Forst: Question Answering System for Second-stage Examinations at NTCIR-12 QA Lab-2 Task

Kotaro Sakamoto
Yokohama National University
National Institute of Informatics
Carnegie Mellon University
sakamoto@forest.eis.ynu.ac.jp

Madoka Ishioroshi
National Institute of Informatics
ishioroshi@nii.ac.jp

Hyogo Matsui
Yokohama National University
m_hyogo@forest.eis.ynu.ac.jp

Takahisa Jin
Yokohama National University
taka_jin@forest.eis.ynu.ac.jp

Fuyuki Wada
Yokohama National University
fuyuki@forest.eis.ynu.ac.jp

Shu Nakayama
Tohoku University
Carnegie Mellon University
nakayamas@ecei.tohoku.ac.jp

Hideyuki Shibuki
Yokohama National University
shib@forest.eis.ynu.ac.jp

Tatsunori Mori
Yokohama National University
mori@forest.eis.ynu.ac.jp

Noriko Kando
National Institute of Informatics
The Graduate University for Advanced Studies (SOKENDAI)
kando@nii.ac.jp

ABSTRACT

Japanese university entrance exams have two stages: The National Center Test (multiple choice-type questions) and second-stage examinations (complex questions including terms and essays). We participated in all phases of NTCIR-11 QA Lab-1 and NTCIR-12 QA Lab-2 task's Japanese subtask and our system answered all of the questions. At QA Lab-2 task, we focused on term and essay questions in the second-stage exams and we improved the term type answering and the essay type answering.

Team Name

Forst

Subtask

Japanese

Keywords

question answering, essay questions, university entrance examination, secondary exams, 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 Japanese university entrance exams of world history. Japanese university entrance exams have the following two stages: The National Center Test (multiple choice-type questions) and second-stage examinations (complex questions including terms and essays). QA-Lab task suppied questions of the National Center Test, second-stage examinations, mock exams of the National Center Test and mock exams of second-stage examinations. We participated in all phases of NTCIR-11 QA Lab-1 and NTCIR-12 QA Lab-2 task's Japanese subtask and our system answered all of the questions. Second-stage examinations and mock exams of second-stage examinations include term and essay questions. At QA Lab-2 task, we focused on the second-stage exams, especially term and essay questions. We mainly present the improvements for answering term and essay questions.

2. KNOWLEDGE SOURCE

We used the following data as the knowledge source.

- four world history textbooks which QA Lab organizers supplied
- world history glossary (6,081 entry words)
- Q&A collection (4,324 Q&A pairs)
- world history event ontology[4][1]
- Japanese thesaurus (about 300,000 entry words)

3. NATIONAL CENTER TEST

We developed three systems as the National Center Test solvers. We present the updates from the system in NTCIR-11 QA Lab[3].

- Forst team system at NTCIR-11 QA Lab-1
- baseline system of NTCIR-11 QA Lab-1 task's Japanese subtask

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

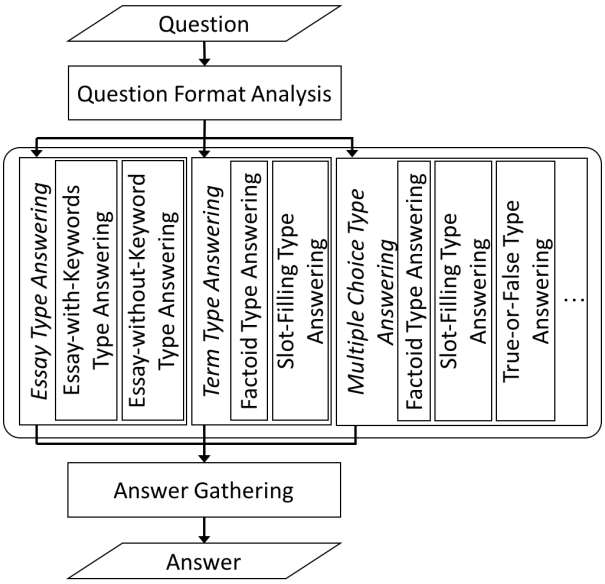

Figure 1: Overall pipeline

- Forst team system updating a module for factoid type question in 4.1 from the module at NTCIR-11 QA Lab-1

4. SECOND-STAGE EXAMINATION

Second-stage examinations has many question types, so we developed a module for each question type and combine the modules vertically. Figure 1 shows the pipeline. We broadly classify question type into two question types of multiple choice type and description type. We adapt the national center test answering to the multiple choice type questions. We classify description type into the two question types of term type and essay type and explain the term type answering and the essay type answering respectively as below.

