FLL: Answering World History Exams by Utilizing Search Results and Virtual Examples

Takuya Makino†, Seiji Okura†, Seiji Okajima†, Shuangyong Song‡, Hiroko Suzuki†,
†Fujitsu Laboratories Ltd.
‡Fujitsu R&D Center Co., Ltd.

ABSTRACT

This paper describes our system for answering Center Exam subtask of QALab with three solvers. The first solver is based on search results obtained with different search engines as clues for answering questions. The second solver is trained with text books and virtual examples, generated automatically from text books by randomly replacing words in the text books. The third solver is for answering correct chronological order of historical events. The experimental results on the formal run data of QALab shows that a combination of solvers contribute to improved accuracy.

Team Name

FLL

Subtasks

Center Exam

Keywords

QALab

1. INTRODUCTION

This paper describes our system to answer multiple choice questions of Center Exam. Our system answers center exams by the following steps with the three solvers.

- Classify questions into types.
- Choose one of the following solvers to be applied to each question with the type of the question.
  - A solver that uses search results on text books and Wikipedia as clues for answering center exams.
  - A solver trained with text books and virtual examples for giving higher scores to choices of exams similar to sentences in text books.
  - A solver for answering correct chronological order of historical events.

Section 2 describes the question types used in this solver. Then Section 3, 4 and 5 describe the three solvers used in our system. Section 6 describes experimental results on the formal run data of QALab.

2. CLASSIFYING QUESTIONS

In this section, the question types and a labeling method is described.

Table 1: Distribution of Answer Category

	1997	2001	2005	2009
1	4	5	8	2
2	3	6	1	0
3	1	0	0	0
4	25	25	24	26
5	0	0	1	4
6	7	3	2	4
total	40	39	36	36

2.1 Question Types

This section describes the definition of the question types used in our system. The question types consists of the following two types of categories.

- Answer Category (AC): AC is used for describing types of answers expected by questions. The AC consists of the following six categories defined based on the four types of [2].
  1. A question to answer a word or words that correctly fill in the blanks of a given text.
  2. A question to answer the most appropriate word or words for a given explanation from the choices.
  3. A question to answer correct chronological order of historical events.
  4. A question to answer the most appropriate sentence for a given explanation from the choices.
  5. A question to answer the correct combination of true or false for given statements.
  6. A question other than the above five categories.
- True or False Question Category (TFQC): TFQC is used for describing whether given questions are to choose a true statement or a false statement for a given question as its answer. In the following, ''+" is used for questions that require to choose a true statement for a given question as its answer, and "-" is used for the others.

Table 1 shows the distribution of question types for each year.

2.2 Labeling Questions

To label questions with AC and TFQC, we annotated four years (1997, 2001, 2005, 2009) Center Exam data of the world history B in Japanese with the categories of AC and TFQC. Then, we manually developed rules for labeling questions with AC, TFQC and NA on the annotated data.

For labeling questions with AC, we developed rules that use tags and the attributes of tags given annotate with the QALab organizer as clues.

For labeling questions with TFQC, we developed rules consisting of pairs of ⟨a keyword/phrase, TFQC label⟩, such as
⟨ 最も適切な (the most appropriate), “+”⟩,
⟨ 正しい (right), “+”⟩, ⟨ 誤った (wrong), “-”⟩,
and so on.

3. A SOLVER BASED ON SEARCH ENGINES

This section describes a solver using search engine results as features.

3.1 Search Engines

To obtain clues to answer questions, we used search engines. First we describe corpus used to obtain clues. Then, the base search engines used in this solver are described.

3.1.1 Corpus

The following three corpus are used in this solver.

