KSU Team's QA System for World History Exams at the NTCIR-13 QA Lab-3 Task

Tasuku Kimura
Kyoto Sangyo University, Japan
i1658047@cse.kyoto-su.ac.jp

Ryo Tagami
Kyoto Sangyo University, Japan
i1788124@cse.kyoto-su.ac.jp

Hisashi Miyamori
Kyoto Sangyo University, Japan
miya@cc.kyoto-su.ac.jp

ABSTRACT

This paper describes the systems and results of the team KSU for QA Lab-3 task in NTCIR-13. We have been developing question answering systems for the world history multiple-choice questions in the National Center Test for University Admissions. We newly developed automatic answering systems for the world history questions in the second-stage exams of Japanese entrance examinations consisting of the term questions and the essay questions. In the multiple-choice question subtask, we improved on automatic answering systems in QA Lab-2 by implementing query generation methods in accordance with the answer types. In the term question subtask, we designed systems that focus on the category prediction using word prediction models and the evaluation score based on the graph of dependency relations. In the essay question subtask, we proposed automatic answering methods that combines the document retrieval depending on the instructions of how the essay should be summarized, and the knowledge sources constructed from various simple sentences.

Team Name

KSU

Subtasks

Japanese

Keywords

factoid type question answering, non-factoid type question answering, university entrance examination, essay, distributed representation, word order, knowledge source

1. INTRODUCTION

Question answering systems are a kind of search systems that return a single answer to a question given in natural language.

In QA Lab tasks[1][2] at NTCIR-11 and NTCIR-12, question answering systems which can automatically answer world history questions in university entrance examinations had been developed and evaluated. The goal of these tasks is to develop more sophisticated QA systems by challenging more realistic and complex questions such as in university entrance examinations. In QA Lab-3[3], participants were required to develop the systems which can automatically answer either or both the National Center Test and the second-stage exams of university entrance examinations in Japan, and to rather focus on the latter. For the National Center Test, it is essential for the system to correctly choose answers from given multiple choices. For the named-entity type questions in the second-stage exams, it is necessary to properly extract the answer candidate words from the knowledge sources. For the essay type questions in the second-stage exams, it is indispensable to summarize sentences and to generate correct descriptions satisfying the given conditions.

In QA Lab-3, we developed systems for answering named-entity type and essay type questions, respectively, in addition to the system developed so far for multiple-choice questions. Figure 1 shows the configuration of the proposed systems for the named-entity type questions and the essay type questions.

Each system is based on the method using document search proposed by Kupiec et. al.[4], and composed of four units: question analysis unit, document search unit, answer candidate extraction unit, and answer selection/evaluation/generation unit. First, the question analysis unit reads the question data and obtains the query words and the answer category. For example, when answering the slot-filling questions, this unit estimates the category of the correct answer word. Then, the document search unit obtains documents including the answer candidate words or sentences from the knowledge sources by full text search using the query. Next, the answer candidate extraction unit extracts the answer candidate words or sentences from the obtained document sets. Finally, the answer selection/evaluation/generation unit ranks the answer candidates and outputs the top-ranked candidate as the answer.

2. AUTOMATIC ANSWERING FOR MULTIPLE-CHOICE QUESTION

For multiple-choice questions, the system [5] built for QA Lab-2 Task at NTCIR-12 was modified and improved, which mainly focused on query generation methods and different knowledge sources. In order to correctly answer the world history questions in the National Center Test, two points are important; the generation of queries with less unnecessary terms, and the use of precise and comprehensive knowledge sources, because the answer candidates are often obtained by document retrieval. Therefore, we implemented several functions such as query generation corresponding to question types, query generation with particular kinds of words including named entities, adaptive query generation based on the underlined texts in given questions, and utilization of various knowledge sources like textbooks, Wikipedia, and ontologies describing only historical events.

In Phase-2 of QA Lab-3, the system was intended to improve accuracy by increasing the knowledge sources for document retrieval and by implementing the query generation based on the answer types.