4.1 Term Type Question

The term type answering module has question analysis, document retrieval, answer candidates extraction and answer selection. Term type question includes factoid type question and slot-filling type question, but we developed factoid type answering and slot-filling type answering as the same module aside from question focus extraction in question analysis. We explain each submodule as below.

4.1.1 Question Analysis

Question analysis module analyzes the input question, extracts the keywords from the question text, judges the number N_a of answers and extracts the lexical answer type. Examples of the lexical answer types are 文字 (character), 飲食物 (food and drink), 部族名 (tribe name), 格言 (proverb), 国際商品 (international commodity), 領土 (territory), etc. Figure 3 shows the list of lexical answer types. In the case of factoid type question, as the lexical answer type, we extract the noun phrase after the postpositional particle "は" in the set of bunsetsu related to interrogative. In the case of slot-filling type question, as the lexical answer type, we extract the noun phrase after the slot. If it is not possible to extract it, we extract the noun phrase before the slot. We use noun and compound noun to extract the keywords from the question. In the case of factoid type question, we take the rule based approach based on the surface expression of question to judge the number N_a of answers. For example, if the question matched the pattern "numeral+(個|つ)+を答えよ", we extracted the number N_a of answers from the numeral. If the question did not match any pattern, we reckoned the number of answers as one.

We detect the semantic answer type from the lexical answer type. We explain how to use the semantic answer type in 4.1.3. The lexical answer type extraction is different between factoid type question and slot-filling type question.

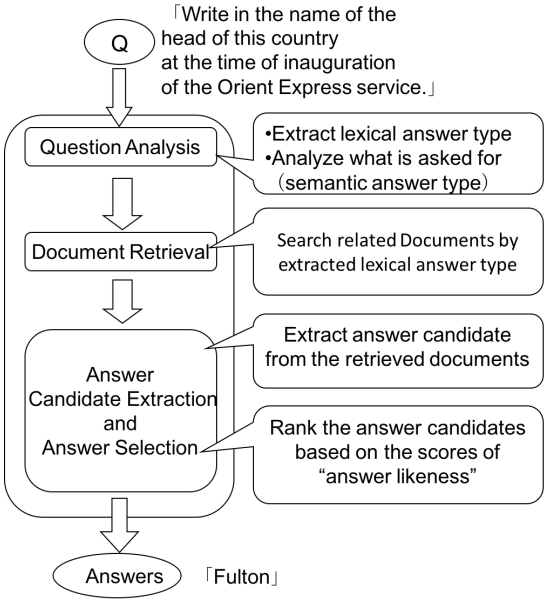

Figure 2: Term type answering pipeline

人物，場所 (国，都市，地域, etc.)，**出来事** (革命，会議，戦争, etc.)，**文明，言語，技術** (発明，文字，道具, etc.)，**時代，建造物** (宮殿，道路，寺院, etc.)，**民族，作品** (小説，詩，絵画, etc.)，**制度** (政策，法令，思想, etc.)，**組織** (同盟，共同体，結社, etc.)，**社会概念** (権利，通貨, etc.)，**宗教，宗教概念** (神, etc.)，**様式，数値**

Figure 3: List of lexical answer types

固有表現クラス, サブクラス, 固有表現, 開始年, 終了年, 異表記, その他の情報 (複数ある場合は@で区切る)
技術, 文字, アラビア文字,, , アラビヤ文字, インド
技術, 文字, キリル文字,900 頃,, ,
技術, 暦, グレゴリウス暦,1582,, グレゴリオ暦, ローマ@グレゴリウス 13 世
人物, 発明家, アークライト,1732,1792,, グレートブリテン王国

Figure 4: Examples of dictionary entries

Factoid type answering extracts the lexical answer type from bunsetsu related to the interrogative in the question text by rule based algorithm with dependency structure. Slot-filling type answering extracts the lexical answer type from two bunsetsu. One of the bunsetsu contains the slot and the other bunsetsu is related to the bunsetsu which contains the slot.

4.1.2 Document Retrieval

A query of document retrieval has the keywords extracted by the question analysis. We use Indri[2] as the document retrieval engine. We indexed world history textbook datasets distributed by QA Lab-2 task organizers in Japanese character uni-gram. Each document in the index is a paragraph in the textbook datasets. The Indri indexing parameters of stemmer, normalize and stopper have nothing because those parameters are for English word uni-gram indexing and not for Japanese character uni-gram indexing.

4.1.3 Answer Candidate Extraction and Answer Selection

The answer candidates extraction module extracts descriptions which can be answer candidates from the retrieved documents. We use noun, noun phrase, named entity as answer candidate.

