

# Forst: Question Answering System Using Basic Element at NTCIR-11 QA-Lab Task

Kotaro Sakamoto<sup>\*1, \*2</sup>, Hyogo Matsui<sup>\*1</sup>, Eisuke Matsunaga<sup>\*1</sup>, Takahisa Jin<sup>\*1</sup>, Hideyuki Shibuki<sup>\*1</sup>, Tatsunori Mori<sup>\*1</sup>, Madoka Ishioroshi<sup>\*2</sup>, Noriko Kando<sup>\*2, \*3</sup>

\*1: Yokohama National University, \*2: National Institute of Informatics,

\*3: The Graduate University for Advanced Studies (SOKENDAI)

{sakamoto|m\_hyogo|shin7240|taka\_jin|shib|mori}@forest.eis.ynu.ac.jp, {ishioroshi|kando}@nii.ac.jp

## ABSTRACT

This paper describes Forst's approach to university entrance examinations at NTCIR-11 QA-Lab Task. Our system consists of two types of modules: dedicated modules for each question format and common modules called by the dedicated modules as necessary. Our system uses Basic Element in order to more exactly grasp and reflect the import of questions. We also tackled short-essay questions in the secondary examinations.

## Team Name

Forst

## Subtask

Japanese

## Keywords

question answering, Basic Element, university entrance examination, world history

## 1. INTRODUCTION

Question answering is widely regarded as an advancement in information retrieval. However, QA systems are not as popular as search engines in the real world. In order to apply QA systems to real-world problems we tackle the QA-Lab task dealing with questions from the National Center Test for University Admissions and from the secondary exams at 5 universities in Japan. Most of questions from the university entrance examinations have more complex structure than general QA questions, and require more exact understanding.

Figure 1 shows an example question from the Center Test and the question structure. A question roughly consists of four types of descriptions: instruction in part of exam, context, instruction and answer candidates. Context has various expressions: sentence, word, figure, table and so on. Although all questions of the Center Test are multiple choice, the substance has various question: factoid, true-or-false, fill-in-the-blank and so on. In the case of secondary exams, context may be merged into instruction, and answer candidates may not exist. Figure 2 shows an example question from the secondary exam at Tokyo University. This question is a short-essay question, and the context except words, which must be included in the answer essay, is merged into the instruction.

To answer such various exam questions, we classify exam questions into several question format types, develop modules dedicated to each question format type. We also use Basic Element[1-2] in order to more exactly grasp the import of question sentence. Basic Element is a minimal semantic unit, which is a dependency between words in a sentence and expressed as a triple (head | modifier | relation).

第1問 人類が営む生業と労働は、経済・社会・政治の動きと密接にかかわりながら、大きく変容してきた。生業と労働の歴史について述べた次の文章A~Cを読み、下の問い(問1~9)に答えよ。(配点 25)

A 清の学者顧炎武は、明代の文化人の趨勢を論じて、①唐宋以来、文化・芸術に秀でた者の多くは科学の合格者であったが、②明代になってその担い手は在野の人物に移っていったと述べている。明代中期の画家唐寅は、まさにその過渡期の人物と言える。彼は科擧で優秀な成績を収めながらも、不運な事件に巻き込まれ、榮達の道を絶たれたからは、蘇州で園樂をなりわいとしながら自由奔放な生活を送った。明代中期から後期にかけて、在野の芸術家や文筆家が続々と現れたのは、③江南を中心とする商工業の発展によって都市の文化が成熟し、繪画や出版物が広く商品としての価値を持つようになったからであった。

問1 下線部①に関連して、次に挙げる人物は、いずれも唐代から宋代にかけての科学の合格者である。それぞれの人物について述べた文として正しいものを、次の①~④のうちから一つ選べ。  1

① 歐陽脩や蘇軾は、唐代を代表する文筆家である。  
 ② 顔真卿は、宋代を代表する書家である。  
 ③ 宋の王安石は、新法と呼ばれる改革を行った。  
 ④ 秦焜は、元との関係をめぐり主戦派と対立した。