3. AUTOMATIC ANSWERING FOR NAMED-ENTITY QUESTIONS

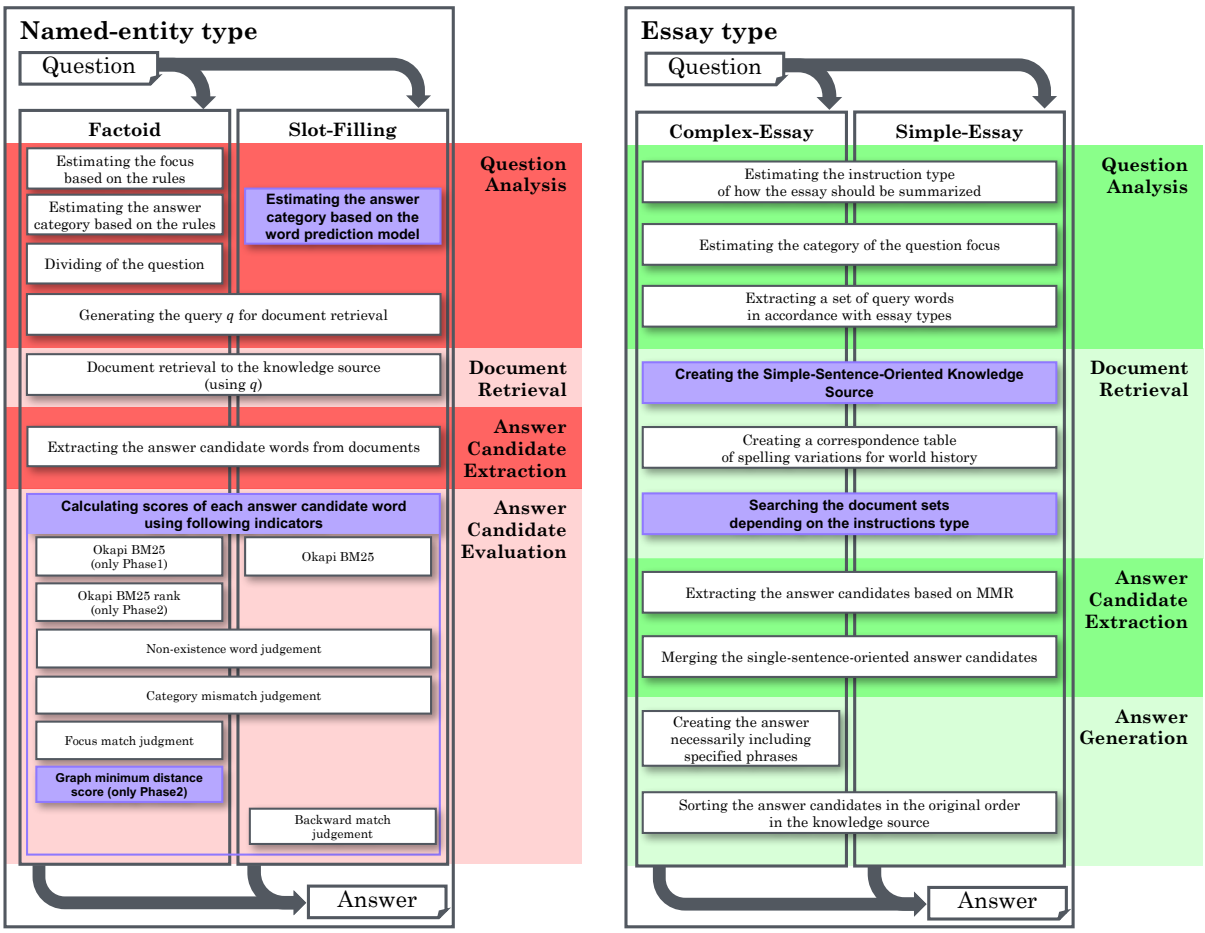

Figure 1: Basic configuration of each automatic answering system.

3.1 Overview

Questions of named-entity type are processed in the order shown as in "Named-entity type" in Fig. 1. Input questions are classified as either factoid type or slot-filling type, using regular expressions. As shown in Fig. 1, the processing and the indicators are different to some extent, depending on the type.

3.1.1 Question Analysis Module

This module mainly estimates the answer category and focus. The answer category represents categories of words that could be an answer to the question, such as "person's name" and "place name", and is classified into 18 types. Also, the focus denotes the attribute of the answer word such as the name of "king" or the name of "politician", and zero or more focuses are assigned to the words relating to world history. Note that the types of the category and the focus are the same as those used in the dictionary for the morphological analysis provided by the organizer, except that an extra category which we judged as necessary was added.

In the factoid type question, the focus is estimated based on the predefined rules, and the answer category is determined from the focus. In some factoid type questions, multiple sub-questions are included in one question sentence. Such a question is divided into parts corresponding to sub-questions based on the predetermined rules, since it is treated as multiple questions in the question data provided by the organizers. Finally, all nouns in the question are extracted by morphological analysis, to obtain the query q for document retrieval.

In the slot-filling type question, only the answer category is estimated by the word prediction model considering the word order. For further details, see Sec. 3.3. Then, from the sentence containing the slot, all nouns are extracted by morphological analysis, to generate the query q.

3.1.2 Document Retrieval Module

This module obtains the document set containing the answer candidates w by document retrieval with the query q obtained in Sec. 3.1.1, against the knowledge source prepared in advance. From the set of the documents d obtained as the search result by the query q, the top k results are outputted as inputs to the answer candidate extraction module.

The knowledge sources were constructed based on the information sources shown in Tab. 5 in Sec. 5.1.3, but they were constructed in different ways in Phase-1 and -2. In Phase-1, the knowledge source was generated from four textbooks and one reference book, with one sentence in the text being considered as one document. In Phase-2, the knowledge source was created from four textbooks and one Web site, with one paragraph in the text as one document. Accordingly, the value of k was set to $k = 50$ for Phase-1 and $k = 5$ for Phase-2, respectively.