- **The Textbook of Yamakawa Shuppansha Ltd. Corpus** (TY): This is a textbook for a Japanese world history prepared by the QALab organizer. This textbook is annotated with two tags: paragraph and topic.
- **The Textbook of TOKYO SHOSEKI CO., LTD. Corpus** (TT): This is also a Japanese textbook for a world history prepared by the QALab organizer. This textbook is annotated with three tags: paragraph, section and topic.
- **Wikipedia Corpus** (WIKI): We used Japanese Wikipedia as of April 14 2014. The markups were removed by some rules in advance for indexing of search engines. We also used characters in links to each article as the part of the texts of the article.

3.1.2 Base Search Engines

The following an N-gram Search Engine and a semantic search engine are used.

N-gram Search Engine: We used an in-company search engine. We built indexes of corpus with character bigrams. When indexing corpus, all characters in the corpus were normalized with the NFKC normalization method.

Semantic Search Engine: In order to consider meaning of queries, we used a semantic search engine [4].

Figure 1 shows a part of a semantic representation of “太郎 (Taro) は (Japanese particle) 花子 (Hanako) に (Japanese particle) 本 (book) を (Japanese particle) あげた (gave) 。(period) ”. Each node represented by a circle indicates a semantic symbol of a word and each rectangle indicates a relation between nodes.

From such a semantic expression, we extract a set of pairs of nodes that have a relation as queries. Then we search sentences. For example, “⟨GIVE, HANAKO⟩” “⟨GIVE, TARO⟩” and “⟨GIVE, BOOK⟩” are extracted as queries from the semantic representation in Figure 1.

A search result includes sentences that include one of a pair of nodes at least. A search result also includes a number of times queries appears in each sentence.

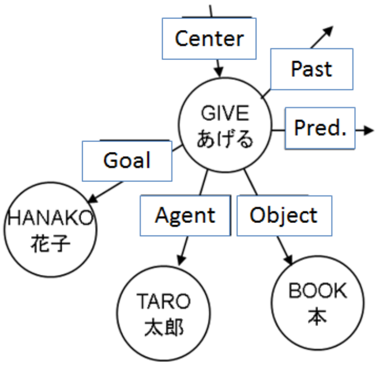

Figure 1: A part of a semantic representation of “Taro gave Hanako a book.”

A Japanese Wikipedia is used for this semantic search. Sentences in the Japanese Wikipedia are parsed in advance.

3.1.3 Search Engines used in This Solver

We developed the following seven search engines with the two search engine in Section 3.1.2.

- TY Paragraph search based on N-gram search (TYP-N): a text annotated with a paragraph in TYC is treated as a document.
- TY Topic search based on N-gram search (TYT-N): a text annotated with a topic in TYC is treated as a document.
- TT Paragraph search based on N-gram search(TTP-N): a text annotated with a paragraph in TTC is treated as a document.
- TT Section search based on N-gram Search (TTS-N): a text annotated with a section in TTC is treated as a document.
- TT Topic search based on N-gram Search (TTT-N): a text annotated with a topic in TTC is treated as a document.
- WIKI search based on N-gram Search (WIKI-N): a Wikipedia article is treated as a document.
- WIKI search based on Semantic Search (WIKI-S): a Wikipedia article is treated as a document.

In addition we used the following search method.

- WIKI search based on Wikipedia Titles (WIKI-T): If a Wikipedia title is included in a given text the title of the Wikipedia article is returned as its search result.

3.2 Searching Clues

We collect clues with the search engines in Section 3.1.3.

3.2.1 Selecting Texts for Building Queries

The following parts in a question used for bulding queries.

Texts annotaed with choice, instruction and underline tags: The QALab center exams are annotated with choice, instruction, and underline tags. A text in a choice tag is a text that we have to decide true or false. Texts in instruction and underline tags include additional information of choices. This solver generates a query from the texts in choice, instruction and underline tags.

Texts annotaed with data tags: The QALab center exams are also annotated with data tags. A text in a data tag usually describes background of an answer. Therefore, we also generated query terms from the text in a data tag, which has a link to a current question.