Figure 1: Example of the Center Test question

第1問

次の文章は日本国憲法第二十条である。

第二十条 信教の自由は、何人に対してもこれを保障する。いかなる宗教団体も、国から特権を受け、又は政治上の権力を行使してはならない。

2. 何人も、宗教上の行為、祝典、儀式又は行事に参加することを強制されない。

3. 国及びその機関は、宗教教育その他いかなる宗教的活動もしてはならない。

この条文に見られるような政治と宗教の関係についての考えは、18世紀後半以降、アメリカやフランスにおける革命を経て、しだいに世界の多くの国々で力をもつようになった。

それ以前の時期、世界各地の政治権力は、その支配領域内の宗教・宗派とそれらに属する人々をどのように取り扱っていたか。18世紀前半までの西ヨーロッパ、西アジア、東アジアにおける具体的な実例を挙げ、この3つの地域の特徴を比較して、解答欄( )に20行以内で論じなさい。その際に、次の7つの語句を必ず一度は用い、その語句に下線を付しなさい。

1. ジズヤ 2. 首長法 3. ダライ=ラマ 4. ナントの王命廃止  
 5. ミット 6. 理藩院 7. 領邦教会制

Figure 2: Example of the secondary exam question

Glossary

(モスクワ大公国) 14-16世紀, モスクワを中心に建国されたロシアの国家。キプチャク=ハン国(1243-1502)の支配下にあったが, じょじょに力をつけてモンゴルからの自立をはかり, 15世紀後半イヴァン3世の時代に独立に成功した。16世紀後半イヴァン4世の時代に最盛期を迎えたが, その死後急速に衰え, 混乱の中で滅亡した。

(イヴァン4世 | 雷帝) 1530-84 モスクワ大公(在位1533-84)。当初は貴族が権力を握っていたが1547年から親政を開始し, ツァーリを名乗った。対立する諸侯をきびしく弾圧して「雷帝」と呼ばれた。イェルマークに命じてシベリア進出に着手した。また, 農奴制を再強化した。

Q&A Collection

- 8-4-1-1-a. 1526年に成立した, インド史上最大のイスラーム国家は何か。(A. ムガル帝国)
- 8-4-1-1-b. ティムールの子孫で, この国の建国者は誰か。(A. パール)
- 8-4-1-1-c. 1526年に彼がロディー朝を破り, 帝国創始のきっかけとなった戦いは何か。(A. パーニーバットの戦い)
- 8-4-1-2-a. 建国者の死後の混乱から, 16世紀後半にムガル帝国を再興した第3代皇帝は誰か。(A. アクバル)
- 8-4-1-2-b. 第3代皇帝がデリーから遷都した町はどこか。(A. アグラ)

Figure 3: Examples of the glossary and the Q&A collection.

## 2. KNOWLEDGE SOURCE

The knowledge of our system was obtained from two textbooks, Wikipedia, a glossary[3] and a Q&A collection[4]. The two textbooks and Wikipedia are given in this task. The glossary and the Q&A collection are used on our own accord. The number of entry words in the glossary is 6,081, and the number of Q&A pairs in the Q&A collection is 4,324. Figure 3 shows examples of the glossary and the Q&A collection. Note that the Q&A collection has a hierarchy and that closer Q&A pairs in the hierarchy have a closer connection in terms of time and space. The glossary and the Q&A collection will produce higher quality knowledge than general Web documents.

## 3. SYSTEM

### 3.1 Outline

Figure 4 shows the outline of our system. Our system mainly consists of dedicated modules for each question format type and common modules that is called by dedicated modules as necessary. In this task, we defined 18 question format types as shown in Table 1. Table 1 also shows dedicated modules corresponding to each question format type. Note that some types are forced to correspond to unsuitable dedicated module. We could not complete all modules planned originally due to shortage of labor.