The answer selection module selects answers from answer candidates from two scores of the similarity S_c between the content of the question and the content of the answer candidate and the relatedness R_a of the answer candidate and the semantic answer type.

The similarity S_c is based on how much frequency the words in the question text appear in the retrieved document containing the answer candidate. The more words in the question appear in the retrieved document containing the answer candidate, the higher the similarity S_c is.

In the instance data, each concept class is classified into a named entity category. We merged some of named entity categories, made a new category and moved the original categories to subcategories of the new category.[3] The instance data has, shown as Figure 4, not only concept class but also starting year, ending year, variants, miscellaneous information, e.g. the name of capital in country class, so we used all of them.

The relatedness R_a is based on how related the semantic answer type and the named entity category of the answer candidate are. We judge the semantic answer type by the lexical answer type as shown in Figure 5. We calculate the relatedness R_a with a dictionary. If the semantic answer type did not matche any of named entity categories of answer candidates, the question has no answer. As the dictionary, we tried two cases of the Japanese thesaurus and the world history named entity glossary. We generated the world history named entity glossary from the instance data of the world history event ontology by human craft. If the answer candidate meets the following criteria, the related ness R_a increases by 1 per each criterion.

- (mandatory) The named entity category of the answer candidate matches the semantic answer type.
- (optional) The subcategories of the named entity category of the answer candidate matches the lexical answer type
- (optional) When the question has the time information, the named entity of the answer candidate has year information.
- (optional) Other information of the named entity of the answer candidate has a keyword of the question.

The answer selection module sorts the answer candidates based on $S_c \times R_a$ in descending order and outputs top N_a answer candidates as answers. If all of the relatedness R_a was zero, The answer selection module sorts the answer candidates based on S_c.

lexical answer type → semantic answer type
領土 (territory) → 場所 (place)
国際商品 (international commodity) → 技術 (technology)
文字 (character) → 技術 (technology)

Figure 5: Transformations from the lexical answer type to the semantic answer type

4.2 Essay Type Question

The essay type answering has eleven modules of question analysis, document retrieval, sentence extraction, sentence compression, sentence grouping, sentence ranking, answer candidate generation, sentence sorting, answer candidate reduction, answer candidate ranking, answer candidate selection. Figure 6 shows the pipeline of the essay type answering. Although the essay type question has essay-with-keywords type question and essay-without-keyword type question, the modules of the essay type answering are same except question analysis. The essay-with-keywords type question has keywords, so we extracted the keywords from the question in the question analysis. However, the essay-without-keyword type question has no keyword, so we added keyword generation instead of keyword extraction in the question analysis. The updates from our Forst team essay type question answering at NTCIR-11 QA Lab[3] are keyword generation and sentence compression. Also, we updated them in phase-3, so in phase-1 and -2, we used the same system as essay type answering at NTCIR-11 QA Lab. We explain the updated modules of keyword generation and sentence compression as follows.

4.2.1 Keyword Generation

The term type answering shown in 4.1 extracts keywords from the texbooks. We give the essay-without-keyword type question the term type answering and recieve the term ranking as answer ranking of the term type answering. We give the term ranking as keywords the essay-with-keywords answering. If we recieve no answer, we remove the lowest

[2]http://www.lemurproject.org/indri.php
[3]We moved 人物 (person) to a subcategory of 職業 (job)

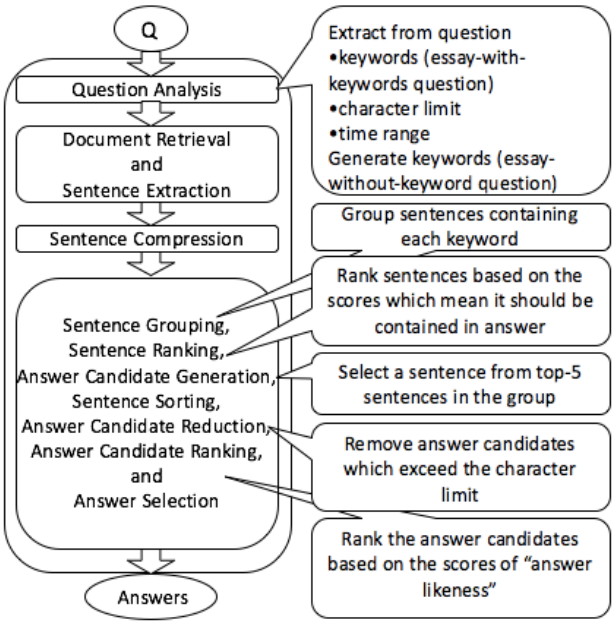

Figure 6: Essay type answering pipeline

scored term from the term ranking and give the new term ranking as keywords the essay-with-keywords answering. We continue removing the lowest scored term from the term ranking until we recieve an answer.

