Oki QA System for QAC-1

Atsushi IKENO, Hiroyuki OHNUMA
Service Media Laboratory, Corporate Research & Development Center
Oki Electric Industry, Co., Ltd.
1-2-27 Shiromi, Chuo-ku, Osaka 540-6025, Japan
ikeno546@oki.com, ohnuma838@oki.com

Abstract

This paper describes Oki QA system for QAC-1. Our system has been designed to return a swift response, and obtained a satisfactory 3-5 sec/question result, although the scores are not high enough due to misretrieval at IR stage. We have conducted an experiment about score accumulation on task 1, and found it effective for some of the questions. We also have tried a time-saving strategy on task 3, but the result did not clarify the effect of it.

Keywords: *Information Extraction, Named Entity, Information Retrieval, Question Analysis, Score Accumulation*

1 Introduction

Our system is initially oriented for a real-time response system with much larger document set as knowledge. We, therefore, have adopted a simple structure and algorithm for this attempt. That is, no original information retrieval module, no thesaurus to interpret questions, no complicated score calculation.

Section 2 shows the overview of our system, and the answer selection algorithms for task 1 and 2 are explained in section 3. In section 4, the arrangement of the system and the strategy for task 3, followed by results and discussions in section 5, and conclusion in section 6.

2 System Description

The approach of our system is "Information Retrieval(IR)" + "Information Extraction(IE)." Documents that likely to contain the answers to a question are retrieved first by using the words in the question. Answer candidates are then extracted from the documents, and the output of the system is chosen among them according to the scores.

Figure 1 shows the architecture of Oki QA system for QAC-1. The followings are the description of the modules in detail.

2.1 Named Entity Tagging

All the articles are preprocessed by Named Entity(NE) tagger based on the module of Oki's MET-2 system [1]. It inserts NE tags in such a format as XML tags. It is a pre-set surface pattern based tagger and does not use morphological analysis or parsing.

The NE tagged articles are stored to be referenced when the answer candidates are extracted.

Since the answers of the questions to the system are chosen from these NEs, the ability of the NE tagger is one of the important factors of our system.

2.2 Question Analysis

Question analysis module recognizes the answer type that questions demand according to the interrogatives and the words around. It also extracts search keywords for IR. A question is morphologically analyzed by Chasen [2], then the answer type and the search keywords are determined based on the word strings and POS(Part Of Speech)s.

2.3 Query Construction

Some search keywords are better not included in a query to retrieve desirable documents. Query construction module removes such keywords and forms an appropriate query to hand over to the IR module.

Because it does not employ thesauruses of any kind, the query elements are basically the exact strings appeared in the input question, even when they are misspelled. Verbs are omitted from the queries, while adjectives and adverbs remain.

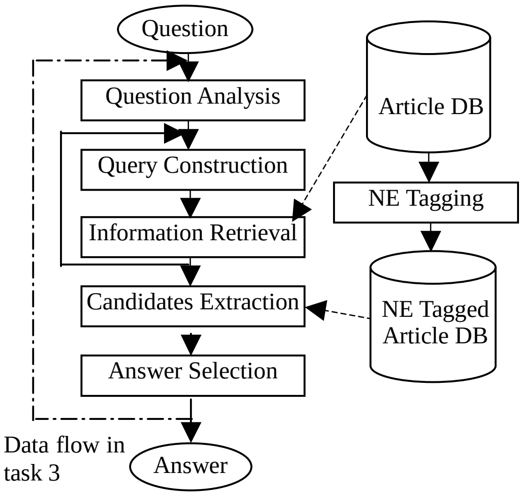

Figure 1. System Architecture.

2.4 Information Retrieval

Namazu with Kakasi index [3] is used to retrieve documents in this system. Top 20 documents are adopted for later processes.

When no documents are obtained with the initial query, it is reconstructed at the query construction module by deleting one of the elements, and retrieval is tried again with the new query. This process is repeated recursively till more than one document is found. If the query runs short of its elements without finding any documents, then the system's answer is set to empty.

The entire system depends a lot on this IR module, because the answers to a question are extracted only from the documents obtained here.

2.5 Candidates Extraction

From the retrieved documents, "basic units" [4] which contain search keywords are identified. Within the units, all the NEs whose attributes coincide with the answer type on target are extracted as answer candidates.

Each candidate is weighed according to the (length of the) keywords in the unit.

The score does not reflect the distances among keywords or the ones between a candidate and keywords in this system.

2.6 Answer Selection

Answers are basically selected based on the scores of the candidates (and the document rank that the IR module returns for task 2). We have not taken too much effort on excluding inappropriate candidates. Details are shown in the following section.

3 Answer Selection Algorithms

We have taken different algorithms to select answers for task 1 and 2. No threshold is required in both algorithms, which results in no threshold adjustment.

3.1 Score Accumulation Algorithm

Based on the assumption that the most probable candidate must be appeared frequently, the candidates' scores of the same word strings are accumulated for task1. The candidate strings are sorted with the accumulated scores, and top 5 on the list are selected as the answers. Support information is provided from the one of the candidates'.