First, our system classify an input question into question format types as shown in Table 1. The classification is done by clue expressions such as “空欄(blank)” and “正誤(true or false)” in the instruction of the question. Next, the question passes to a dedicated module corresponding to the classified question format type. In this task, we developed the following five dedicated modules: Blank type answering, True-or-false type answering, Factoid type answering, Essay-specifying-words type answering and Essay-no-specifying type answering. The dedicated module collaborates with

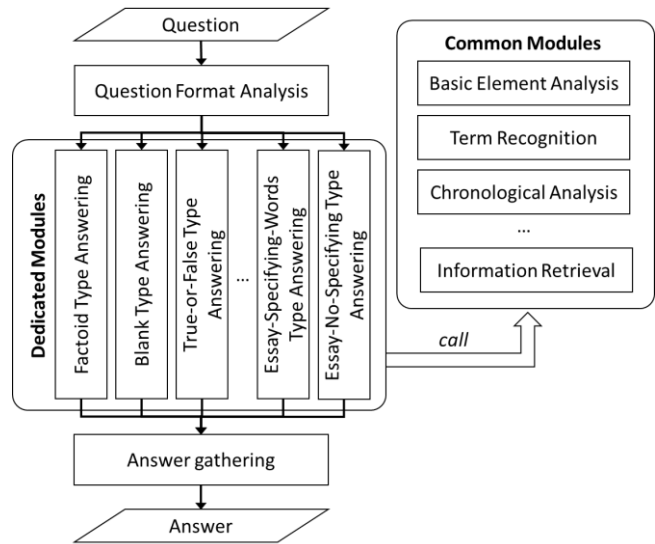


Figure 4: Outline of our system

Table 1: Question format types and the corresponding dedicated modules

| Question format types  | Dedicated modules                    |
|--|--------------------------------------|
| Blank<br>Blank(Combo)<br>Blank+YesNo<br>Graph+Blank  | Blank type answering                 |
| YesNo(True)<br>YesNo(True+Focus)<br>YesNo(False)<br>YesNo(False+Focus)<br>YesNo(Combo)<br>Time<br>Timeline<br>Graph<br>Graph+Other | True-or-false type answering         |
| Factoid(True)<br>Factoid(False)<br>List  | Factoid type answering               |
| Essay(withKeyword)   | Essay-specifying-word type answering |
| Essay  | Essay-no-specifying type answering   |

necessary common modules. The common modules, which are Basic Element analysis, term recognition, chronological analysis and information retrieval in this task, are called by any dedicated modules. Finally, the system arranges answers from each dedicated module in question order.

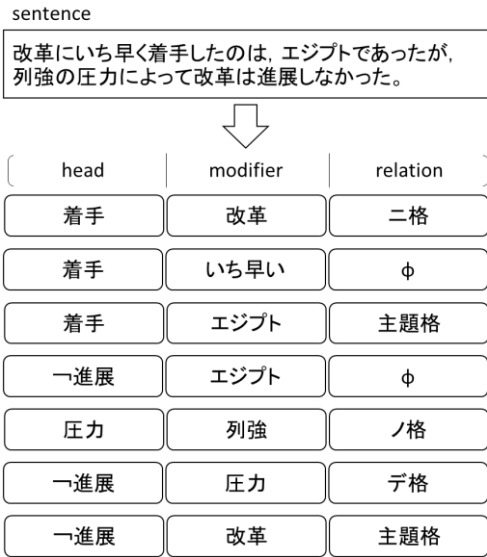


Figure 5: Example of Japanese Basic Element

### 3.2 Common Modules

#### 3.2.1 Basic Element Analysis

The original Basic Element[1-2] is for English texts. In order to apply it to Japanese texts, we developed a Japanese Basic Element analyzer. This module accepts a sentence, and outputs a set of Basic Elements from results of the Japanese part-of-speech and morphological analyzer MeCab[5] and the Japanese dependency structure analyzer CaboCha[6]. Figure 5 shows an example of Japanese Basic Elements. Note that the analyzer guesses unknown word, negative form, modality, and so on. This guess is based on heuristics. Basic Elements are used for similarity calculation. In general, similarity is calculated by feature vectors using the Bag-of-Words representation. We use a Bag-of-Basic-Element representation instead of the Bag-of-Words representation.

#### 3.2.2 Term Recognition

We defined entry words in the glossary as terms. Since a term has an explanation, we use words in the explanation for word expansion. This module accepts a text, and outputs a set of terms with the explanations. First, the module extracts term candidates with exact match. Then, syntactic constraints get rid of inappropriate candidates. Thereby, short length terms such as Chinese dynasty names can be recognized correctly.

#### 3.2.3 Chronological Analysis

This module accepts a text such as a chapter of textbook, and outputs a set of time ranges when each event in the text happens. Figure 6 shows an image of the chronological analysis. This module guesses a time range of a sentence by clue expressions such as “年 (year)” and “世紀(century)”. If there is no clue expression, a time range of last sentence is carried over on the assumption that events written in the input text are in chronological order.

