Question Answering System with Fine Grain Answer Types and Search Refinement

David Ramamonjisoa
Faculty of Software and Information Science, Iwate Prefectural University
152-52 Sugo Takizawa, Iwate 020-0193, Japan
david@soft.iwate-pu.ac.jp

Abstract

In the present paper, we describe the improvement of our Question Answering System (QAS). We added keywords relevance factor, search refinement and fine grain type extraction of the expected answer to the system. We attempted to avoid using heavy natural language processing techniques in order to process large amounts of data from the newspaper corpus database. These changes have yielded promising experimental results. These changes and the experimental results are detailed herein.

Keywords: Fine-grain answer types, keyword relevance, search refinement.

1. Introduction

The present paper describes the improvement of our Japanese newspaper corpus QAS. The system accepts questions in Japanese natural language and provides a list of possible answers [1]. The system extracts phrases or queries from the question and then uses an existing search engine to retrieve articles that are relevant to the queries. From these articles, the system extracts passages and answer candidates. We added a relevancy property for each keyword during the question analysis and a fine-grain answer type extraction module for the answer candidate selection. The answer candidates are extracted from relevant documents. Relevant documents are determined based not only on the presence of question keywords, but also on identification of the semantic category of the expected answer type. In addition, the most important improvement to our system is the search refinement. The search strategy guarantees that articles will be provided for each question to the answer extraction.

Related researches are the works in CRL1-2 [4], Toshiba ASKMi [5], and YNU QA [6].

The present paper describes the improvement of and experimental findings for our system. The next section describes the system architecture and details each module algorithm. We then present the results of our experiments. Finally, we discuss our findings and present our conclusions.

2. System Architecture

Our system consists of the following modules: question analysis, documents search, and answer candidate extraction (figure 1). The question analysis is a module that parses the question and extracts keywords and answer candidate type. The search module is an iterative search and accepts boolean queries. The answer candidate extraction is responsible for the selection and ranking of the answer to the question.

2.1 Fine grain Answer types and Keyword relevance

The question analysis module involves the construction of the search query for the search module. A natural-language question is transformed into a phrase and keywords list. A phrase is a set of words within quotes, as in search engine. The keywords are selected terms within the question sentence obtained using a parser and a keywords extractor program. We use the degree of term variation as the primary criteria for keyword selection. High-relevance keywords are terms that vary little across different texts, for example, terms within quotations, proper nouns, and comparative or superlative adjectives. Medium-relevance keywords are terms such as modified nouns, single nouns, and first names of persons. Low-relevance keywords are terms such as verbs, as well as single nouns that are answer type terms. We use a table (see table 1) to compute the relevance of the terms [2].

A sub-module called the answer type extractor determines the type of the expected answer. Pattern matching is used for implementation. Our previous system uses a lexicon, which provides 10 types of words, was found to be insufficient for real-world applications. Our program can extract now the answer types of questions without question phrase "*who*" or "*where*" such as: "Which American writer has been married four times?", "Which country has the highest debt?", or "Which prefecture in Kyushu has the warmest winter?" The answer type of the last question is a *location*, but the fine-grain answer type is a *prefecture*. The general and fine-grain types are extracted.

Table 1: Keyword relevance calculation

katakana words	+2
one morpheme	+0.2
two morphemes	+0.25
three morphemes	+0.5
four morphemes	+1.1
five or more morphemes	+1.2
proper nouns, names	+3
Nouns	+1
Verbs	+0.001
words within quote 「」	+3

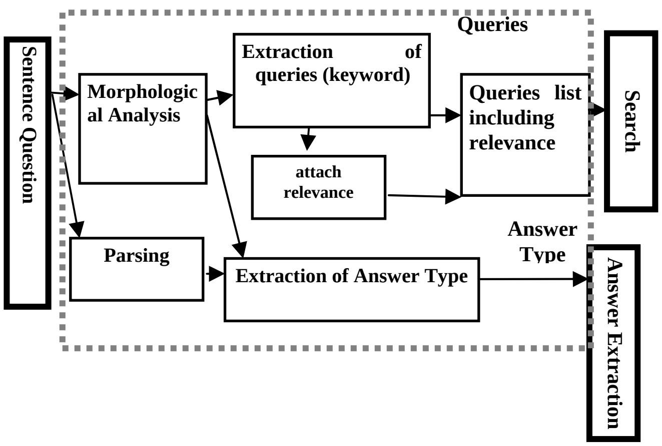

Figure 1: Question Analysis flowchart

2.2 Iterative search and passage selection

The search module looks for relevant documents by using a homemade search program for a corpus. The search is performed iteratively by adding or removing terms in the Boolean search query in order to obtain a sufficient number of documents for the answer extraction. The result of the search is a list of documents that satisfies the search criteria. The flowchart of the document search is depicted in figure 2. The passage retrieval sub-module selects the best passages from the list of documents.

