QUARK: A question and answering system using newspaper corpus as a knowledge source

Keizo KAWATA   Hiroyuki SAKAI   Shigeru MASUYAMA
Department of Knowledge-based Information Engineering, Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan
kawata@smlab.tutkie.tut.ac.jp

Abstract

We developed a question and answering system QUARK. This system extracts an answer from newspaper corpus as a knowledge source by a statistical technique. We participated in NTCIR3 QAC task to evaluate our system.

Keywords: *NTCIR, Question and Answering,*

1 Introduction

A question and answering system, in this paper, means a system for returning an appropriate answer by a word or a series of words and phrases employing extensive and unorganized information resources like a corpus of newspaper articles to answer the question expressed in natural language inputted by a user [1] [3].

A question and answering system may be regarded as a smart information retrieval system. An input to an IR system consists of reference keywords or a logical expression constructed by them.

On the other hand, a question and answering system accepts an input as a quesiton sentence written in natural language (For example, “Who is the president of the United States ?”). A question by natural language provides a more user-friendly interface. Moreover, in information retrieval, a user has to look for the target information from the output document list. In contract, an answer to a question itself is the system output in a question and answering system.

The advantage of a question and answering system is that a user’s burden is alleviated compared with a conventional information retrieval system. But the accuracy of the present question and answering system is not necessarily high. If accuracy is improved to be satisfactory, it becomes a useful assistance tool for a user to obtain the appropriate information from a vast quantity of information resources.

In this paper, we develop and evaluate a question and answering system QUARK [1] which extracts an answer from knowledge sources by a statistical method.

2 Formula for getting an answer

A word which appears in a question sentence is related to its answer. These words are defined as keywords. For example, “Who is the president of the United States?”, words “the United States”, “President” are keywords for looking for an answer to the question.

In the case of this example, the correct answer is ”Clinton,” because we used the ’98 and ’99 editions during two years of newspaper aricles of Mainichi Shimbun as a corpus.

A possibility that keywords and answers are included in the same sentence of the same article is considerably high.

The sentence containing at least one keyword in the knowledge source is defined to be a **content sentence**. The content sentence may contain the answer to the question. When all the content sentences are obtained from the knowledge source, we consider that words chosen from content sentences are candidates for answers.

These answer candidates are given weight by the following formula which is obtained by modifying tf·idf method [2].

$$P(w_i) = (\frac{ds(w_i)}{S}) \log(\frac{N}{df(w_i)}), \qquad (1)$$

where
w_i: phrase of an answer candidate; $i = 1, ..., n$
$P(w_i)$: weight of answer candidate w_i,
N: frequency of all documents in corpus,
$df(w_i)$: frequency of documents containing the answer candidate w_i,
S: the number of all content sentences,

[1] QUestion and AnsweRing system using newspaper corpus as a Knowledge source.

$ds(w_i)$: the number of content sentences containing the answer candidate w_i,

This formula has a large value, when w_i has important contents and is contained only in content sentences.

This method estimates relevance between an answer and the keyword on the basis of difference in term frequency between the entire corpus and the part in corpus which consists of content sentences.

As the weight of answer candidates become larger, the possibility that the answer candidates is a correct answer increases. The less the number of answer candidates becomes, the less weight values disperse.

If the system can estimate the appropriate answer type from the question sentence, it is necessary to improve accuracy by collecting only terms which have an answer type same as that of the question sentence.

3 System configuration

QUARK is composed of the following four modules.

- Article extraction module
- Answer type determination module
- Answer candidate extraction module
- Answer candidate weight calculation module

3.1 Article extraction module

3.1.1 Purpose

This module analyses a question sentence as an input, and extracts, from corpus, articles which may contain an answer to the question.

3.1.2 Method of the article extraction module

First, only noun n extracted from the question sentence by using morphological analyser JUMAN 3.61[2] and syntactic analyser KNP 2.0 [3] are used as keywords. Complex words are regarded as one noun. On the other hand, numerals are excluded from keywords.

