# NTT's Question Answering System for NTCIR-6 QAC-4

**Ryuichiro Higashinaka** and **Hideki Isozaki**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{rh,isozaki}@cslab.kecl.ntt.co.jp

## Abstract

*NTCIR-6 QAC-4 organizers announced that there would be no restriction (such as factoid) on QAC4 questions, but they plan to include many* 'definition' *questions and* 'why' *questions. Therefore, we focused on these two question types. For 'definition' questions, we used a simple pattern-based approach. For 'why' questions, hand-crafted rules were used in previous work for answer candidate extraction [5]. However, such rules greatly depend on developers' intuition and are costly to make. We adopt a supervised machine learning approach. We collected causal expressions from the* EDR *corpus and trained a causal expression classifier, integrating lexical, syntactic, and semantic features. The experimental results show that our system is effective for 'why' and 'definition' questions.*

## 1 Introduction

Our QAC-4 system **NCQAW** (**N**TT **C**S Labs' **Q**uestion **A**nswering System for **W**hy Questions) is based on SAIQA-QAC2, our factoid question answering system [2]. Although SAIQA-QAC2 can answer some 'definition' questions and 'why' questions by using ad hoc rules, its performance for these question types has been poor. We modified the answer extraction module and the answer evaluation module for these question types to improve the performance.

In Sections 2 and 3, we describe the answer extraction and evaluation modules for 'definition' and 'why' questions in NCQAW. After briefly describing how we deal with 'how' questions in Section 4, Section 5 presents the results of our system for the QAC-4 formal run. Section 6 analyzes errors made by our system, and Section 7 summarizes and mentions future work.

## 2 'Definition' questions

### 2.1 Answer Candidate Extraction

We use a simple pattern-based approach. Given a phrase $X$, the system generates typical definition patterns for $X$ in a manner similar to Joho et al. [3]. For instance, '$Y$ such as $X$' and '$Y(X)$' are such patterns.

When one of these patterns matches a sentence, $Y$ becomes a candidate definition of $X$. Although SAIQA-QAC2 used some of these patterns, it simply considered noun phrases as $Y$. Therefore, the extracted $Y$ was sometimes too short to be informative as a definition. To solve this problem, we focus on the dependency structure of the patterns and extend them to match modifiers of all words expressed in the pattern. For example, when $X$ is 'cats', the pattern '$Y$ such as cats' matches 'pet animals such as cats' with $X$ = 'pet animals' and $Y$ = 'cats'.

To allow this matching, we first fill $X$ of the patterns with the definition target; e.g., 'cats'. Then, we create dependency trees for them using *CaboCha*.[1] Finally, We search for these tree patterns through documents by using a tree-based search program **tgrep2**[2] to obtain the matching trees. Since modifiers are allowed to be included in the matched results, the length of $Y$ can be long, overcoming the shortcomings of SAIQA-QAC2. The current system has 13 patterns, including one that simply regards any modifiers of $X$ as $Y$, which principally looks for *rentai* (adnominal modification) or *renyou* (adverbial modification) clauses of $X$.

### 2.2 Answer Evaluation

We evaluate each candidate $C$ by the sum of the scores of content words in $C$. That is,

$$\mathrm{candscore}_{\mathrm{def}}(C) = \sum_{w \in \mathrm{CW}(C)} \mathrm{wordscore}_{\mathrm{def}}(w)$$

where $\mathrm{CW}(C)$ is the set of content words (verbs, nouns, and adjectives) in $C$.

These candidates share many words that are useful to define the specified phrase $X$. It is reasonable to expect that a content word shared by many candidates indicates a better definition than another word shared by only a few candidates. Therefore, we define the

[1] http://chasen.org/~taku/software/cabocha/
[2] http://tedlab.mit.edu/~dr/TGrep2/index.html

word score by the log of the count (term frequency without any normalization) of the word $w$ in the set of all candidates $\{C_i\}$ found by tgrep2.

$$\mathrm{wordscore}_{\mathrm{def}}(w) = \log(\mathrm{tf}(w; \{C_i\}))$$

# 3 'Why' questions

## 3.1 Answer Candidate Extraction

Since we thought it would be difficult to find exact answers for 'why' questions, we used sentences as the unit of answer candidates ($C$), making the task a sentence extraction problem. We regard sentences having at least one of the query words as answer candidates since evaluating all sentences in the top-ranked documents would be computationally expensive in the answer evaluation stage as we explain later.

## 3.2 Answer Evaluation

We evaluate each candidate by using the following two scores:

(a) certainty of the existence of a *causal expression* in the candidate,
(b) *similarity* of the question and the candidate.

The final score is determined based on these scores.

