Finding Relevant Answers in Question Answering System Contest

David R. 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

We have investigated the potential use of question answering systems and participated in the Question Answering Challenge (QAC) of National institute of informatics Test Collection for Information Retrieval systems (NTCIR).

In this paper, we describe our question answering system, the preliminary results of our experiments to the contest and some possible improvement to question answering systems.

Keywords: *Question Answering System, Document Retrieval, Answer scoring, Experimentation.*

1. Introduction

Open-domain question answering (QA) constitutes a modern and exciting information retrieval topic. The question is formed in natural language thus eliminating any artificial constraints sometimes imposed by a particular input syntax [1].

The system, and not the user, is responsible for analyzing the content of full-length documents and identifying short, relevant text fragments. Large-scale open-domain QA such as Question Answering Challenge (QAC) of National institute of informatics Test Collection for Information Retrieval systems (NTCIR) [2] is a challenging field, due to the high complexity of the sub-problem specific to QA. Requirements include an adequate understanding and representation of the question semantics, along with precise extraction of relevant answers, from vast amounts of unrestricted text.

We have investigated the potential use of question answering systems [12] and participated in the Question Answering Challenge (QAC) of National institute of informatics Test Collection for Information Retrieval systems (NTCIR).

In our previous work, we studied a web-based question answering system similar to AnswerBus [3], START [4] or Mulder [5]. It is trying to answer question such as ``*How many meter is the summit of Mount Fuji?*", ``*Which Highschool is the baseball player Hideki Matsui from?*'' or ``*Who is the prime minister of Japan originated from Iwate?*'' , but not the question such as ``*How to cook a curry rice?*". Indeed, the last question is difficult to answer by the existing question answering systems. Our QA system relies on existing search engine such as Google and it gives answers according to its ranking algorithms. We have obtained an interesting result that it is difficult to evaluate because questions are not guaranteed to have at least one correct answer as in TREC [6] or NTCIR-QAC [7]. The NTCIR-QAC seemed appropriate to make an evaluation of our system and should give us a feedback on the future development.

This paper is organized as follows. We summarize the background of question answering systems. We depict the task of the NTCIR-QAC. The next section describes the system architecture. Then, we explain the answer extraction algorithm and scoring module. Finally, we conclude with the result of the experiments and perspectives on the improvement of our system.

2. Background

2.1 General Question Answering Systems

Question Answering is a computer-based activity that combines searching large amounts of documents and understanding both questions and textual passages to the degree necessary to select a text fragment as answer to a question. This activity is bringing together Information Retrieval and Natural Language Processing fields.

If multiple candidate answers are identified in documents but only few are actually correct, how can a system rank them such that answers are returned based on relevance?

The solution consists on capturing the semantics of the questions submitted by users and making intelligent

decisions when accessing and searching the text collection.

The main characteristics structure of question answering system are the decomposition of the problem into sub-problems such as question analysis problem, passage retrieval problem, and answer extraction problem. A question answering system contains modules to deal with these sub-problems.

Our system is based on this general structure of the question answering system.

2.2 Assumptions

In terms of media type and input format, we consider the task of answering written questions from text-only documents.

Answers are strings that are identified and extracted from documents, rather that strings that are generated separately within direct relation to the document content.

Why and *How* questions are not answered reliably with our system. Those questions are still difficult.

3. The NTCIR-QAC

3.1 Task definition

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.

During the contest, we officially subscribed to the task which concerns to answer all correct answers to each question. If all the answers are correct, full score will be given. If there are several answers, system has to return all the answer. If there are some wrong answers, this will be a penalty of the score. Average F-Measure (AFM) is used for evaluation of this task.

The task we have evaluated and described in this paper concerns to provide one to five ordered answers for each question.

For target documents, four years Japanese newspaper articles spanning a period of two years (1998 and 1999) 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 Evaluation

The system extracted five answers from the documents in some order. The inverse number of the order, Reciprocal Rank (RR), was the score of the question. The Mean Reciprocal Rank (MRR) was used for the evaluation.

4. The system architecture

The figure 1 shows the system architecture. The question analysis, the search and collection of articles, and the answers extraction are the main modules.