3.2.2 Search

For the N-gram search engines in Section 3.1.3, we used nouns as query terms in the texts described in Section 3.2.1. To recognize nouns from Japanese texts, an in-company Japanese morphological analyzer is used. The query given to the N-gram search engines is described "OR query", which indicates a search that returns texts that include one of nouns at least.

For the semantic search engine WIKI-S in Section 3.1.3, we give texts in choices to the semantic search engine. The semantic search engine internally parses the given text and searches Wikipedia.

For WIKI-T, if there exist characters corresponding to Wikipedia titles in the selected texts, the titles of the articles are returned as its search results.

3.3 Features

This solver was trained with a machine learning algorithm for using different search results. The following features were used.

- Word match based features: Similarity is a sum of weight of nouns in choice and instruction between a choice and a sentence given by a search engine. This similarity measure is used for all search results.

- Tree Edit Distance based features: To calculate a syntactic similarity between sentences, we used Tree Edit Distance[5]. A similarity based on Tree Edit Distance is a inverse of Tree Edit Distance normalized by a number of nodes of tree. We used an in-company Japanese dependency parser to get parse tree from Japanese text. This measure is used for search results obtained with TTP-N.

- Date based Features: We assumed a difference of dates of events included in a choice is larger than those of the other choices, the choice has the largest difference would be false. To generate features, this solver first finds characters corresponding to Wikipedia titles in choice, instruction and underline Then, this solver extracts all date expressions from the abstract of the found Wikipedia titles. Each abstract is a lead sentence in an article. Then, this solver calculates difference between median of date expressions in a choice and those of the other choices.

- Search Results on English Corpus based Features: We also used features used in the system of FRDC_QA obtained with English Wikipedia. Please see their paper for the detail.

The features are used in a classifier trained with a machine learning algorithm described in Section 3.4. When answering questions, if the question of TFQC is "+", the choice that has the highest score given by the classifier as the answer of the question. Otherwise (TFQC is "-"), the choice that has the lowest score given by the classifier as the answer of the question.

To consider differences of scales of feature values, instead of using values calculated with the above methods, we used rankings of each feature in the choices of a question decided by values of features. For example, a feature in a choice of a question that has the largest value among the values of the same type feature in all the choices of the question is ranked as first.

3.4 Training

We used 4 data sets as a training data which are National Center Test for University Admissions in 1997, 2001, 2005 and 2009. Amounts of training data are shown in Table 2.

Table 2: Number of training data of type 4 questions

	# of pos	# of neg
1997	47	53
2001	35	65
2005	29	71
2009	36	68

We used SVMs with a polynomial kernel to assign scores to choices. A 5-fold cross validation was conducted to decide parameters of SVMs. The parameters maximized the accuracy in the training data were chosen for the parameter of the test data. A polynomial kernel to 3 and regularization parameter C to 0.001 were chosen.

4. A SOLVER TRAINED WITH VIRTUAL EXAMPLES

We trained a solver that assigns scores of the correctness in terms of history to the sentences. Our assumption is the following.

- Sentences in the text books, Tokyo Shoseki and Yamakawa ones, are historically correct.

- Sentences generated from the text books by replacing the words in the text books randomly are historically incorrect.

In order to generate sentences from the text books, we used a set of Named Entities (NEs) for world history [3]. The set of NEs includes 1,798 types of NEs.

First we annotated the text books with NE tags by using the set of NEs as dictionaries. The annotation was done by the leftmost longest match. Then we replaced each recognized NE in the text books with one of the set of NEs that has the same NE type.

For example, we assume the following result is obtained with the set of NEs;

- ⟨person⟩Ieyasu Tokugawa⟨/person⟩ is the founder of Tokugawa Shogunate.

The "Ieyasu Tokugawa" between the person tag is an NE. Then we replace "Ieyasu Tokugawa" with an NE that type is person in the set of NEs. If "Napoleon" was included in the set of NEs as person and chosen, we generate the following.

- ⟨person⟩Napoleon⟨/person⟩ is the founder of Tokugawa Shogunate.