#### 3.2.4 Information Retrieval

We retrieved documents via the search engine Indri [7] based on character-level unigram model.

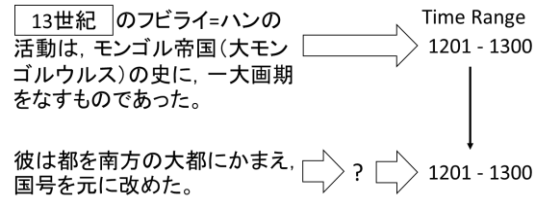


Figure 6: Image of the chronological analysis

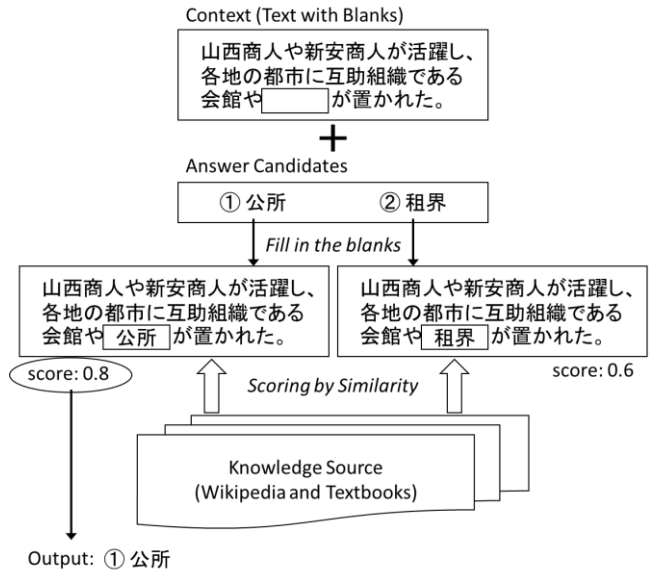


Figure 7: Image of the blank type answering

### 3.3 Dedicated Modules

#### 3.3.1 Factoid Type Answering

This module requires an instruction and an underlined part of context if it exists, and uses the Q&A collection as knowledge source. As the basic idea, we make the module output an answer term of a Q&A pair of which question text is most similar to input text.

As shown in Figure 3, a question text of a Q&A pair is so short that the module may not make a feature vector enough to similarity calculation. Therefore, at first, the module gathers Q&A pairs that have the same path except the last alphabet in the Q&A collection hierarchy, for example, top three pairs from “8-4-1-1-a” to “8-4-1-1-c” shown in Figure 3. The module calculates similarity between a vector from input text and a vector from all texts of gathered pairs. Then, the module calculate similarity between the input text and each question text in the gathered pairs with the highest similarity.

#### 3.3.2 Blank Type Answering

This module requires a text with blanks of context and answer candidates, and uses the given textbooks and Wikipedia as knowledge source. As the basic idea, we make the module output terms if a text with blanks is correct when the blanks are filled with the terms.

Figure 7 shows an image of the blank type answering. First, the module makes other texts by filling the blanks of the input text with words of each answer candidate. Then, the module estimates the

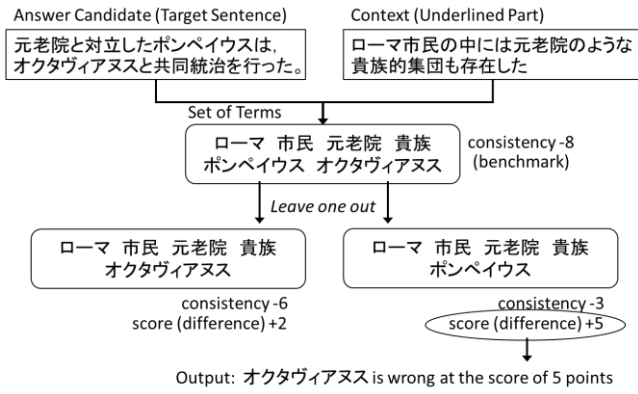


Figure 8: Image of guessing a wrong word.

correctness of each filled text by max similarity to descriptions of Wikipedia or the textbooks. Finally, the module output an answer candidate used for making text with the highest similarity.