Secondly, articles which include all keywords from corpus using IR system Namazu [4] are obtained. Retrieval uses a query expressed by the following formula (2) using keywords $A_1, ..., A_n$.

$$(A_1 \wedge A_2 \wedge A_3 \wedge ... \wedge A_n) \qquad (2)$$

As a result, this module outputs an articles list. The number of articles is restricted to the maximum of 30. If no article is retrieved, one word is deleted from keywords using heuristics below and retrieval is retried.

[Heuristics for reducing keywords]

1. Check noun type of each keyword.
2. Delete one word from the set **K** of keywords. The word to be deleted is decided in the following way.
   - If a common noun exists in **K**, delete it from **K**.
   - If a common noun does not exist in **K**, delete a verbal noun from **K**.
   - If a common noun and a verbal noun does not exist in **K**, delete a proper noun from **K**.

   When there are two or more types of nouns, delete a word which is near the head of a question sentence.

   (Note) This deletion order is based on the following assumption. Relevance of proper nouns to an answer to the question is higher and they are not deleted.

   The difference of the importance of a common noun and a verval noun is not so clear, but, we assume that a verbal noun which describes an operation is more important here. Therefore, deletion order is determined as above.
3. After reduction of keywords, if $|\mathbf{K}| \neq 0$, go to the next step.
4. Retry retrieval using remaining keywords.

If we get more than one article, this process is repeated until the number of keywords becomes zero by this reduction. In this case, answer candidates included in articles do not exist.

3.2 Answer type determination module

3.2.1 Purpose

The Purpose of this module is to determine the answer type from the question sentence.

The answer type is selected from those in table 1. Extended IREX expression in table 1 is defined from IREX expression [5] by adding the following two answer types: Type 'COUNT' for numerical expression, and type 'MISC(miscellaneous)' for named entity.

[2] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/juman.html
[3] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/knp.html
[4] http://www.namazu.org/
[5] http://cs.nyu.edu/cs/projects/proteus/irex/

Table 1. Answer type (extended IREX expression)

Answer type	meaning
PERSON	person's name
LOCATION	place's name
ORGANIZATION	organization's name
ARTIFACT	various name of things
DATE	date (year, month, day)
TIME	time (o'clock, minute, second)
MONEY	amount of money
PERCENT	proportion, rate
COUNT	countable noun (extended expression)
MISC	other(extended expression)

3.2.2 Method of the answer type determination module

Table 2 shows correspondence between interrogative pronoun and answer type used for determination of the answer type for the given question.

This table is maked by collecting interrogative pronouns which may be contained in a question sentence and a corresponding answer type is assigned manually.

All the answer types expected was assigned about an interrogative pronoun which may have multiple answer types (For example, what, how much, how many)

An interrogative pronoun which does not exist in table 2 is assigned answer type 'MISC(miscellaneous)'.

Detecting an interrogative pronoun in a question sentence by pattern matching with a question sentence and interrogative pronouns in table 2 is performed.

Table 2. Correspondence between interrogative pronouns and answer types

interrogative	answer type
who	PERSON
when	DATE, TIME
where	LOCATION, ORGANIZATION
how many	COUNT, MONEY, PERCENT
much,how much	COUNT, MONEY, PERCENT
what	PERSON, ARTIFACT, LOCATION, ORGANIZATION
-other-	MISC

3.3 Answer candidate extraction module

3.3.1 Purpose

In this module, we extract terms from articles. And extracted terms become answer candidates.

3.3.2 Method of the answer candidate extraction module

First, content sentences are extracted from articles.

Next, only nouns are extracted from content sentences by using morphological analyser JUMAN 3.61.

Words satisfying the following conditions are deleted from words extracted from content sentences.

- The word that is the same as a keyword.
- Type of words not coinciding with any answer type got by the answer type determination module. But, if the answer type is MISC, then the word is not deleted.