The sentences in the original text books were used as positive samples and the generated sentences were used as negative samples. As a result, we expect that this solver gives higher scores to choices similar to sentences in the text books. The size of the negative samples was approximately up to 10 times of the positive samples in this experiment. We used AROW [1] for training this solver. Features are combinations of nouns in a sentence.

This solver is applied to texts described in Section 3.2.1. When answering questions, if the question of TFQC is "+", the choice that has the highest score given by this solver as the answer of the question. Otherwise (TFQC is "-"), the choice that has the lowest score as the answer of the question.

5. A SOLVER FOR ANSWERING CHRONOLOGICAL ORDER OF EVENTS

In order to answer the correct chronological order of historical events, we employed an approach that estimates the year in which each event happened and sort historical events with their years in chronological order.

The years in which events happened are estimated by using textbook search engines TYP-N and TTP-N with the following procedure.

- Step1: Extract nouns from choices in a question to be sorted in chronological order with a morphological analyzer and execute OR searches with extracted nouns as queries.

- Step 2: Extract year expressions from top 5 search results of Step 1.

- Step 3: Calculate a weighted mean value of extracted years. Let $\{y_1, ..., y_n\}$ be a set of years and $\{w_1, ..., w_n\}$ be a set of weight values . We define weight w is inverse of search ranking. Then, the weighted mean value $\bar{y}$ is calculated by the following equation.

$$\bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \quad (1)$$

  The calculation result of this equation is used as an estimation of the year of event.

This procedure enables us to sort events in chronological order.

6. EXPERIMENTS

6.1 Solver Selection

We decided a solver to be used for each answer category based on the performance on four years (1997, 2001, 2005, 2009) Center Exam data of the world history B in Japanese.

- The solver trained from virtual examples in Section 4 was used for Answer Categories (AC) 1, 2 and 5.

- The solver based on search results in Section 3 was used for AC 4.

- A solver for answering chronological order of events in Section 5 was used for AC 3.

- For Answer Category 6, we just randomly choose answers from given choices.

Table 3: Scores on the formal run data.

Run	Score	Accuracy
R1	48	19 / 42
R2	41	17 / 42
R3	43	17 / 42

6.2 Experimental Results

Table 3 shows the experimental results on the formal run data. The R1, R2 and R3 indicate the following.

- R1 is the result obtained with the solver selection for questions described in Section 6.1.

- R2 is the result obtained with the solver based on virtual examples described in Section 4 except for AC 3.

- R3 is the result obtained with the solver based on search results described in Section 3 except for AC 3.

We see that the combination of the three solvers contributes to improved accuracy. The score of R1 is 7 points higher than the result obtained with the solver based on virtual examples and 5 points higher than the result obtained with the solver based on search engine results.

7. CONCLUSION

This paper has described our system for National Center Test for University Admissions as a Center Exam subtask. Our system used three solvers for answering different types of questions. The experimental results on the formal run data of QALab showed that a combination of the three solvers contributed to improved accuracy.

8. ADDITIONAL AUTHORS

Additional authors: Tomoya Iwakura†, Yuchang CHENG†, Satoko Shiga†, Masrau Fuji†, Nobuyuki Igata†

9. REFERENCES

[1] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In *NIPS'09*, pages 414–422. 2009.

[2] M. Ishioroshi, Y. Kano, and N. Kando. An analysis of the questions of the university entrance examination to answer using the question answering system (in japanese). In *Proc. of The 27th Annual Conference of the Japanese Society for Artificial Intelligence*, pages 412–415, 2013.

[3] Y. Miyao and A. Kawazoe. University entrance examinations as a benchmark resource for nlp-based problem solving. In *Proc. of IJCNLP2013*, 2013.

[4] S. Okura and A. Ushioda. Developing a semantic structure search system for complex sentences (in japanese). In *Proc. of Annual Meetings of the Association for Natural Language Processing*, pages 412–415, 2014.

[5] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, Dec. 1989.