011 are showed in Table 2. Three runs were submitted in English subtask. Our main multiple choice module was practiced in run 3. Due to a certain degree of accuracy to use random method, the random method is applied on our run 1 and run 2.

Run 1 is a random method used to choose an answer from the choices one to four. The random selected answer was applied to all questions. Run 2 is set the choice 2 as the candidate answer for each question. Run 3 uses the module of multiple-choice showed in Figure 6. The results of multiple choice questions in Phase2 are showed in Table 3 and Table 4.

Table 2. Correct Rate of fixed choice as answer for all questions in each year

	Choice1	Choice2	Choice3	Choice4
1997	23%	23%	25%	30%
1999	24%	20%	27%	29%
2001	29%	20%	27%	22%
2003	17%	32%	29%	20%
2005	17%	25%	33%	22%
2007	19%	22%	28%	31%
2009	33%	19%	19%	28%
2011	22%	25%	28%	19%
Total	23%	23%	**27%**	25%

Table 3. Results of IMTKU multiple-choice subtask in Phase-2

Run	Language	Correct Rate	Total Score
IMTKURUN01	EN	0.333	34
IMTKURUN02	**EN**	**0.389**	**40**
IMTKURUN03	EN	0.194	18

Table 4. The number of correct answers, incorrect answers, unanswered questions and score of each IMTKU multiple-choice run in Phase-2

Run	Correct	Incorrect	Total	Total Score
IMTKURUN01	12	24	12/36	34
IMTKURUN02	**14**	**22**	**14/36**	**40**
IMTKURUN03	7	29	7/36	18

4. DISCUSSIONS

In order to evaluate the performance of IMTKU question answering system, we participated Phase-2 of QA Lab-3 subtask. We used the train dataset of center exam (1997, 2001, 2003, 2005, 2007, 2009, 2011) and secondary stage examination (Tokyo) from organizers to train and test our question answering system.

The best performance score of IMTKU question answering system in English multiple-choice subtask is 40. The run 3, which runs our main module of multiple choice in Phase-2, does not generate a high correct rate. Some question types remain to be resolved, such as Unique-image questions, Unique-time questions, and Slot-Filling questions.

In essay subtask, according to the essay of introducing ROUGE [4], ROUGE-1, ROUGE-SU4, ROUGE-SU9, ROUGE-L, and ROUGE-W are suitable for evaluating the short summarization, and ROUGE-1, ROUGE-2, ROUGE-S4, ROUGE-S9, ROUGE-SU4, and ROUGE-SU9 are suitable for evaluating the multi-document summarization. The results are showed in Table 5, Table 6 and Table 7. The highest score of each run is ROUGE-1, and the second highest score is ROUGE-S* regardless of the type of essay in English and Japanese subtask.

Generally, ROUGE-1 score and ROUGE-S* score are much higher than other scores. Due to the difference between ROUGE-1 and ROUGE-2, the similarity of one word is higher than the similarity of two words. ROUGE-S* score is also higher, which means that any pair of words in sentences has higher similarity. Based on these two reasons, the summaries developed by our system include some matched words which are compared with the gold-standard answers although the presentations of the orders of words do not have a similar pattern.

Compare run 1 and run 2, the performance of run 2 is worse than one of run 1. We have yet developed a machine learning model for essay subtask, so we just finished the set of tokens from articles and vectors of common words from Wikipedia.

Table 5. Results of IMTKU English essay subtask

SYSTEM	IMTKU1QALab3						IMTKU2QALab3					
TYPE	SIMPLE			COMPLEX			SIMPLE			COMPLEX		
METHOD	CASE	STEM	STOP	CASE	STEM	STOP	CASE	STEM	STOP	CASE	STEM	STOP
R-1	**0.075**	**0.077**	**0.026**	**0.312**	**0.329**	**0.131**	**0.006**	**0.009**	**0.012**	**0.008**	**0.014**	**0.013**
R-2	0.005	0.007	0	0.052	0.054	0.007	0	0	0	0	0	0
R-S*	**0.056**	**0.057**	**0.023**	**0.164**	**0.167**	**0.063**	**0.006**	**0.009**	**0.012**	**0.007**	**0.012**	**0.012**
R-S4	0.031	0.032	0.015	0.047	0.048	0.025	0.003	0.006	0.008	0.004	0.006	0.007
R-S9	0.007	0.007	0	0.092	0.102	0.013	0	0	0	0	0	0
R-SU*	0.007	0.008	0	0.063	0.069	0.005	0	0	0	0	0	0
R-SU4	0.008	0.009	0	0.073	0.080	0.006	0	0	0	0	0	0
R-SU9	0.009	0.010	0.001	0.094	0.104	0.015	0	0	0	0	0	0
R-L	0.018	0.019	0.003	0.105	0.113	0.027	0	0.001	0.001	0.001	0.002	0.003
R-W1.2	0.015	0.015	0.002	0.095	0.103	0.018	0	0	0	0.001	0.002	0.002