3.1.3 Answer Candidate Extraction Module

This module extracts all of the answer candidate word w

from the set of document d obtained in Sec. 3.1.2. Since we think that the correct word of slot-filling type question of world history is necessarily proper noun, all the proper nouns are extracted as answer candidates from each document d.

3.1.4 Answer Candidate Evaluation Module

This module evaluates the likelihood of answers for each candidate word w obtained in Sec. 3.1.3 and determines the final answer word. In this system, for each w, $Score(w)$ is calculated using several indicators. Finally, the system outputs the answer based on the ranking of the score.

3.2 Evaluation Indicator in Answer Candidate Evaluation

Various evaluation indicators are used for the calculation of $Score(w)$ described in Sec. 3.1.4. Seven indicators are used as shown in Tab. 1. Also, as shown in Tab. 2, different indicators are used depending on Phase, RUN and the type of question.

$Score(w)$ is the score of each answer candidate w, and is calculated by summing all the scores output by each indicator.

3.3 Category Estimation using Word Prediction Model

For the slot-filling type questions, a word prediction model[6] was constructed which estimates the center word from the surrounding words of the filling part. This model was constructed using distributed representations of words, with four textbooks shown in Tab. 5 of Sec. 5.1.3 as the training data. The center words are predicted in consideration of word order, by adopting the Word Order model proposed by Ariga et al. [7] as the training model.

Using the word prediction model above, the processing to estimate the answer category from the input question is incorporated into Question Analysis module as described in Sec. 3.1.1. When inputting the surrounding words of the slot into the model, the set of the center word candidates is output. Categories assigned to each word in the set are collated, and all matched is set as the category of the question.

3.4 Graph Minimum Distance Score

As shown in Tab. 1, the graph minimum distance score was introduced from Phase-2 as one of the evaluation indicators of the answer candidates in the factoid type questions. The value of the score is calculated by the method proposed by Kurata et al. [8] using the graph created from the dependency analysis on the relevant documents obtained by Document Retrieval module.

Figure 2 shows an example of the process of constructing the graph. First, dependency analysis is performed for each sentence for the top k documents with high relevance to the question, obtained by Document Retrieval module. Then, the analysis results of the whole sentences are integrated to construct a graph such that the Japanese bunsetsu unit is a node and the dependency relationship is an edge. When creating a node, adjuncts etc. are excluded in advance.

Next, the graph minimum distance score is calculated for each candidate word w using the graph generated for each question, in Answer Candidate Evaluation module. Specifically, for each node of w on the graph, the minimum distance from the node of the word used in the query q generated in Sec. 3.1.1 is calculated using Dijkstra's algorithm, as the graph minimum distance score for w. When q contains multiple words, the score is the sum of the distance for each word in q. That is, the value of the graph minimum distance score becomes smaller as the words used in w and q exist closer to each other on the graph. Therefore, this indicator shows that the evaluation becomes higher as the score value is smaller.

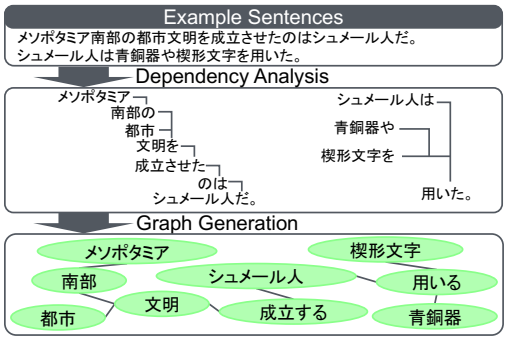

Figure 2: Example of the process of constructing the graph.

4. AUTOMATIC ANSWERING FOR ESSAY QUESTIONS

As shown in the right of Fig. 1, the system generates answers through the question analysis unit, the document search unit, the answer candidate extraction unit, and the answer generation unit.

There are two kinds of essay questions: the complex essay questions where the test-takers write an essay of about ten lines, and the simple essay questions to write an essay of about a few lines. The main differences in generating answers for the complex essay questions and for the simple essay questions are only the methods to obtain the query words described in Sec. 4.1.3, and other processes are basically the same.

4.1 The Question Analysis Units

The following three elements are extracted from the question section by rule-based methods: the instruction type of how the essay should be summarized, the focus of the questions which is an indication of what matters to describe the essay, and the character limit.

4.1.1 Estimating the instruction type of how the essay should be summarized

In order to estimate the instruction type of how the essay should be summarized shown in Tab. 3, a correspondence table between each instruction type and the clue words were manually created. Table 3 shows an example of the correspondence between instruction types and clue words.