Suppose the system extracted three answer candidates for a question.

Q. Why did John steal the cake?
$C_1$ John was hungry.
$C_2$ John did it because he was hungry.
$C_3$ John stole the cake because he was hungry.

Then, $C_2$ is preferred to $C_1$ because $C_2$ has a causal expression 'because he was hungry' whereas $C_1$ does not. $C_3$ is preferred to $C_2$ because $C_3$ share more words with the question than $C_2$.

**Causal expression** For (a), although hand-crafted rules were used in previous work for answer candidate extraction [5], such rules greatly depend on developers' intuition and are costly to make. Therefore, we adopt a supervised machine learning approach. First, we build a classifier that determines whether a sentence contains a causal expression. For this, we use the *EDR* corpus[3] for obtaining the training samples. Sentences in the corpus have annotations for *causal* expressions by 'cause' tags. Sentences with causal expressions are considered positive examples, while those without causal expressions are considered negative examples.

We first analyzed each sentence in the EDR corpus by CaboCha for word segmentation and dependency and added word sense tags by using Nihongo Goi-Taikei [1]. Then, we built for each sentence a tree that integrates these lexical, semantic, and syntactic features. We employed **BACT**, a tree-based boosting algorithm [4], to train the classifier. Some 'why' questions might request *purpose* (e.g., why do you want to…?). Although the EDR corpus has 'purpose' tags, we did not use them because not all purposes can be answers to 'why' questions. We use BACT's output score ($\mathrm{causal}_{\mathrm{why}}(C)$) as the certainty of the existence of a causal expression in the sentence.

**Similarity** For (b), we used a simple idf (inverse document frequency) score given by the log of the inverse ratio of the number of documents that contain the specified query word $w$. That is,

$$\mathrm{sim}_{\mathrm{why}}(S) = \sum_{w \in Q(S)} \mathrm{idf}(w)$$

where $Q(S)$ is the set of query words in the sentence $S$. Since the number of sentences is generally large, sentence classification by BACT can be sometimes computationally expensive, hence the removal of answer candidates without any query words in answer candidate extraction. Another justification for the removal is that such sentences with no query words have similarity scores of zero, meaning completely irrelevant to the question.

We normalize the above sentence score by a sigmoid function.

$$\mathrm{sim}'_{\mathrm{why}}(C) = 1/(1 + \exp(-\mathrm{sim}_{\mathrm{why}}(C))$$

The final answer ranking is determined by a heuristic function combining the two scores.

$$\mathrm{candscore}_{\mathrm{why}}(C) = \mathrm{causal}_{\mathrm{why}}(C) + \mathrm{sim}'_{\mathrm{why}}(C)$$

.

# 4 'How' questions

We also applied the supervised machine learning method to 'how' (procedural) questions. Although it was not clear which tag in the EDR corpus corresponds to procedures, we used 'condition' tags because we found, through mining the corpus, that some procedural expressions are likely to appear just after conditional expressions. For example, in the sentence 'If the Olympic flame goes out, it gets re-ignited.', 'If the Olympic flame goes out' indicates a condition and 'it gets re-ignited' a procedure. The answer candidate extraction and evaluation processes are exactly the same as those for 'why' questions except for the change of the tag used in obtaining training samples for BACT.

[3] http://www2.nict.go.jp/r/r312/EDR/index.html

## 5 Results

The QAC-4 question set consists of 100 questions of varying question types. To answer the questions, the system first identified their question types by hand-crafted rules. Then, 20 most related documents from the Mainichi Newspaper (1998–2001) were retrieved for each question using Decayed IDF (DIDF) [2]. The documents were sent to the answer extraction and answer evaluation modules depending on the question type. For 'definition', 'why', and 'how', the modules mentioned in previous sections were utilized.

NTCQAW1 in Table 1 shows the organizers' judgments for our system's outputs. As a baseline, we submitted the output of another system NTCQAW2 that uses SAIQA's ad hoc rules for 'why' and 'how' questions. We did not change processing of 'definition' and 'other' question types. The ad hoc rules search general key phrases such as *"riyuu (reason)", "mokuteki (purpose)", and "niyori (because of)"*.

Since we submitted the top five answers for each question, we received five scores for each question. The table summarizes the results using only the best score among the five where A–D stand for the following:

- A: The system output is a correct answer.
- B: The system output has a correct answer but also contains other statements.
- C: The system output has a part of a correct answer.
- D: The system output is an incorrect answer.

It is difficult to decide yet how good the performance was because the evaluation of non-factoid question answering systems is still an open question. One noticeable difficulty we encountered was the classification of question types, which resulted in the large number of questions left unanswered. Note that question types in the table are derived by our manual classification.