4.2.2 Sentence Compression

The sentence compression module inputs a sentence and extracts all of propositions consist of bunsetsu in the sentence. We compose summaries focusing on the existance of each proposition. For example, the sentence S has a set of propositions $\{P1, P2, P3\}$. From the power set of it, we get six subsets except the empty set and the universal set. The subsets are $\{P1\}$, $\{P2\}$, $\{P3\}$, $\{P1, P2\}$, $\{P2, P3\}$ and $\{P3, P1\}$. From the sentence and the subsets, we want six compressed sentences by making a sentence from each subset. However, if we compress a sentence by combining multiple bunsetsu of each proposition, the compressed sentence basically goes so unnatural. So we compress a sentence by removing bunsetsu of each subset from the sentence.

We automatically extract all of propositions from the sentence. The proposition is a group of three kinds of bunsetsu "predicate bunsetsu", "bunsetsu related to predicate bunsetsu" and "bunsetsu which has a parallel structure to bunsetsu related to predicate bunsetsu". We use predicate, parallel structure and dependency from the output of ChaPAS[4] to extract the proposition bunsetsu from the sentence. We compress a sentence by removing propositions as groups of the extracted proposition bunsetsu. For example, the sentence S has propositions $\{P1, P2, P3\}$. If we want the compressed sentence containing just $\{P1\}$ by removing $\{P2, P3\}$ from the sentence, we parse the sentence to a list of bunsetsu. We remove bunsetsu of $P2$ and bunsetsu of $P3$ from the list of bunsetsu of the sentence. We combine the rest of bunsets in the list and get the compressed sentence.

5. RESULT

[4]https://sites.google.com/site/yotarow/chapas

Table 5 shows the formal run evaluation results of term type questions. In Phase-2 and -3, we have improved the term type answering from the Forst team system at NTCIR-11 QA Lab, written in Section 4.1 and the accuracies increased.

In Phase-3, the Priority-2 essay type answering system includes the sentence compression module of Section 4.2.2. Table 2 shows the formal run evaluation results of the complex-essay-with-keywords type questions. Hardly there is any difference in the ROUGE scores of the Priority-2 comprex essay type answering system and other systems. Table 3 shows the formal run evaluation results of the complex-essay-without-keyword type question. Table 4 shows the formal run evaluation results of the simple-essay type questions. As a result of the sentence compression, the number of no answers of the Priority-2 system with the sentence compression are less than the number of no answers of the Priority-1 and -3 systems without the sentence compression. The character limits of short essay type questions are mostly 30 to 90 characters, but mostly appropriate sentences for answers in knowlege sources have more characters. The sentence compression generates shorter sentences than the character limit. That is why the Priority-2 system could answer more questions than the Priority-1 and -3 systems.

Table 1 shows the formal run evaluation results of the multiple choice type questions.

6. CONCLUSION

Our system answered all of the questions in the Japanese subtask and improved the term type answering and the essay type answering. In term type questions, the accuracies increased by the proposed term type answering. In simple essay type questions, the proposed sentence compression module let the number of no answers decrease. In complex essay type questions, hardly there was any difference in the ROUGE scores whether the system had the sentence compression or not. The sentence compression generates many compressed sentences from one sentence and the high-ranked sentences are often the compressed sentences generated from the same sentence. The answer selection often needs to select an answer from similar answer candidates generated from the same sentence. We need to improve the sentence compression as generating one compressed sentence from one sentence to give the answer selection answer candidates generated from various sentences.

7. REFERENCES

[1] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. Overview of the NTCIR-11 QA-Lab Task. Proceedings of the 11th NTCIR Conference, 2014.

[2] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. Overview of the NTCIR-12 QA Lab-2 Task. Proceedings of the 12th NTCIR Conference, 2016.

[3] Kotaro Sakamoto, Hyogo Matsui, Eisuke Matsunaga, Takahisa Jin, Hideyuki Shibuki, Tatsunori Mori, Madoka Ishioroshi, and Noriko Kando. Forst: Question Answering System Using Basic Element at NTCIR-11 QA-Lab Task. Proceedings of the 11th NTCIR Conference, 2014.