4.1.2 Estimating the category of the question focus

In order to estimate the category of the question focus, the method which was proposed by the authors[5] for the world history questions in the National Center Test for university admissions, was used to assign the labels of superordinate concepts to the named entities related to world history. We created manually a correspondence table between superordinate concept labels from the event ontology EVT[1] and the named entity tags from the textbook in which each word is annotated with the named entities (NE_Tokyoshoseki) provided from the organizer of QA Lab-2 Task. The event ontology EVT has the total of 4,793 important events and people described in high school textbooks, which are classified in their superordinate concept labels such as "nation and dynasty", "social systems", and "technology and invention". NE_Tokyoshoseki contains 14,622 named entities annotated with 32 kinds of tags. Each entity was labeled with at least one or more tag such as "person type, social role", "social system", and "historical event".

[1]event ontology EVT : http://researchmap.jp/zoeai/event-ontology-EVT/

Table 1: Evaluation indicators for named-entity questions.

#	Indicator names	Summary
1	Okapi BM25	The maximum value of BM25 of the documents d including w for the query q, meaning that the larger the value, the higher the evaluation.
2	Okapi BM25 rank	The rank of the corresponding document above when sorted in descending order by the value of BM 25, meaning that the smaller the value, the higher the evaluation.
3	Non-existence word judgement	This indicator increases an evaluation if w does not exist in the question.
4	Category mismatch judgement	This indicator collates the category of w and decreases an evaluation if it does not match any one of the categories predicted in Sec. 3.1.1.
5	Focus match judgement	This indicator collates the focus of w and increases an evaluation if it matches the focus predicted in Sec. 3.1.1.
6	Graph minimum distance score	For this indicator, see Sec. 3.4.
7	Backward match judgement	This indicator increases an evaluation if the next word of the slot is a noun and the word matches the backward part of w.

Table 2: Indicators used in each Phase and RUN.

	Factoid					Slot-filling[1]
	Phase-1		Phase-2			Phase-2
	RUN	RUN	RUN	RUN	RUN	RUN
#	1	2	1	2	3	1 – 3
1	✓	✓	✓	✓		✓
2					✓	
3	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓
5	✓		✓	✓	✓	
6				✓[2]	✓	
7						✓

[1] Slot-filling type questions existed only in the test data for Phase-2.

[2] This indicator was used only when $Score(w)$ were equal after being evaluated using other indicators.

In the question analysis unit, the question part is converted into the word sets which is divided for each morpheme. It matches the word sets and the clue words of the correspondence table, and if a target word in the word sets was attached one of the named-entity tags used in NE Tokyoshoseki, the super-ordinate concept label was obtained from the correspondence table. The question part has all super-ordinate concept labels acquired by this method.

4.1.3 *Extracting a set of query words in accordance with essay types*

In this Section, we explain the acquisition method of each query keywords for the complex essay questions and the simple essay questions.

In answering the complex essay questions, when there are some phrases which must be included in the essay, the phrases are used as query keywords. Otherwise, the query keywords are obtained in the same way as for the simple essay questions.

Meanwhile, since no phrase was given which must be included in the essay in case of the simple essay question, documents were searched using the content words of the question part as query keywords. Also, when the question part has an expression referring to the context part, the content words of the sentence containing the referenced phrase in the context part were added to the query keywords. Also, the words co-occurring with the question focus category and ones co-occurring with the query keywords of the named entities were obtained from the co-occurrence words knowledge sources in Sec. 4.2 as extended query keywords.

4.2 The Document Search Units

A set of sentences which are the candidate sentences for the essay are acquired by searching for the knowledge sources described in Sec. 4.2.2 using the query words obtained in Sec. 4.1.3, according to the instruction type extracted by the question analysis unit.

4.2.1 *Searching the document sets depending on the instructions type*

At first, when the instruction type is "summary", "process/change" or "characteristic", the system performs simply OR search for the knowledge sources with the query words to obtain a set of sentences which becomes the candidate sentences of the essay.

Then, when the instruction type is "relevance/affect", only sentences with cause or reason expressions in the knowledge sources were OR searched with the query words to obtain the candidates sentences of the essay.

Finally, when the instruction type is "comparison", the common points are at first searched for a certain knowledge source based on the question. For example, "belonging to an allied country" are extracted as the common points from the question "Please describe common points between the United States and Britain in World War II". Then, only sentences including the common points were OR searched with the query words to obtain the candidate sentences of the essay. In order to identify the common points, a co-occurrence word knowledge source was constructed in advance by calculating the PMI (Pointwise Mutual Information) between words of the knowledge sources which were created from the textbook described in Sec. 4.2.2 with each sentence being registered as a document. The label of the superordinate concepts estimated by the method explained was assigned as the focus category to the word B co-occurring with the word A. This enables to search "word B having a specific focus category co-occurring with word A". When identifying the common points to word A and B, the set of words co-occurring with each word were acquired using the focus category and the co-occurrence word knowledge source, and the words commonly included in the two sets were extracted as the common points. If we could not identify common points by this method, the system performs a simple OR search for knowledge sources with the query words and acquires the candidates sentences of the essay.

4.2.2 *Creating the simple-sentence-oriented knowledge source*

We introduce the simple-sentence-oriented knowledge sources where the surface expressions are simplified in various ways compared to their original sentences, so that the system can obtain concise answer candidate sentences containing only the content which should be included in the answer.