3.2 Highest Score Document Algorithm

As just a small trick to exclude false answers, the candidates that hold the highest score are selected for task2.

When more than one answers are selected, these 2 rules below are applied as the appeared order to complete answer selection.

(1) If they are from different documents, the ones only from the highest ranked document by IR module are selected.

(2) If they are within the same "basic unit" in the same document, all of them are selected as the answers.

4 System Arrangement for Task 3

To answer the succeeding questions in a series of questions task(task 3), the system must be arranged to handle the information of their preceding questions.

Our strategy to explore the answers for the succeeding questions is explained below. Task 2 answer selection is adapted as basic algorithm.

4.1 Question Completion

Succeeding question usually has insufficient information itself, because there is an implicit understanding that the information provided by the preceding question is still available.

We have assumed that the answer of the preceding question is the topic of the succeeding question if there is no demonstrative in the succeeding. The preceding answer is included as a missing word in the succeeding question (and query) in this case.

If there is a demonstrative, it is replaced by the word it refers, which is supposed to be in the preceding question or answer. The referred word is identified according to the types of the demonstrative.

4.2 Search Space Limitation

We first limit the search space(document set) to save time, then broaden it gradually. Once any answers are found, they are returned as the system's answers. The search space is set in the order shown below.

(1) The document that contains the answer of the preceding question.
(2) All the documents retrieved in the process of the preceding question.
(3) The resulted documents of the IR from the entire articles with the completed succeeding question.

5 Results and Discussions

The resulting scores of our system are lower than we expected. Evaluation of our experimental algorithms and strategy is described below, with failure analysis discussion.

5.1 Task 1 Analysis

Table 1 shows the main cause lead to the wrong answers. The weak point of this system is the performance of the IR module.

The installed module had a problem when search keywords are year and common country name such as "２０００年(the year 2000)" or "日本(Japan)." We also noticed that the inconsistency of the word unit between the question analysis module, which divides sentences with Chasen, and IR module, whose indexing is based on Kakasi.

Table 1. Reasons of failure in task 1.

Information Retrieval	44%
Question Analysis	30%
Named Entity Tagging	17%
Other	9%

The results will be improved for IR reasoned failures by almost 60%, which leads to about 20% up for recall value, if the IR system can obtain the desired documents.

The failures in question analysis and NE tagging are inevitable because the system did not expect to process those words and had no knowledge about them at the time. That should be covered by labor or automatic learning, which we did not focus on this time.

Score Accumulation generally worked for improving the recall in this case. We have to be careful, however, that the articles of the almost the same contents are included and in fact retrieved, this algorithm may not be suitable for the document set which holds such duplicated contents.

5.2 Task 2 Analysis

Although we lost some of the correct answers captured in task 1 by adapting other algorithm, it worked rather effectively for excluding the wrong answers, which is beyond our expectation.

Since the IR problem discussed above still holds here and much more effect on the result considering the algorithm, the score has high chance of improving by the slight adjustment of IR module.

5.3 Task 3 Analysis

The resulting score was far from good, but this does not mean our strategy is failed, because, here again, the IR result becomes the problem. The succeeding answers should apparently fall on wrong with the wrong preceding answers.

We have checked the correct answers and found that correct answers to only 5 out of 40 succeeding questions are in completely different documents from the preceding answers' documents. This may prove the search space limitation proposed above is one of the ideal strategies.

We are still working on checking our question completion technique with the correct answer. Further discussion will be made at the workshop.

5.4 Response Time

The time required to obtain an answer varies due to recursive IR, but the average is approximately 3 to 5 seconds per question in every task performed on Pentium III 800MHz PC. This result is satisfactory because real-time response can be achieved with some programming technique.

6. Conclusion

We have developed a simple, swift response QA system for QAC-1. The time to obtain the answers is satisfactory, but the overall scores are not high enough mainly due to misretrieval of the adapted IR module.

Although the examination of the experimental results on the algorithms and strategy are still in progress, some of them show the possibilities to improve the results.

Our system still has some bugs to be fixed, and IR module to be improved to begin with. We will consider introducing thesauruses, automatic learning or other techniques after that, with the real-time response in mind.

References

[1] J. Fukumoto, M. Shimohata, F. Masui, M. Sasaki. Oki Electric Industry: Description of the Oki System as Used for MET-2. In Proceedings of 7^{th} Message Understanding Conference (MUC-7), 1998.

[2] Y. Matsumoto, et al. Japanese Morphological Analysis System Chasen Version 2.0 manual 2^{nd} Edition. Technical Report NAIST-IS-TR99009, NAIST, 1999.

[3] Namazu Project. Namazu: a Full-Text Search Engine. http://www.namazu.org/index.html.en.

[4] H. Ohnuma, A. Ikeno. Answer Extraction of Question and Answering System on HTML Documents. In Proceedings of 63^{rd} annual meeting of IPSJ, 3-41, 2001. (in Japanese)