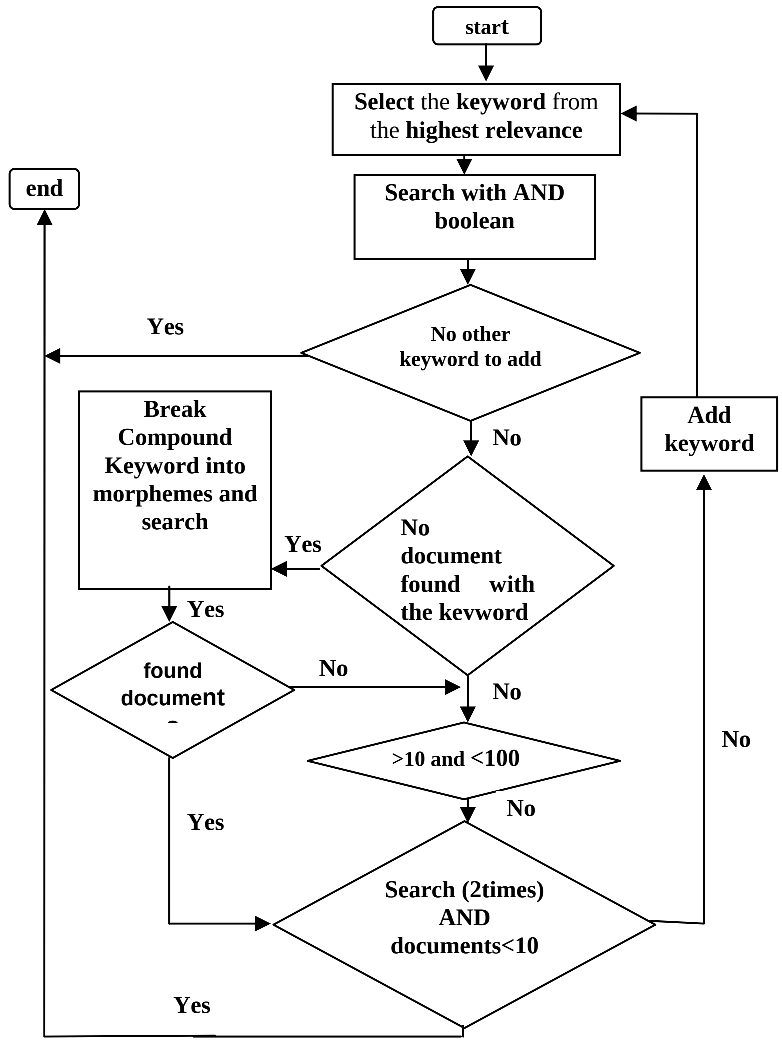

Figure 2: Page search strategy

After necessary documents are collected, this module starts to index and to calculate the rank of each of them (noted r_d). The vector space model is used to retrieve the top 50 paragraphs for each query from the underlying collection. The query vector and each document are compared according to their frequency known as TF*IDF. The similarity of the query and the document reduced to vectors is then the measure of their product known as cosine (see below formula).

$$sim(q, d_i) = \left(\frac{\vec{q} \bullet \vec{d}_i}{|\vec{q}||\vec{d}_i|} \right)$$

A rank of a passage is a part of the score of an answer candidate it contains.

2.3 Answer extraction and scoring

The answer extraction module concerns the extraction of potential answers (also called candidates) from the text files output by the search module. We implemented the extraction area for each passage as 50 Japanese characters or five sentences around the phrase or keywords. The areas that are not in the vicinity of query keywords are unlikely to contain the answer.

Rules are written to identify the answer type (general type) by pattern matching in the passages. The other types (specific types) are detected using the hierarchy answer type database. The answer candidates are sorted in descending order according to a score obtained from the scoring algorithm described below (see also in [1]).