Table 3: Examples of instruction types of how the essay should be summarized and the corresponding clue words.

Instruction types	Contents of essay to be generated	Example of Clue words
summary	the brief description	まとめよ (summarize)
process/change	the overview of the specific period	経緯 (process)
relevance/affect	the relationship between phrases	背景 (background)
characteristic	the characteristics of phrases	特質 (characteristic)
comparison	the comparison between instructed phrases	違い (difference)

The simple-sentence-oriented sentence is defined which is a document in the simple-sentence-oriented knowledge source. At first, classification of sentences based on subject and predicate structure is explained.

Japanese sentences are composed of clauses, and clauses are classified into the following five components according to its role; subject, predicate, modifier, conjunction, and independent word. Figure 3 shows the classification of sentences based on the structure of subjects and predicates in a sentence. Normal sentences are often any one of "simple

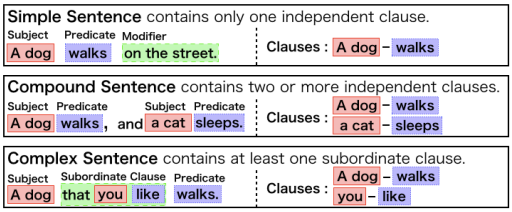

Figure 3: Instructions of sentences by clauses (a relationship between subject and predicate).

sentence", "compound sentence" and "complex sentence" including components other than subjects or predicates. In other words, when it is assumed that "one meaning is represented by a pair of a subject and its predicate", "the complex sentence" and "the compound sentence" can be said to be a complicated sentence having plural meanings.

By contrast, a "simple sentence" tend to be short and simple, because it basically only contains a subject and a predicate. Therefore, as shown in Fig. 4, we introduce a simplified surface representation by converting a "simple sentence", "complex sentence" or "compound sentence" which include components other than subjects and predicates, into one or more "simple sentences" which have less components than the original sentence.

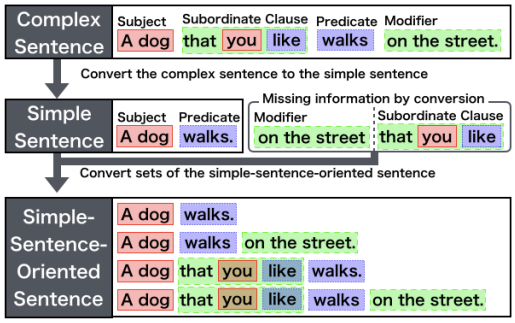

Figure 4: Examples of converting a complex sentence into a simple sentence or simple-sentence-oriented sentences.

However, it is not preferable that the information of clauses in the modifier gets completely lost because of this conversion. For this reason, a "simple sentence" without any components other than a subject and a predicate is taken as a base sentence, and several other sentences are generated by adding to the base sentence one of more components other than the subjects and the predicate in the original sentence.

Based on the above, we define a simple-sentence-oriented sentence as follows: a simple-sentence-oriented sentence ≡ a simple sentence composed only of a subject and a predicate, added by zero or more components other than the subject and the predicate.

Algorithm 1 shows the procedure for generating the single-sentence-oriented knowledge source. In this algorithm, the simple sentence containing only a subject and a predicate in the main clause is built and added by other clauses to generate a series of simple-sentence-oriented sentences.

We created four knowledge sources to be used in document search from the textbooks of world history for high school students published by Tokyo Shoseki Co., and by Yamakawa Shuppansha Ltd., provided by QA Lab organizers. The "Original Tokyo Shoseki" knowledge source, hereafter referred to as "T", was generated from the textbook published by Tokyo Shoseki Co., and by registering each sentence as a document. The "Original Yamakawa Shuppansha" knowledge source, hereafter referred to as "Y", was generated from the textbook published by Yamakawa Shuppansha Ltd. in the same way as "T". The "Simple-Sentence-Oriented Tokyo Shoseki" knowledge source, hereafter referred to as "TR", was generated from the textbook published by Tokyo Shoseki Co., and by registering the simple-sentence-oriented sentence converted from each sentence as a document. The "Simple-Sentence-Oriented Yamakawa Shuppansha" knowledge source, hereafter referred to as "YR", was generated from the textbook published by Yamakawa Shuppansha Ltd., in the same way as "TR".

Also, each knowledge source has the flag indicating the presence or absence of the representation of the cause or the reason. This flag is used as one of the query keywords when the question has the instruction type of "relevance/affect".

4.2.3 *Creating a correspondence table of spelling variations for world history*

In order to handle the spelling variations peculiar to world history, a correspondence table of spelling variations were used which was created by the authors[5] and which was specialized in the world history questions in the National Center Test for university admissions.