### 3.3.3 True-or-False Type Answering

This module requires an answer candidate as a target of the true-or-false judgment and an underlined part of context if it exists, and uses the glossary as knowledge source. In order to judge the truth of answer candidate, the module guesses a wrong word that makes an answer candidate false, and estimates wrong degree of the word. If a wrong degree of an answer candidate is greater than wrong degrees of other answer candidates, the module judge the answer candidate false, and vice versa.

As the basic idea of guessing a wrong word, we makes the module output a term if consistency among terms except for the term increases. Using Basic Elements in term's explanations, a consistency degree between terms is approximated as the inconsistent number from the consistent number. A consistency degree of a term set is approximated as sum of consistency degrees between all term pairs in the set.

Figure 8 shows an image of guessing a wrong word. First, the module extracts terms from an answer candidate and an underlined part of context. Next, the module calculates a consistency degree of the extracted terms as a benchmark. Then, according to the differences between the benchmark and a consistency degree in the case of leaving out each term in the answer candidate, the module guesses a wrong word and the wrong degree.

### 3.3.4 Essay-Specifying-Word Type Answering

This module requires instruction and specified words in context, and uses given textbooks as knowledge source. Note that the specified words must be included to answer essay. As the basic idea, we make the module output a sentence sequence, which includes all specified words, sorted in chronological order. We give the character limit the highest priority because the character limit is so small that there is seldom any choice of alternative sentences.

Figure 9 shows an image of the essay-specifying-words type answering. First, for each word specified in context, the module retrieves all sentences including the word. Next, the module chooses a sentence from each retrieved sentence set so as to maximize the total length of chosen sentences within the character limit. Finally, the module outputs the chosen sentences sorted in chronological order.

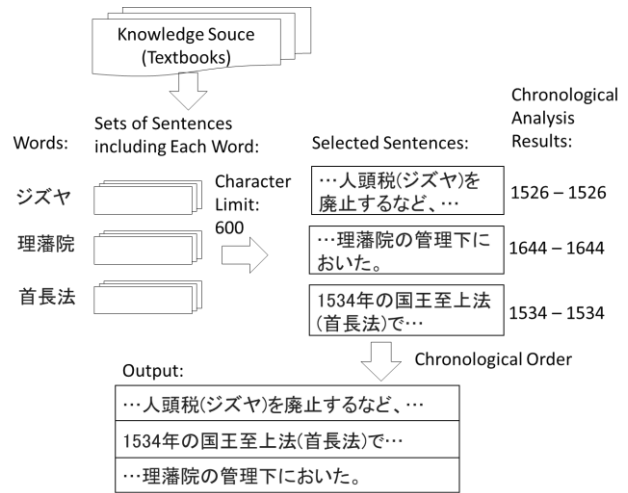


Figure 9: Image of the essay-specifying-words type answering

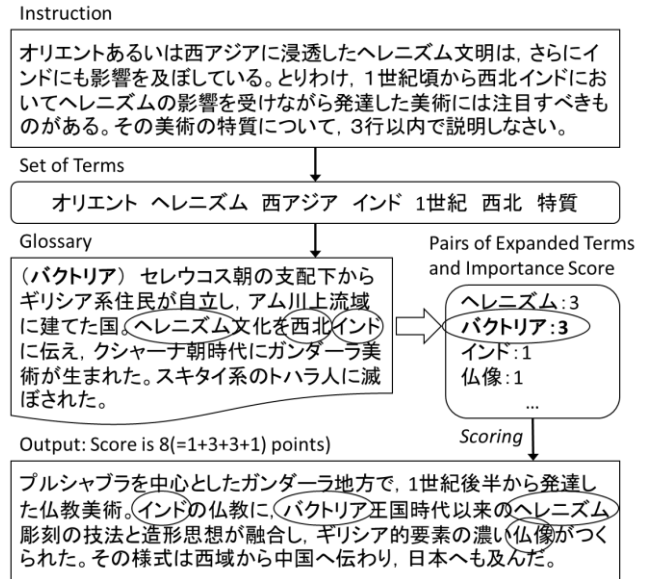


Figure 10: Image of the essay-no-specifying type answering

### Essay-No-Specifying Type Answering

This module requires instruction, and uses the given textbooks and the glossary as knowledge source. As the basic idea, we makes the module output a text including more terms relative to the instruction.