We assume that the answer is the nearest term to each keyword in the selected documents. Based on this assumption, the score for each answer candidate is given by the formula below:

$$S1(a) = \sum_{i=1}^{n} \frac{1}{d_i + 1} \quad (1),$$

$$S1'(a) = S1(a) + \frac{1}{r(a)} \quad (2),$$

$$S2(a) = S1'(a) * (1 + \frac{1}{n_{types}}) \quad (3)$$

where, $\boldsymbol{d_i}$ is the distance (as the number of morphemes) between the nearest keyword denoted with number $\boldsymbol{i}$ in the area and the answer candidate a, $\boldsymbol{n}$ is the number of keywords in the area and **S1(** a **)** is the score of the answer candidate a. **S1' (** a **)** is the score obtained after taking into account the rank information by the retrieval module and **S2(** a **)** is the final rank by adding the answer type matching to the score. $\boldsymbol{n_{types}}$ is the number of semantic category of the candidate a in one passage**.** The total score is the sum of all **S2(** a **)** obtained in the retrieved passages.

$$TotalScore(a_i) = \sum_{j=1}^{num\_passages} S2(a_i) \quad (4)$$

For example, the question "What is the occupation of Mr. C. W. Nichols?" has a candidate answer noun such as "writer", "explorer", or "researcher". Those nouns have a semantic category PERSON. The number of the category is $\boldsymbol{n_{types}}$ =1 and used in the score S2. Therefore, if we have 4 passages and 3 scores (3.5, 1.5, 1.2) of a candidate, then we get a total score 6.2 for this candidate. It is clear that the redundancy is important.

3. Results and Discussions

3.1 Task definitions

The purpose of the QAC was to develop practical QA systems in an open domain focusing on research of user interaction and information extraction. It has also an objective to evaluate the method for the question answering system and information resources.

The tasks we have evaluated and described in this paper concern to provide one to three ordered answers for each question named QAC2 task1 in NTCIR 2004 and QAC3 in NTCIR 2005.

For target documents, four years Japanese newspaper articles spanning a period of four years (1998, 1999, 2000 and 2001) taken from both the Mainichi Newspaper and Yomiuri Newspaper.

Questions used for evaluation require short answers which were exact answers consisting of a noun or noun phrase indicating name of person, an organization, or facts such as money, date, size, …

Every participant can use other information sources such as encyclopedia, thesaurus, corpus of data and so on. However, answer expressions have to exist in newspaper articles and information of document ID is required as support information for each question.

3.2 Environment of the experiment

A laptop with a memory 1GB and CPU 2.2GHz was used to do the experimentation. Perl is used to program the modules in the system and a morphological and syntactical analyzer (Chasen and Cabocha) [7] is added to parse the sentences.

3.3 Evaluation

3.3.1 Result of the NTCIR4-QAC2 post-workshop

For 200 questions, the previous system provides only 38 answers in the top 10, compared to 81 answers for the new system (see figure 3b). The new system has better performance in the top 11 to 20 and top 21 to 50, even when the performance is not considerable. For the remaining questions, which do not have answers in top 50, the new system could retrieve 81 documents related to the 81 questions, in contrast to only 21 questions retrieved by the previous system. The new system retrieved a document for each question, whereas the previous system returned documents for only 77 of the questions (see figure 3a). The bars in figure 3b show that the previous system could answer only 64 questions, whereas the new system could answer a maximum 173 questions.

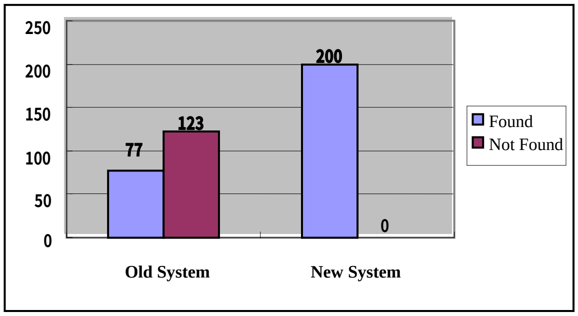

Figure 3a: Comparison of the documents obtained by the previous and new systems for a test involving 200 questions

In figure 3b, correct answers, represented by the light grey bars, are among the obtained documents, but they are not extracted. There are 81 questions in this case, which is important to consider in the future.