4.3 The Answer Candidate Extraction Units

4.3.1 *Extracting the answer candidates based on MMR*

Based on the idea of Maximal Marginal Relevance (MMR)[9], the answer candidate sentences were extracted from the document sets obtained by the document search unit. In the sentence summarization method based on MMR, each answer candidate sentence is scored according to the importance degree for an answer, and is selected as summary sentences in descending order of the score. This process is repeated until the character limit gets satisfied. In this process, the degree of similarity is calculated between the content of one answer candidate sentence which has not yet

Algorithm 1: Algorithm of creation processing of simple sentences

```
Data: Text
Result: SimpleSentences
begin
    /* Get the list of clauses from the text by syntax parsing */
    Clauses = syntacticParsing(Text)
    /* Variable contains the predicate of the main clause */
    Predicate_Main = NULL
    /* Variable contains the subject of the main clause */
    Subject_Main = NULL
    for Clause ∈ Clauses do
        if Clauseispredicate then
            Predicate = Clause
            if Clause is MainClause then
                Predicate_Main = Predicate
                Subject_Main = Predicate_Main.getSubject()
                Clauses.remove(Predicate)
            Clauses.remove(Predicate.getSubject())
    /* The list of simple-sentence-oriented sentences */
    SimpleSentences = []
    /* Add the simple sentence generated from the main clause to the list */
    SimpleSentences.add(Subject_Main.toText() + Predicate_Main.toText())
    /* Generate all combinations of the simple sentences from the main clause and other clauses */
    for counter = 1 to Clauses.length do
        /* Generate _Clauses C_counter */
        Combinations = C(Clauses, counter)
        for Combination ∈ Combinations do
            for Clause ∈ Combination do
                /* If the element of Combination contains the predicate, add the corresponding subject to Combination */
                if Clause = predicate then
                    Combination.add(Clause.getSubject)
            /* Add the simple sentence generated from the main clause to the list */
            Combination.add(Subject_Main)
            Combination.add(Predicate_Main)
            /* Sort the elements of each combination in the order of arrangement in the original text */
            Combination.sortInOriginalTextOrder
            SimpleSentences.add(Combination.toText())
```

been selected and the content of each answer candidate sentence already selected. Then, the maximum degree of similarity is taken as the penalty score, and the final score is calculated by subtracting the penalty score from the importance score as a answer. In the proposed method, an answer candidate sentence having the maximum final score $Score_f$ is added to the set of answer sentences. $Score_f$ is calculated from the importance score and the penalty score, as follows:

$$Score_f(D, A) = \max_{D_i \in D \setminus A} [Okapi\ BM25(D_i, Q) - Simpson's\ Coefficient(D_i, A)] \quad (1)$$

Where D is the set of answer candidate sentences A represents the set of answer sentences already selected, D_i denotes the answer candidate sentence which has not yet been selected, and Q expresses the query used to search D. The value of Okapi BM25[10] given at the document retrieval was used as the importance score of each answer candidate sentence. The penalty score was calculated using Simpson's Coefficient with an original morpheme trigram as an element.

4.3.2 *Merging the single-sentence-oriented answer candidates*

The simple-sentence-oriented knowledge source described in Sec. 4.2.2 is used and when an answer candidate sentence to be newly added and one of the answer candidate sentences already selected are both single-sentence type documents originally generated from the same sentence, the two sentences are merged.

4.4 The Answer Generation Units

The answer candidate sentences obtained in the answer candidate extraction unit were sorted in the original order in the knowledge source and concatenated to output the final essay as an answer. Answer candidate sentences were sorted in the original order in the knowledge source and concatenated to be output as the answer.

4.4.1 *Creating the answer necessarily including specified phrases*

When particular phrases are given in complex essay questions, those phrases must be included in the answer. If any of them is not included, the answer will be scored considerably low. Therefore, the answer generation method was implemented, that always includes specified phrases, based on the method by Sakamoto et al.[11].

In order to obtain the answer candidates including each specified phrase, the following processing was performed in the document retrieval units in advance. First, the query words obtained by using the query generation method for simple essay questions in Sec. 4.1.3 are considered as the common query words in the subsequent processing. Then, documents were searched for each specified phrase, and OR search was performed using the common query words only for the obtained documents, to acquire a set of candidate sentences for summary.

Finally, from the set of candidate summary sentences obtained from each specified phrase, the documents with the largest Okapi BM 25 given at the time of document search were selected one by one as summary sentences.

4.5 System Configuration

Table 4 shows the difference of the system configurations for Phase-1 and -2 in (1) and (2).

(1) Creating the simple-sentence-oriented knowledge source
(2) Creating the answer necessarily including specified phrases

For convenience and clarity, each phase is represented as "PH1" and "PH2", respectively, and included in the system's ID below.

In Phase-1, two systems were developed depending on whether the single-sentence knowledge source was used or not. In Phase-2, based on KSU-ESSAY-01@PH1 and KSU-ESSAY-02@PH1 developed in Phase-1, three types of systems were constructed depending on whether or not to use the answer generation method necessarily including the specified phrases.

5. EXPERIMENTS

5.1 Common Resources of Each Answering System

Table 4: The comparison of the system configuration for the essay questions.