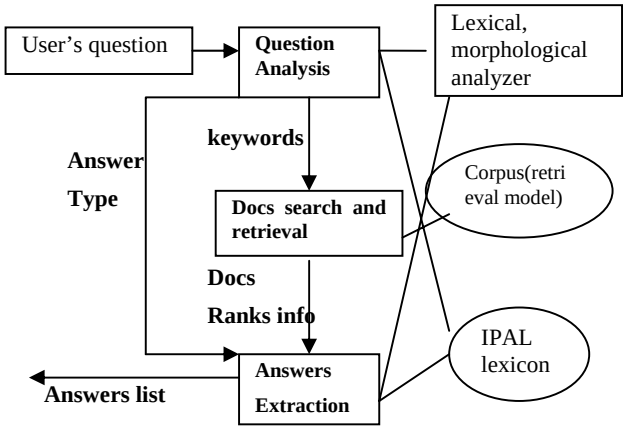


Figure 1. System architecture.

Answer type function, morphological analyzer, and search function are other modules used in the system.

This section describes each internal structure and function of each main module.

4.1 Question analysis

This module consists with the construction of the search query for the search module. The question in natural language is transformed into phrase and keywords list. The phrase is the positive or negative form of the sentence obtained from the interrogative form. For example, a question such as ``*Who is the prime minister of Japan?*'' is then transformed into ``*The prime minister of Japan is*''. The question also is converted into keywords with the help of the morphological analyzer (Chasen/ChaboCha) [8]. From the previous example, the system gives two kind of keywords list such as the first

one is like *prime minister* and *Japan*, and the second one is obtained by splitting into a set of single morphemes the sentence.

A sub-module called answer types matching has a task to determine the type of the answer according to a lexicon (IPAL) [9]. The IPAL lexicon provides 10 types of word (see table 1). We use these types of answer such as human (names), quantity (number) and time (date, hour, minute and seconds) by using the pattern-based identification shown in table 2.

Table 1. Semantic category of words

	Answer Type	**Semantic Category**
Name	Human	HUM
	Organization	ORG
	Animal	ANI
	Plant	PLA
	Natural	NAT
	Product	PRO
	Location	LOC
	Language product	LIN
Quantity	QUA	QUA
Time	TIM	TIM

Table 2. Pattern-based identification of answer type

Wh-question in Japanese (equivalence in English)	**Answer type(semantic category)**
who	HUM
where	ORG、LOC
when	TIM
how much, how many, how old	QUA
what+ {alphabet:m,cm,…}	QUA
What＋{year,month,day.hour,min,sec}	TIM
What＋noun (e.g. position) is	QUA
What place is	ORG、LOC
What (not matched above)	PRO、LIN、NAT、ANI、PLA
Not matched above	none

4.2 Corpus data and search mechanism

The corpus data is formatted in SGML where each document is delimited by tags. Each document has a number identification, headline, category, and text body for the news article.

Table 3. Used tags in the corpus data

<DOC>	</DOC>
<DOCNO>	</DOCNO>
<TEXT>	</TEXT>
<SECTION>	</SECTION>
<WORDS>	</WORDS>
<HEADLINE>	</HEADLINE>

There are around four hundred sixty thousand documents in the corpus data. Therefore, the search processing was slowed by this large size corpus.

This module allows to access the set of documents based on their semantic content rapidly.

We use a vector model to ameliorate our search mechanism. We assume that the search keywords obtained from the question sentence is a vector **q** and each document also is a vector $\mathbf{d_j}$. The similarity of the document and the query is the inner product of the two vectors as $\mathbf{cos(d_j,q)}$. We sort this result to get the most relevant document.

4.3 Answers extraction

This module concerns with the extraction of the potential answers (called also *candidates*) from the text files output of the search module.

Firstly, the result of the search query as phrase is processed. The process is a pattern matching of the regular expression. The extraction also depends on the type of the answer.

Secondly, if there is no result from the search module according to the phrase query, the keyword search query result is processed.

We used similar process as described in the system of Kwok [5] for the answers extraction. We implemented the extraction area (called *summary* in Kwok's system and *passage* in general QA) for each document as 50 Japanese characters or five sentences around the phrase or keywords. The areas that are not close to any query keywords are unlikely to contain the answer.

The area (50 morphemes) that keywords gather is selected to the candidate area that is looked for and an actual answer is extracted in the document containing the area.

The measure of the distance of keywords each other is based on the following formula. The score of the area is then given as:

$$SCH(i) = \frac{1}{dk(1)} + \frac{1}{dk(2)} + ... + \frac{1}{dk(n-1)}$$

where, dk(j) is the distance (as the number of morphemes) between the nearest keyword in the area, n is the number of keywords in the area, j is the number of morphemes in the document, and SCH(i) is the score of the extraction area.