This process is repeated for all of content sentences. And terms for answer candidates are extracted.

3.4 Answer candidate weight calculation module

3.4.1 Purpose

In this module, each answer candidate is given weight, and sorted with respect to the assigned weight.

Finally, answer candidates whose weights are higher than a predetermined threshold are chosen as outputs.

3.4.2 Method of the answer candidate weight calculation module

Each of answer candidates is given weight using fomula (1) obtained by the answer candidate extraction module.

Answer candidates with large weight determined by fomula (1) are chosen as answers.

4 Evaluation

In the evaluation, we used the '98 and '99 editions during two years of Mainichi Shimbun corpus as a knowledge source. Input data used in the question sentence set of 200 questions provided for NTCIR3 QAC task formal run. The correct answer exists with 195 questions among 200 questions.

Our system is evaluated by participating in NTCIR3 QAC task 1. System outputs at least five answers per one question. Reciprocal of rank in which a correct answer appears the first one among answers is adopted as the score for an answer to the question. If the correct answer appears first in rank 1, score is 1. If the correct answer appears first in rank 2, score is $\frac{1}{2}$. There is no correct answer having rank less than 5, or the question does not have a correct answer, then score is zero.

We also participated in tasks 2 and 3 by the same system developed for task 1.

In task 2, a score is determined by both the correct answer and the wrong answer contained in the answer list. If the wrong answer is contained in the answer list, the score is decreased. Therefore, not only choosing a correct answer, but also eliminating a wrong answer from answer candidates list result in higher rank. In QUARK, wrong answers are not tried to be eliminated expecting a correct answer comes to a higher rank by assigning as many answer candidates as possible and gaining weight appropreately. In task 3 a question is classified into a main question and a branch question relevant to a main question. A branch question is constructed using information of the main quesiton, and extracting an answer for a branch question needs various infomation on a main question.

Since QUARK was customized to task 1, analysis of information is not carried over from a main question to its branch questions. The correct answer on a main question is one. No correct answer was obtained for branch questions by QUARK.

The result of an evaluation experiment shows that the total number of the answers outputted by QUARK is 586 (one question has the maximum of five answers). Among all 200 questions, 32 questions had a correct answer by QUARK. Rate of a correct answer is 16%. Average score is 0.099.

The details of an output are shown in table 3 .

Table 3. Answers in detail

Correct/Error	count
Correct at Rank 1	12
Correct at Rank 2	9
Correct at Rank 3	5
Correct at Rank 4	0
Correct at Rank 5	1
Correct:Answering no answer question	3
Correct:Not answering no answer question	2
Error:Answer candidate is zero	105
Error:Answer mistake	63
total	200

Recall and precision are shown in table 4 .

When the question has no correct answer, any answer, even no answer, is treated as a correct answer on task1 at the evaluation. Causes of errors and the reason why some questions have no correct answer may be classfied into the following three cases.

Table 4. Recall and Precision

Recall	Precision
10.492	5.4618

Case 1. The case when extracting answers candidate is successful, but extracted answer candidates do not include a correct answer, or a correct answer exists among answer candidates, but these answer candidates are not assigned high weight.

Case 2. The case when getting articles including answer candidates is failed. In this case, a system can not extract any answer candidate.

Case 3. The case when the question does not have an answer, but some answer candidate is chosen.

5 Discussions

Given 200 questions, the rate of correct answers by QUARK is 13.5%.

To improve this rate, we consider countermeasures for three cases of errors disucussed in the previous section.

For case 1, by more strictly selecting a term in the answer candidate extraction module, more answer candidates seem to become close to a correct answer. In the present condition, choice of a word is done only by using an analysis result with morphological analyser JUMAN 3.61.

Answer candidate with low relevance with a correct answer should be deleted by getting details of word which extracted using some method of extracting named entity.