System Id	(1)	(2)
KSU-ESSAY-01@PH1	T	No
KSU-ESSAY-02@PH1	TR	No
KSU-ESSAY-01@PH2	T	No
KSU-ESSAY-02@PH2	T	Yes
KSU-ESSAY-03@PH2	TR	Yes

5.1.1 *Tools*

For document retrieval of all the systems, Apache Solr[2], an open source full-text search engine, was adopted, and Okapi BM 25 was used as a document weighting method. In addition, in all systems, MeCab[3] and Juman++[4] was used for Japanese morphological analysis, and KNP[5] for Japanese dependency analysis.

5.1.2 *Dictionaries*

As the system dictionary of morphological analysis, mecab-ipadic was used in the systems for multiple-choice and essay type questions, whereas mecab-ipadic-NEologd developed by Sato[12] was utilized in the system for named-entity type questions.

As the user dictionary of morphological analysis, a dictionary[5] was created and used, which includes proper nouns specialized in world history.

5.1.3 *Information sources*

The information sources shown in Tab. 5 were used to construct the knowledge sources for Document Retrieval module.

Table 5: The information sources used in construct the knowledge sources.

Type	Title
Textbook	詳説 世界史 B
Textbook	世界史 B
Textbook	新選世界史 B
Textbook	世界史 A
Reference Book	山川一問一答世界史
Web Site	世界史の窓 [1]

[1] 世界史の窓：http://www.y-history.net

Four textbooks in the table represents the data of textbooks actually used in high schools, and were provided by the organizers of QA Lab-3. Another reference book called "山川一問一答世界史", which literally means "world history in one-question-to-one-answer style by Yamakawa", is composed of a collection of pairs of one factoid question and its answer, and was utilized as an information source by constructing the declarative sentences from each factoid question and its answer. For the Web site called "世界史の窓", which literally means "the windows of world history" in English, all pages explaining the terms present in the site were collected and their texts were adopted as the information source.

5.2 Results of Multiple-choice Questions

As the test data, the problems of National Center Test for University Admissions in 2012 and 2013 were used in Phase-1, and those tests in 2014 in Phase-2. Table 6 shows the results of the correct answer rate by our systems.

[2]Apache Solr：http://lucene.apache.org/solr/
[3]MeCab：http://taku910.github.io/mecab/
[4]JUMAN++：http://nlp.ist.i.kyoto-u.ac.jp/?JUMAN++
[5]KNP：http://nlp.ist.i.kyoto-u.ac.jp/?KNP

Table 6: Results of our runs in multiple-choice end-to-end task.

Phase	System Id	Accuracy
Phase-1	KSU-MULTIPLE-01@PH1	0.31(22/72)
	KSU-MULTIPLE-02@PH1	0.22(16/72)
	KSU-MULTIPLE-03@PH1	0.33(24/72)
Phase-2	KSU-MULTIPLE-01@PH2	0.44(16/36)
	KSU-MULTIPLE-02@PH2	0.44(16/36)

5.3 Results of Named-entity Questions

As the test data, the questions for phrase answer of world history B in the second-stage entrance examinations of the University of Tokyo in 2000, 2004, 2008, 2012 and 2013 were used in Phase-1, and those tests in 2001, 2002, 2006, 2010 and 2014 were used in Phase-2. Table 7 shows the results of the correct answer rate by our systems.

Table 7: Results of our runs in named-entity end-to-end task.

Phase	System Id	Accuracy
Phase-1	KSU-TERM-01@PH1	0.29(20/68)
	KSU-TERM-02@PH1	0.26(18/68)
	(KSU-TERM-03@PH2)[1]	0.35(24/68)
Phase-2	KSU-TERM-01@PH2	0.30(23/77)
	KSU-TERM-02@PH2	0.29(22/77)
	KSU-TERM-03@PH2	0.31(24/77)

[1] This is an informal RUN for further discussion.

5.4 Essay Questions

As the test data, the questions for phrase answer of world history B in the second-stage entrance examinations of the University of Tokyo in 2000, 2004, 2008, 2012 and 2013 were used in Phase-1, and those tests in 2001, 2002, 2006, 2010 and 2014 were used in Phase-2. Figure 5.4 shows the results of F-measure of ROUGE-N (N = 1, 2) by our systems for Phase-2.

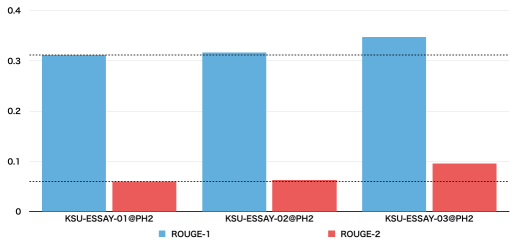

Figure 5: Results of our runs for Phase-2 in essay end-to-end task.

6. DISCUSSION

6.1 Named-entity Type Question

In this section, the systems for the named-entity type questions are discussed. Here, only the factoid type questions are considered because there were just a few slot-filling type questions in the test data for Phase-2.