The areas with the highest score are stored in a list of answer extraction areas.

The answer extraction area is analyzed with the syntactical analyzer (chasen [11]), and an answer candidate is extracted. This candidate should be the combination of nouns, unknown words, numbers or alphabets. The syntactical tags given by this external module allow the realization of this task.

5. Scoring and ranking the answers

5.1 Answer candidates extraction's flowchart

Figure 2 shows the flowchart of the answer extraction module by extracting and scoring an answer candidate. In this flow, ALL represents the number of documents returned by the retrieval module. w is a variable which represents one document.

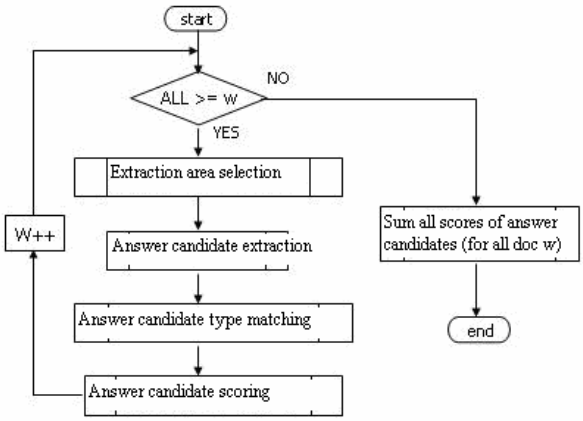


Figure 2. Answer candidates extraction's flowchart .

The extraction area selection and answer candidate extraction procedures are already explained in the section 4.3.

The other procedures are explained in the followings.

5.2 The score of the extracted answer according to the relevance

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

$$S1(a) = \frac{1}{d_1 + 1} + \frac{1}{d_2 + 1} + ... + \frac{1}{d_n + 1}$$

where, d_j is the distance (as the number of morphemes) between the nearest keyword in the area, n is the number of keywords in the area, i is the number of morphemes in the document, and $S1(a)$ is the score of the answer candidate a.

$$S1'(a) = S1(a) + \frac{1}{r_a}$$

$S1'(a)$ is the score obtained after taking into account the rank information by the retrieval module.

5.3 Answer type and answer candidate extraction

Rules are written to identify the type of the answer by pattern matching. The candidates with numeric characters are classified into *quantity* or *time* types and the others are names as *human* type if the question is started with ``who''. The other types are detected with the tables 1,2.

If the expected answer type is a number, then the answer to be extracted is a string with alphanumerical characters.

If the expected answer type is a time, then the answer to be extracted is a string with alphanumerical characters containing special characters such as year, month, day, hours, etc.

If the expected answer type is a name, then the answer should be a noun or an unknown word as a part of speech tag. In this case the answer is a noun phrase or a series of strings.

In this last case, the system must determine which type is answer. We choose to use the method which the answer type of the answer candidate is decided as is taken by using the choice limitation of verbal rank frame. In the verbal rank frame, each verb and usually term used with it is classified with the hierarchical semantic categories in the figure 3[10]. Some verb has preposition where the following term must be a name or an location. For example, the subject of the verb *to eat* must be used with a human or animal.

To define this type of the answer candidate we propose the following algorithm:

- Look for all the sentences which an answer candidate is included to within the answer candidate extraction documents.
- Analyze syntactically each found sentence
- Verify the sentence containing a particle and its associated verb. If this pattern is redundant then deduce the answer type according to the table in the figure 3 by matching it. It is possible that an answer candidate has multiple answer types or not at all.

Next, when the answer type of the answer candidate corresponds with the answer type of the question sentence, the score S1' is changed as the formula below:

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

MAX and *ntypes* represent a maximum number of types and the number of the answer types own the candidate answer.

Figure 3. Hierarchical semantic categories of IPAL.

5.4 The total score of each answer candidate

The score S2 of the answer candidate of the same name extracted from the document which is different if an answer candidate is extracted from all the documents and the scoring is finished in summing up all scores.

6. Experimentations

The execution of the program is the snapshot of the window presented in the figure 4.

Figure 4.Execution snapshot.

In this snapshot, the question is "what is the capital of Japan?" and the top 1 result is Tokyo. In this snapshot, the data corpus is the Web but not the newspaper.