We focused mainly on 'definition' and 'why' questions, and we believe NTCQAW1's results for these question types are relatively good. For 'definition' questions, the NTCQAW1 system returned correct answers for 9 out of 24 questions (38%). For 'why' questions, the NTCQAW1 system returned correct answers for 17 out of 38 questions (45%). On the other hand, NTCQAW2 answered only 9 'why' quesions.

## 6 Error Analysis

In this section, we analyze errors of the NCQAW1 system.

**Table 1. Performance of NCQAW**

| NTCQAW1 | | | | | | |
|---|---|---|---|---|---|---|
| question type | all | A | B | C | D | no output |
| definition | 24 | 9 | 0 | 0 | 11 | 4 |
| other | 12 | 1 | 0 | 0 | 2 | 9 |
| why | 38 | **11** | **6** | 0 | 18 | 3 |
| how | 26 | 1 | 1 | 1 | 7 | 16 |
| total | 100 | 22 | 7 | 1 | 38 | 32 |
| NTCQAW2 | | | | | | |
| why | 38 | **5** | **3** | 1 | 25 | 4 |
| how | 26 | 1 | 0 | 1 | 9 | 16 |
| total | 100 | 16 | 3 | 2 | 46 | 32 |

### 6.1 Errors of the question type analysis

The system extracts answer candidates for each question types. If the question analyzer does not produce any question type, the system does not output any candidates. Because of the low coverage of our rule-based question analyzer, the system failed to assign any question type to 30 questions. Most of the failures were caused by expressions such as donna, douiu, and donoyouna corresponding to 'what kind of' in English. As a result, the system could answer at most 70 questions. There is clearly a need to improve our question analyzer.

### 6.2 Errors of 'definition' questions

The following two cases show typical erroneous candidates for 'definition' questions.

- QAC4-00036-00: What is the wiretapping law?
  Typical erroneous candidate: laws of ..., national flag and national anthem law, wiretapping law
- QAC4-00054-00: What does the Court of Arbitration for Sport do?
  Typical erroneous candidate: The headquarter of the Court of Arbitration for Sport is located in Lausanne.

In the first case, an extraction pattern for 'definition' matched to a list expression. The system should accurately distinguish adnominal modifications from lists. In the second case, the candidate is indeed a definition statement, but does not answer the question, i.e., the candidate describes the location of the Court but does not state its role. This leads us to believe that 'definition' is not a single question type but a set of question types.

### 6.3 Errors of 'why' questions

- QAC4-00061-00: What is the reason for the success of the prediction of Mt. Usu eruption?

Typical erroneous candidate: Since the activity of Mt. Usu is expected to continue for five or ten years, we should take care of refugees.

In this case, the candidate has a causal expression and query words, but the candidate does not answer the question. Instead of our similarity function based on simple bag-of-words statistics, there is a need to take into account richer information such as semantic roles in order to get higher accuracy in answer candidate evaluation.

### 6.4 Errors of 'how' questions

For 'how' questions, our results were not satisfactory. There are two reasons for this. First, as we have already mentioned in Section 6.1, we did not prepare enough patterns for 'how' questions for question analysis. Therefore, the system frequently failed to determine the question type. In fact, the system's question type was correct for only 4 of the 26 'how' questions. Second, we could not determine appropriate EDR tags for 'how' questions. We are currently investigating how other tags of the EDR corpus may be useful for 'how' questions and are also considering the possibility of exploiting other resources.

## 7 Summary

We presented the overview of NTT's Question Answering System for QAC-4. We implemented a beyond-factoid question answering system called NC-QAW by improving SAIQA-QAC2's answer extraction module and answer evaluation module for 'definition', 'why', and 'how' questions. We used a simple pattern-based approach for 'definition' with an extension to include modifiers. For 'why' and 'how' questions, we adopted a supervised machine learning approach using the EDR corpus to obtain training examples. Consequently, we achieved good performance for 'definition' and 'why' questions.

## References

[1] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. *Goi-Taikei – A Japanese Lexicon*. Iwanami Shoten, 1997.

[2] H. Isozaki. An analysis of a high-performance Japanese question answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(3):263–279, 2005.

[3] H. Joho, Y. K. Liu, and M. Sanderson. Large scale testing of a descriptive phrase finder. In *HLT-2001*, pages 645–649, 2001.

[4] T. Kudo and Y. Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proc. EMNLP*, pages 301–308, 2004.

[5] K. Morooka and J. Fukumoto. Answer extraction method for why-type question answering system. In *IEICE Technical Report*, volume 105, pages 7–12, 2006. (in Japanese).