As shown in Tab. 2, the main difference of each RUN in Phase-2 is whether the indicator based on the graph minimum distance score was used. Table 7 shows the result of their comparison and it indicates that the correct answer rate of the RUN with the graph minimum distance score becomes slightly higher that of the RUN using the value of BM25.

As shown in Tab. 7, an informal experiment was performed where the system with the highest accuracy in Phase-2 answered the test data of Phase-1. Also from this result, it is confirmed that the introduction of graph minimum distance score contributes to improvement of the correct answer rate.

Kurata, et al. [8] proposed a method based on the assumption that the distance between each named entity in the question sentence and the correct word becomes shorter on the dependency graph based on the knowledge source. The graph minimum distance score in this work is also an indicator introduced based on the same assumption. Examining the cases where the questions were correctly answered, it was confirmed that the distance between each named entity and the correct word was relatively small. However, it was also confirmed that the system tends to give incorrect answers in the following cases: when there were few named entities in the question, when the named entity in the question does not exist on the knowledge source in the first place, or when the distance between the named entities on the graph happens to be long. As a cause of these problems, insufficient correspondence to spelling variations of words of each node is considered, because the collation is based on the exact match of the surface strings. Therefore, it is expected that these problems are alleviated by normalization with thesauruses and/or by introduction of partial match.

6.2 Essay Type Question

In this section, the systems for essay type questions developed for Phase-2 are discussed.

Figure 5.4 shows the ROUGE-N (N = 1, 2) of KSU-ESSAY-02@PH2 was a little higher than that of KSU-ESSAY-01@PH2. This improvement of ROUGE-N is considered to be achieved because the sentences including the correct answer were successfully selected by converting these sentences which could not have been selected due to the character limit of the question, into the simple sentences. However, it is necessary to improve the method of converting to the simple-sentence-oriented sentences, because the proposed knowledge sources contain several unnatural sentences without sufficient semantics.

Also, Fig 5.4 indicates that the ROUGE-N (N = 1, 2) of KSU-ESSAY-03@PH2 was higher than that of KSU-ESSAY-02@PH2. It was confirmed that the essay generated by KSU-ESSAY-03@PH2 contained the appropriate sentences as the answer, because it implemented the method of using the candidate sentences always including the specified phrases. These results showed the similar tendency to the characteristics of the original method proposed by Sakamoto et al.[11].

7. CONCLUSION

This paper described the systems and results of the team KSU for QA Lab-3 task in NTCIR-13. We have been developing question answering systems for the world history multiple-choice questions in the National Center Test for University Admissions. We newly developed automatic answering systems for the world history questions in the second-stage exams of Japanese entrance examinations consisting of the term questions and the essay questions. In the multiple-choice question subtask, we improved on automatic answering systems in QA Lab-2 by implementing query generation methods in accordance with the answer types. In the term question subtask, we designed systems that focus on the category prediction using word prediction models and the evaluation score based on the graph of dependency relations. In the essay question subtask, we proposed automatic answering methods that combines the document retrieval depending on the instructions of how the essay should be summarized, and the knowledge sources constructed from various simple sentences.

8. ACKNOWLEDGEMENTS

A part of this work was supported by Kyoto Sangyo University Research Grants.

9. REFERENCES

[1] H. Shibuki *et al.*, Overview of the ntcir-11 qa-lab task., in: NTCIR, 2014.

[2] S. Hideyuki *et al.*, Overview for the ntcir-12 qa lab-2 task, in: Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, 2016, pp. 392–408.

[3] S. Hideyuki *et al.*, Overview of the ntcir-13 qa lab-3 task (draft), in: Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies, 2017.

[4] J. Kupiec, Murax: A robust linguistic approach for question answering using an on-line encyclopedia, in: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '93, ACM, New York, NY, USA, 1993, pp. 181–190.
URL http://doi.acm.org/10.1145/160688.160717

[5] T. Kimura, R. Nakata, H. Miyamori, KSU team's multiple choice QA system at the NTCIR-12 QALab-2 task.

[6] R. Tagami, T. Kimura, H. Miyamori, Automatic answering method considering word order for slot filling question of university entrance examinations, IPSJ Transactions on Databases Vol.10 No.3 (2017) 45–57.

[7] S. Ariga, Y. Tsuruoka, Synonym extension of words according to context by vector representation of words (in japanese), in: 2015 The Association for Natural Language Processing, 2015, pp. 752–755.

[8] G. Kurata, N. Okazaki, M. Ishizuka, Question answering system with graph structure from dependency analysis, IPSJ SIG Technical Report (2003) 69–74.

[9] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1998, pp. 335–336.

[10] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.

[11] K. Sakamoto *et al.*, Forst: Question answering system for second-stage examinations at ntcir-12 qa lab-2 task.

[12] T. Sato, Neologism dictionary based on the language resources on the web for mecab (2015).
URL
https://github.com/neologd/mecab-ipadic-neologd