[4] Kawazoe, A., Miyao, Y., Matsuzaki, T., Yokono, H., Arai, N (2014). "World History Ontology for Reasoning Truth/Falsehood of Sentences: Event Classification to Fill in the Gaps between Knowledge Resources and Natural Language Texts," In Nakano, Yukiko, Satoh, Ken, Bekki, Daisuke (Eds.), New Frontiers in Artificial Intelligence (JSAI-isAI 2013 Workshops), Lecture Notes in Computer Science 8417, pp.42-50.

Table 1: Formal run results of multiple choice type

		Priority	Total Score	# of Correct	# of Incorrect	Accuracy
Phase1						
	Center Test	1	31	13	23	0.36
		2	22	9	27	0.25
	Benesse	1	29	11	25	0.31
		2	26	9	27	0.25
	Yozemi (2012)	1	20	7	29	0.19
		2	38	13	23	0.36
	Yozemi (2013a)	1	41	14	22	0.39
		2	32	11	25	0.31
Phase2						
	Benesse	1	62	22	41	0.35
		2	74	28	35	0.44
		3	67	23	40	0.37
Phase3						
	Center Test	1	42	15	21	0.42
		2	48	17	19	0.47
		3	38	13	23	0.36
	Benesse	1	44	16	29	0.36
		2	38	14	31	0.31
		3	29	11	34	0.24
	Yozemi (2014a)	1	46	16	20	0.44
		2	31	11	25	0.31
		3	29	11	25	0.31
	Yozemi (2013d)	1	26	9	27	0.25
		2	51	18	18	0.5
		3	28	11	25	0.31

Table 2: Formal run results of complex-essay-with-keywords type

Complex Essay with Keywords		Priority	ROUGE-N 1	ROUGE-N 2	# of N/A
Phase1					
	Secondary Exams	1	0.525	0.163	0/4
		2	0.5	0.156	0/4
		3	0.472	0.161	0/4
	Sundai	1	0.502	0.138	0/1
		2	0.546	0.159	0/1
Phase2					
	Sundai	1	0.541	0.19	0/1
		2	0.54	0.196	0/1
		3*	0.54	0.196	0/1
		4*	0.595	0.293	0/1
		5*	0.567	0.257	0/1
Phase3					
	Secondary Exams	1	0.592	0.19	0/5
		2	0.547	0.164	0/5
		3	0.592	0.187	0/5
	Sundai	1	0.486	0.128	0/1
		2	0.487	0.131	0/1
		3	0.486	0.128	0/1

* including manual intervention.

Table 3: Formal run results of complex-essay-without-keyword

Complex Essay without Keyword		Priority	ROUGE-N 1	ROUGE-N 2	# of N/A
Phase1					
	Secondary Exams	1	0.457	0.137	0/6
		2	0.454	0.135	0/6
		3	0.457	0.14	0/6
Phase3					
	Secondary Exams	1	0.398	0.111	0/5
		2	0.409	0.11	0/5
		3	0.41	0.115	0/5

Table 4: Formal run results of simple-essay type

Simple Essay		Priority	ROUGE-N 1	ROUGE-N 2	# of N/A
Phase1					
	Secondary Exams	1	0.225	0.0483	1/15
		2	0.219	0.0406	1/15
		3	0.0927	0.0219	1/7
	Sundai	1	0.204	0.0286	1/5
		2	0.204	0.0286	1/5
Phase2					
	Sundai	1	0.204	0.0202	0/5
		2	0.204	0.0202	0/5
		3*	0.342	0.0716	0/5
		4*	0.345	0.0806	0/5
		5*	0.392	0.0997	0/5
Phase3					
	Secondary Exams	1	0.208	0.0362	7/21
		2	0.25	0.0433	3/21
		3	0.155	0.0249	11/21
	Sundai	1	0.171	0.026	2/4
		2	0.18	0.0367	2/4
		3	0.171	0.026	2/4

* including manual intervention.

Table 5: Formal run results of term type

other type questions		Priority	# of Correct	# of Incorrect	# of N/A	Accuracy
Phase1						
	Secondary Exams	1	28	155	34	0.13
		2	10	97	6	0.088
		3	6	69	6	0.074
	Sundai	1	0	10	0	0
		2	0	10	0	0
Phase2						
	Sundai	1	8	2	0	0.8
		2	5	5	0	0.5
		3	7	3	0	0.7
		4	8	2	0	0.8
		5	8	2	0	0.8
Phase3						
	Secondary Exams	1	47	96	2	0.32
		2	35	102	8	0.24
		3	47	96	2	0.32
	Sundai	1	5	5	0	0.5
		2	3	7	0	0.3
		3	5	5	0	0.5