6.1 Environment of the experiment

A laptop with a memory 512MB and CPU 1.2GHz is used to do the experimentation. Perl is used to program the modules in the system and a syntactical analyzer Chasen [11] is added to make the parsing.

6.2 Results of the NTCIR-QAC

6.2.1 Comparison of two tests

In our experiments, for 200 questions, the system found 59 answers and the correct answers found in the top 3 are 27 for a method with ranking (see figure 5). 2 questions have responded correctly in comparing to the result without ranking information. The top 4 to 20 did not change in both system. The number of incorrect answers from the system is same with or without ranking information among the 59 questions with answers.

We can conclude that the ranking information brought some improvement to the system. However, it does not really extend the capability of the system to find more answers in the contest questions.

We had better performance with the system which includes the WEB rather than using the corpus. Indeed, our system is exploiting the redundancy to boost the score and when the answer is given in small information source, the evaluation is becoming difficult.

6.2.2 Result of the contest

The result of the official participation to the contest is not really good. We obtained an Average F-Measure of 0.15 which is the last in the rank for all participants. We should participate to the task 1 which is easier. In the following, we describe the result of task 1 that we conducted individually.

The MRR obtained from our system is equal to 0.354. The best system in the contest has an MRR equal to 0.600 and the worst system has an MRR equal to 0.03. The average MRR of all participants is around 0.300. We think that our system is above the average system. However it is far from the best one.

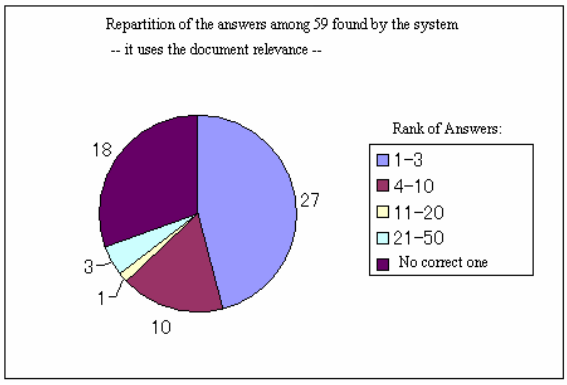
6.2.3 Remarks on the current system

Our system has many weaknesses in the question analysis and passage extraction. We didn't use fully the power of the natural language processing. A thesaurus is necessary to expand or remove a keyword. When the retrieval module did not find document according to the input keywords, the system is stuck. It can't pursue the extraction and the system return "no answer found". There are many cases in this situation during the test as 141 questions have no answers.

A name entity recognizer also is sometimes useful to capture the possible answer type and used with the IPAL hierarchical semantic categories.

The question analysis should include inter-terms relations that can be used to ameliorate the precisions of the answer.

When the answer is name of person or an organization, our system can not make difference between two strings with the same meaning but in different syntax. For example, "G.W. Bush" is not the same as "President Bush". An addition of a substring matching may solve this problem.



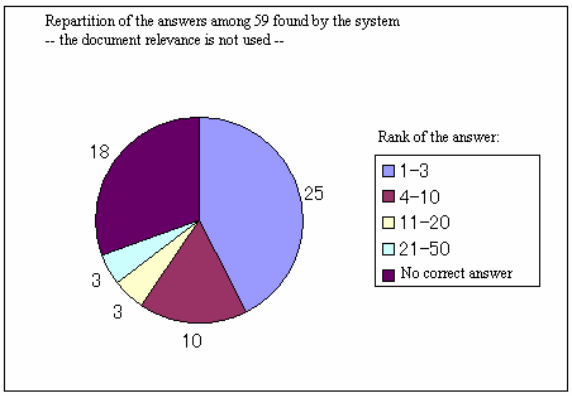


Figure 5.Comparison of the result with or without document relevance (rank information).

6.3 Post analysis of the results

6.3.1 Effect of the keyword selection module

We modified our question analysis module in order to find more documents according to the query. We added a sub-module called keyword selection for this task. The keyword selection concerns to remove or add terms in the query in order to get sufficient documents for the answer extraction. Terms in the query are obtained at first from the question phrase. Terms are classified to high, medium and low relevance (see [1]). High relevance terms are words such as proper nouns, words within Japanese quote " 「」 ", and comparative or superlative adjectives. Medium relevance terms are verbs and nouns. If we have terms (k1, k2, k3, k4, k5) in the question phrase, then the search should be like $search(k1 \wedge k2 \wedge k3 \wedge k4 \wedge k5)$

By eliminating low and/or medium relevance terms, there are more chances to find documents related the query. Our experiment demonstrates that the system found 120 answers out of 200 questions with this technique (removing terms in the query increases two fold the found answers). We noticed also that Japanese terms should sometimes split as several terms in order to be found in the documents. For example, the term "Japanese baseball player" is considered as one term in Japanese ("日本人野球選手") and should be considered as three terms "日本人" "野球", " 選手". Making several searches is necessary to detect those exceptions. When low and medium relevance terms are removed and after the search, there is no document found then this second technique must be applied.