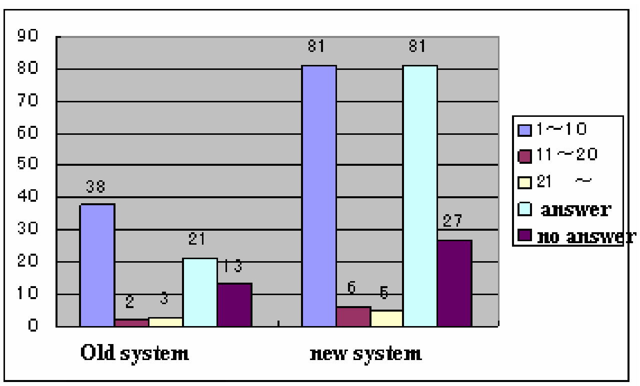

Figure 3b: answers obtained by the previous and new systems during a test involving 200 questions from NTCIR-4 QAC2 [3]

Some questions that were answered using the previous system could not be answered using the new system (total of five questions). The reason for this situation is that the number of documents returned by the new system is limited to 100. Therefore, the relevant document may not be selected, and so the answer cannot be found.

Although the effect of keyword relevance on the answer is not important, it was useful in the search. The graph in figure 4 shows the system results with and without keyword relevance. In the results without keyword relevance, only five questions returned no documents and no answers.

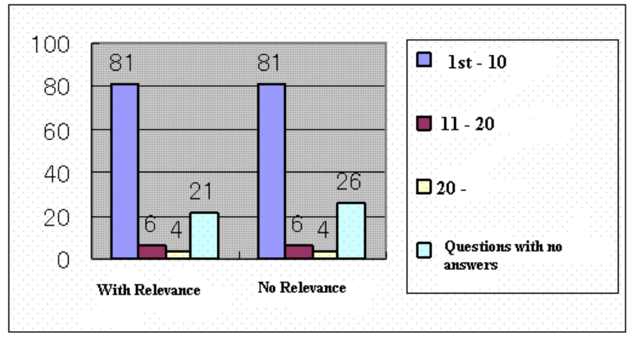

Figure 4: Results with and without keyword relevance

The use of fine-grain answer type in the answer extraction module is important, as the answer is restricted to belong to a single category. A clustering of the answers in the list remains to be performed in order to remove redundancy in the answer.

The Japanese language has a problem in terms of the accuracy of question sentences. There is no distinction between plural or singular subjects. “Who is” and “Who are” are written in the same way. The same is true for “What” questions. For example, the question: “What countries were formed after British India became independent?” does not specify a plural or singular subject. We are investigating a solution to this problem whereby, rather than stopping the search after one answer is found, all possible answers are provided.

For conclusion, our experiments revealed that the likelihood that the correct answer will be found in the top five candidates on the list has been increased to 70% and that the likelihood of finding an answer candidate of the wrong type is approximately zero.

3.3.2 Result of the contest NTCIR5-QAC3

The result of the official participation to the contest can be summarized as follows. The system performed poorly with the series of related questions. The reference run gave a better result but it is still far from the best system. We are working on those problems including the answer extraction module that we need to improve.

4. Conclusions

In this paper, in order to improve the quality of passages and answers in our QAS, we introduce a new question analysis module and a new passage search technique. According to the experiments in NTCIR4 –QAC2 subtask 1 and NTCIR5-QAC3, we obtained better performance than our previous QAS. However, the overall results by the MRR criteria are still low. Further improvements include a new answer extraction module based on new algorithm analyzing sentences structure rather than based on keywords and answer candidates distance.

5. References

[1] Ramamonjisoa, D., 2004, Finding Relevant Answers in Question Answering Contest, *NTCIR workshop 4 Meeting*, Tokyo, Japan.

[2] Tonoike M. et al., 2004, Answer Validation by Keyword Association. *Japanese Natural Language Conference*. Japan.

[3] Fukumoto J. et al., 2004, Question Answering Challenge for Five ranked answers and List answers – Overview of NTCIR4 QAC2 subtask1 and 2. *NTCIR workshop 4 Meeting*, June 2-4, Japan.

[4] Murata M. Et al., 2004, Japanese Question Answering System Using Decreased Adding with Multiple Answers, *NTCIR workshop 4 Meeting* pp.353-360, June 2-4, Japan.

[5] Sakai T. et al., 2004, Toshiba ASKMi at NTCIR-4 QAC2, *NTCIR workshop 4 Meeting* pp.387-394, June 2-4, Japan.

[6] Mori. T, Japanese Q/A System using A* Search and Its Improvement – Yokohama National University at QAC2, *NTCIR workshop 4 Meeting* pp.345-352, June 2-4, Japan.

[7] Taku Kudo and Yuji Matsumoto, Fast Methods for Kernel-Based Text Analysis, ACL 2003 in Sapporo, Japan.