For case 2, the technique of reducing a keyword and to repeat retrieval until it can gain an article answer candidates may contributed to improve the rate of countermeasure. However, heuristics of reducting keywords has room for improvement.

We need more strict heuristics which has a statistical basis. We consider another techniques for getting articles.

First, a method by which the most important word is chosen from keywords may be promissing. Retrieval by a query expressed by the following expression (3) constructed by the main word and each of support words, may improve its outputs, where the deleted word Q is a main word in a question sentence, while $A_1, ..., A_n$ are support words.

$$Q \wedge (A_1 \vee A_2 \vee A_3 \vee ... \vee A_n) \qquad (3)$$

In case 3, a system must not output any answer. One of considerable countermeasure is as follows; if the maximum weight among those of answer candidates is very low or distribution of weights is flat, a system outputs “no answer.”

Next, we discuss about questions to which QUARK returned a correct answer. In table 3, in the question to which a correct answer can be outputted by QUARK, the appearance ranking of a correct answer tends to have a higher rank. We think that these results by the answer type determination module work effectively, and the word which is not related to a correct answer has been eliminated in the answer candidate extraction module.

But, we think formula (1) is effective to obtain correct answer for the question which asks named entity like a person’s name and a place’s name. In the question which asks named entity, it is easy to use extracted words from articles as an answer candidates with no change. Therefore, weight alligned by formula (1) for using term frequency seems appropriate.

On the other hand, to the question which asks number like time, date, and the number of something countable, QUARK cannot generate correct answers. That is, formula (1) is not so effective to answer the question which asks a numerical value.

A numerical expression tends to have some relative expressions(For example, “next year”,“two days ago.”) Moreover, we cannot tell which parameter is used only from the numerical expression. Words which were semantically the same but differ on the notation makes weights of answer candidates by formula (1) dispersed. Therefore, the correct answer has become hard to be obtained.

To the question which asks named entity, it is expected that accuracy increases by improving the determination method of answer types.

The following two countermeasures can be considered.

- Improvement of the determination method of answer types.

  Adds heuristics for determination, to increase the question sentence which can be distinguished.

- Improvement of the acquirement method of information of a word on the answer candidate extraction module.

  Investigation of feature of a word so that a word is assigned to answer types other than ‘COUNT’ or ‘MISC(miscellaneous)’ as much as possible.

On the question which asks a numerical value, although it differs on the notation in numerical expression, there are a number of synonymous expressions (For example, “July 13 2002” and “(In article on July 6 2002) one week later”,“1000 kilo gram”and “1 ton.”)

Accordingly, simple term frequency seems useful for obtaining a correct answer for this case. The technique which absorbs the difference in the notation and which enables to treat expression with the same meaning is needed.

By technique of getting articles, technique of eliminating words not related to a correct answer for instance, the number of answer candidates is reduced and accuracy of a system is increased.

6 Conclusion

In this paper, we reported on an elementary question and answering system QUARK.

Accuracy of QUARK at present is not satisfactory.

The aim of research was to develop flexible question and answering system. But, for named entity, date and time for instance, against different answer type which estimated, different approach is needed. We are now improving QUARK, and planning to report the results elsewhere in the future.

Acknowledgment

We express our gratitude to Mainichi Shimbun, Inc. who permit us to use the Mainichi newspaper articles in a machine readable form.

References

[1] M. Murata, M. Utiyama, and H. Isahara. Question answering system using similarity-guided reasoning. In *Natural Language Processing 135-24*, pages 181–188, 2000 (in Japanese).

[2] G. Salton. *Automatic Text Processing*. Addison Wesley, 1988.

[3] Y. Sasaki, H. Isozaki, H. Taira, K. Hirota, H. Kazawa, T. Hirao, H.Nakajima, and T. Kato. An evaluation and comparision of japanese question answering system. In *IEICE Technical Report NLC2000-24*, pages 17–24, 2000 (in Japanese).