6.3.2 More heuristics on question patterns

The most difficult question is a question with *"what"* and *something*. Rules are necessary to deal with different question patterns for this "what" question. In the task2, we can cite question number 3, 7, 8, 13, 17, 18, 19, 21, 23, 25, 34, 37, 40, 48, 50, 52, 56, 65, 66, 70, 79, 80, and so on.

The answer type of this question is an artifact, natural things, or others. This makes the processing difficult. Some participants use more than 300 rules to deal with problem (see [13],[14]).

6.3.3 Necessity for better answer extraction and scoring

Our current scoring is not enough. We sum at the moment three techniques scores to rank our answer candidates. Other systems use complicated algorithms (machine learning, search algorithm, etc) to process this scoring. Name entity based identification should be considered in our system. It is used in almost QAC systems. We can take into account too the semantic matching relevance features from the question itself (see [1]) and add that score to make the overall score.

7. Conclusions and perspectives

7.1 Conclusions

A question answering system is described in this paper. It uses the architecture of the Question Answering in general. We tried to modify the retrieval module to test impact of the document relevance in the results. Indeed, it changes a little the system.

7.2 Perspectives

Several areas of future work have appeared while analyzing results. First, question analysis has to be improved by allowing the addition or removing of keywords according the system needs. Second, the retrieval module used for ranking relevant documents has to be adapted for passage retrieval useful for the answer extraction module. The use of passage retrieval engine should improve considerably the performance of the system. Third, the answer type matching is not reliable yet and has to be changed. This can ameliorate the precision of each answer.

8. Acknowledgments

Our thanks to the NTCIR group, Mr. T. Chiba and Mr. N. Murooka for their participation to the realization of the system.

9. References

[1] Pasca, M. *Open-Domain Question Answering from Large Text Collections*. CSLI Publications, 2003.

[2] Fukumoto, J., et al. Question Answering Challenge (QAC) – Question Evaluation at NTCIR workshop 3. In *Proceedings of the AAAI 2003 Spring Symposium New Direction in Question Answering*, pp. 122-133, 2003.

[3] Zheng, Z. AnswerBus Question Answering System. In *Proceedings of HLT 2002*, San Diego, March 24-27, 2002

[4] Katz, B. From Sentence Processing to Information Access on the WWW. In *Proceedings of the AAAI 1997*, Stanford, 1997.

[5] Kwok et al. Scaling Question Answering to the Web. In *Proceedings of the WWW01*, Hong Kong, May 1-5, 2001.

[6] Voorhees, E.M. The TREC-8 Question Answering Track Report. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, p.77-82, Maryland, 1999.

[7] NTCIR homepage:

http://research.nii.ac.jp/ntcir/index-e.html

[8] Kudo, T. and Matsumoto, Y. , *Japanese Dependency Analysis using Cascaded Chunking*, Proceedings of the 6th Conference on Natural Language Learning, pp. 63-69, 2002.

http://cactus.aist-nara.ac.jp/~taku-ku/software/cabocha/

[9] IPAL homepage:
http://www.ipa.go.jp/STC/NIHONGO/IPAL/ipal.html

[10] Nagao, M. *Natural Language Processing (in Japanese).* Iwanami Publications, 1996.

[11] Chasen homepage

http:// chasen.aist-nara.ac.jp

[12] Ramamonjisoa, D. and Chiba, T. Solving Japanese Quiz with a Web-based Question Answering System. In *Proceedings of the WWW03*, Budapest, May 20-24, 2003.

[13] Isozaki, H. NTT's Question Answering System for NTCIR QAC2. In working notes of the NTCIR-QAC2, pp. 326-332, 2004.

[14] Kosugi, T. et al. Qustion Answering System with a vector similarity scoring method. In working notes of the NTCIR-QAC2, pp. 333-337, 2004.