Figure 10 shows an image of the essay-no-specifying type answering. First, the module extracts terms from the instruction. Next, the module counts the type number of the extracted terms included in a term's explanation in the glossary. Using the type number as an importance score of the term, the module approximate a score of a text as the sum of importance score of terms included in the text.

## 4. EXPERIMENTS

Table 3 shows the experimental results of the formal runs. Note that detailed scores our system could not answer correctly are omitted.

**Table 3: Results at the formal runs**

|                        | Total Score          | # of Correct         | # of Incorrect        |
|------------------------|----------------------|----------------------|-----------------------|
| Phase 1 Center Test    | 46                   | 16                   | 19                    |
| Blank                  |                      | 2                    | 0                     |
| Blank(Combo)           |                      | 1                    | 2                     |
| YesNo(True)            |                      | 9                    | 11                    |
| YesNo(False)           |                      | 2                    | 1                     |
| Factoid                |                      | 1                    | 0                     |
| List                   |                      | 1                    | 0                     |
| Phase 2 Center Test    | 49                   | 19                   | 22                    |
| Blank                  |                      | 1                    | 1                     |
| Blank(Combo)           |                      | 1                    | 1                     |
| YesNo(True)            |                      | 12                   | 9                     |
| YesNo(False)           |                      | 3                    | 3                     |
| Graph+Other            |                      | 1                    | 4                     |
| Factoid                |                      | 1                    | 0                     |
| Phase 2 Secondary Exam | <b>ROUGE-1 Score</b> | <b>ROUGE-2 Score</b> | <b>ROUGE-L Score</b>  |
| Short-essay questions  | 0.125                | 0.062                | 0.097                 |
| Essay(withKeyord)      | 0.667                | 0.462                | 0.533                 |
| Essay                  | 0.095                | 0.040                | 0.073                 |
| Phase 2 Secondary Exam |                      | <b># of Correct</b>  | <b># of Incorrect</b> |
| Other type questions   |                      | 46                   | 166                   |

For questions of Center Test, our system achieved 46 points (among 97) in Phase 1 and 49 points (among 100) in Phase 2. For short-essay questions of the secondary exam in Phase 2, the system achieved 0.125 of ROUGE-1 score, 0.062 of ROUGE-2 and 0.097 of ROUGE-L score.

Although the results leave a lot of rooms of improvement, our system could answer all types of question in the university entrance examinations at NTCIR-11 QA-Lab Task. Especially, Essay(withKeyword) type questions could be answered by only our system. Therefore, we steadily took the first step toward applying QA systems to real-world problems.

## 5. CONCLUSION

We reported our work at NTCIR-11 QA-Lab Task. Our system consists of two types of modules: dedicated modules for each question format and common modules called by the dedicated modules as necessary. We used Basic Element in order to more exactly grasp and reflect the import of questions.

Our system could answer all types of question in the university entrance examinations at NTCIR-11 QA-Lab Task. The system achieved 46 points (among 97) and 49 points (among 100) in the Center Test tasks. For short-essay questions of the secondary exam, the system achieved 0.125 of ROUGE-1 score, 0.062 of ROUGE-2 and 0.097 of ROUGE-L score.

We plan to develop suitable modules to each question format type and to improve the existing modules in future work.

## 6. REFERENCES

- [1] L. Zhou, C.-Y. Lin and E. Hovy. 2005. A BE-based Multi-document Summarizer with Query Interpretation. Proceedings of Document Understanding Conference 2005,
- [2] E. Hovy, C. Lin, L. Zhou, J. Fukumoto, “Automated Summarization Evaluation with Basic Elements,” In Proc. of the Fifth Conference on Language Resources and Evaluation (LREC 2006), 2006.
- [3] SekaishiB yougosyu, Yamakawa Shuppansha Ltd.
- [4] Ichimon-ittou SekaishiB yougo-mondaisyu, Yamakawa Shuppansha Ltd.
- [5] <http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>
- [6] <https://code.google.com/p/cabocha/>
- [7] <http://www.lemurproject.org/indri.php>
- [8] <http://akahon.net/>