Table 6. Results of IMTKU run 1 for Japanese essay subtask

SYSTEM	IMTKU1QALab3							
TYPE	SIMPLE				COMPLEX			
METHOD	content	text	shortest unit (stem)	shortest unit (root)	content	text	shortest unit (stem)	shortest unit (root)
R-1	**0.014**	**0.185**	**0.175**	**0.180**	**0.098**	**0.408**	**0.347**	**0.352**
R-2	0	0.052	0.040	0.041	0.002	0.164	0.109	0.113
R-S*	**0.006**	**0.147**	**0.150**	**0.144**	**0.070**	**0.354**	**0.317**	**0.308**
R-S4	0.005	0.075	0.082	0.079	0.038	0.129	0.119	0.117
R-S9	0	0.041	0.038	0.039	0.006	0.139	0.105	0.108
R-SU*	0.001	0.043	0.041	0.042	0.003	0.144	0.122	0.128
R-SU4	0	0.048	0.049	0.051	0.005	0.158	0.136	0.143
R-SU9	0.001	0.043	0.041	0.042	0.007	0.140	0.106	0.108
R-L	0.003	0.066	0.062	0.064	0.019	0.188	0.160	0.165
R-W1.2	0.002	0.060	0.060	0.062	0.013	0.181	0.155	0.162

Table 7. Results of IMTKU run 2 for Japanese essay subtask

SYSTEM	IMTKU2QALab3							
TYPE	SIMPLE				COMPLEX			
METHOD	content	text	shortest unit (stem)	shortest unit (root)	content	text	shortest unit (stem)	shortest unit (root)
R-1	**0.004**	**0.067**	**0.043**	**0.050**	**0.010**	**0.146**	**0.059**	**0.075**
R-2	0	0.007	0.003	0.003	0	0.022	0.007	0.008
R-S*	**0.004**	**0.049**	**0.041**	**0.048**	**0.009**	**0.124**	**0.058**	**0.071**
R-S4	0.003	0.028	0.026	0.030	0.006	0.043	0.022	0.027
R-S9	0	0.006	0.002	0.003	0	0.027	0.006	0.008
R-SU*	0	0.004	0.001	0.002	0	0.016	0.003	0.005
R-SU4	0	0.004	0.001	0.002	0	0.018	0.004	0.006
R-SU9	0	0.007	0.003	0.004	0	0.028	0.006	0.009
R-L	0.001	0.013	0.006	0.008	0.002	0.037	0.012	0.016
R-W1.2	0.001	0.010	0.004	0.006	0.001	0.030	0.008	0.012

5. CONCLUSIONS

This paper proposed a question answering system using a hybrid approach that integrates natural language processing and deep learning approach for Japanese university entrance exams at NTCIR-13 QA Lab-3. In phase-2, we submitted 3 English End-to-End multiple-choice run results, 2 English End-to-End essay run results, 2 Japanese End-to-End essay run results, 2 English extraction essay run results, 2 Japanese extraction essay run results, 1 English summarization essay run result, and 1 Japanese summarization essay run result, for National Center Tests and Second-stage Examinations. In NTCIR-13 QA Lab-3 phase-3, the IMTKU team total score obtained 34, 40 and 18 in the English subtask of multiple-choice. In Run 1 of English essay subtask, the best scores in the simple English essay and the complex English essay are both found in the feature of stem, namely 0.077 and 0.329 respectively. The best marks in Run 1 are found with the text feature in both the simple Japanese essay and the complex Japanese essay, that is, 0.185 and 0.408 respectively.

The main contribution of this study is that we proposed the IMTKU Question Answering System for world history exams focusing on NTCIR-13 QA Lab-3 National Center Test and Second-Stage Exam. We integrated natural language processing with deep learning approach in the IMTKU Question Answering System with the capability for resolving the National Center Test for University Admissions and Secondary exams in Japanese and English.

6. ACKNOWLEDGMENTS

This research was supported in part of TKU research grant and Ministry of Science and Technology. We would like to thank the support of IASL, IIS, Academia Sinica, Taiwan.

7. REFERENCES

[1] K. Sakamoto, H. Matsui, E. Matsunaga, T. Jin, H. Shibuki, T. Mori, M. Ishioroshi and N. Kando. Forst: Question answering system using basic element at NTCIR-11 QA-lab task. Presented at NTCIR. 2014.

[2] H. Shibuki, K. Sakamoto,Y. Kano, T. Mitamura, M. Ishioroshi, T. Mori, N. Kando NTCIR-12 QA-Lab Task second Pilot.

[3] H. Shibuki, K. Sakamoto, Y. Kano, T. Mitamura, M. Ishioroshi, K. Y. Itakura, D. Wang, T. Mori and N. Kando. Overview of the NTCIR-11 QA-lab task. Presented at Ntcir. 2014.

[4] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: Proceedings of the ACL-04 workshop (Vol. 8).

[5] Stanford University, California. The Stanford Natural Language Processing Group. Stanford CoreNLP – a suite of core NLP tools. July 20, 2015, http://stanfordnlp.github.io/CoreNLP/

[6] M. Steinbach, G. Karypis and V. Kumar. A comparison of document clustering techniques. Presented at KDD Workshop on Text Mining. 2000.

[7] Hideyuki Shibuki, K. S., Madoka Ishioroshi, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, Noriko Kando. (2017). Overview of the NTCIR-13 QA Lab-3 Task. Paper presented at the NTCIR-13, Tokyo.