MAIQA: Mie Univ. Participated System at NTCIR4 QAC2

Naoya HIDAKA　Fumito MASUI　Keiko TOSAKI
Department of Information Engineering, Faculty of Engineering, Mie University
1515 Kamihama-cho, Tsu 514-8705, Japan
{masui,naoya}@ai.info.mie-u.ac.jp

Abstract

This paper describes a Question Answering system that participating question answering challenge 2 (QAC-2). Our system decides answer type in 200 Named Entity (NE) types. Answer candidates are ranked by score that calculated by TF・IDF and word distance between answer candidates and weighty words. For some kinds of words to be extracted as answer candidates and weighty words, we utilized part of speech tagging, named entity extraction, document retrieval.

Experiments were conducted with the formal run test set and results showed 0.217 and 0.152 MRR for subtask1, 0.154 MF for subtask2 and 0.071 and 0.068 MF for subtask3.

Keyword: *named entity, TF・IDF, word distance.*

1 Introduction

Generally, Question Answering systems consist of some techniques such as query analysis, document retrieval, named entity extraction and answer selection. When even just one technique fails, the probability of detecting correct answers goes down. And it can be considered that every technique affects other techniques.

Our system are composed of three main modules which are a query analysis module, document retrieval module and answer selection module. The query analysis module extracts weighty words from a query. And the query's answer types are decided from 200 answer types that defined by the Extended Named Entity Definition ver 6.1[1]. 8 answer types are defined on IREX[2]. But detecting correct answers is very difficult because of a wide range of the answer candidates. Then increasing answer types restricts the range of answer candidates and detecting correct answers becomes easier. The answer selection module makes answer candidates ranked by score that calculated by TF・IDF and word distance between answer candidates and weighty words.

The method based on TF・IDF and the method based on similarity of syntax information and the structural distance were compared[3]. The result of the comparison showed that the TF・IDF method can't detect correct answers of some kinds of queries. Then we utilized the TF・IDF method and the method based on word distance between answer candidates and weighty words as the answer selection module. Utilizing the word distance is for the time to respond in a real-time.

In this paper, the each module of our system is explained in Selection 2. And we conducted experiments to evaluate the performance of the system and the each module with QAC2 formal run data[4] in Selection 3. Then some topics of results are discussed in Section 4. At the end, we describe the conclusions.

2 Methods of the System

This section explains on each method of the system we implemented.

Our system consists of three main modules, which are the query analysis module, the document retrieval module and the answer selection module. The outline of the system is illustrated in Figure 1. Let us explain each module that illustrated.

2.1 Query Analysis

Query analysis module extracts a set of weighty words from query and answer types to detect the answer candidates in articles.

We utilized 200 NE answer types. The definition is that is based on the Extended Named Entity Definition ver 6.1[3].

The way to decide the answer types is explained. First, an interrogative such as "どこ (where)", "いつ(when)", "誰(who)", "何(what)", "いくら(how much)" is extracted. According to a pattern of interrogative, the answer types can be decided from the neighbor noun of interrogative or the suffix behind interrogative.

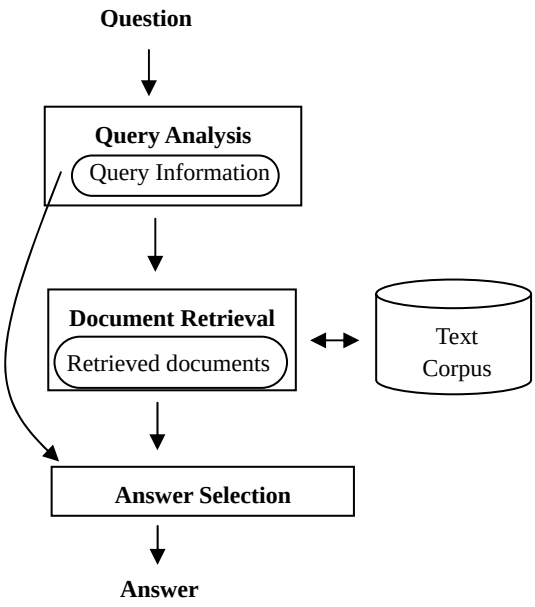

Figure 1. The outline of the system

For example, if an query is "~の首都はどこですか(What is the capital of ~)", the pattern is "はどこですか". Then the answer type is decided from the neighbor noun of interrogative, 首都(capital).And the answer type is "市区町村名 (city)".If an query is "～は何という楽器ですか (what is the instrument called ~), the pattern is "は何という". Then the answer type is decided from the noun, 楽器(instrument). And the answer type is "楽器名 (instrument)".If an query is "～は何名ですか(how many people ~), the pattern is "何名". "名" just behind interrogative is the suffix to express the number of people. Then the answer type is "人数 (the number of people)".

To utilize this answer type, all nouns in articles must be defined the same kinds of type as the answer type. Then the type definition of a named entity tagger, NExT ver0.82[5] is utilized. We customized the NE type that is 7 kinds of type at the default and 71 kinds of types are detected. For example, the NE type of "～美術館 (~art museum)" is "美術博物館名 (museum) and the NE type of "~cm" is "長さ (length)". We named this type set that is defined by NExT output "Suffix Type Set".

To prepossess for named entity extraction, part of speeech tagging is required. As the part of speech tagger, we used ChaSen ver2.3.3[6].

There are some answer types that can't be decided by NExT output, for example,"色名 (color)" like "赤(red)" and "動物名(animal)" like "キリン(giraffe)".Then we used a type dictionary and detected 155 kinds of type. We named this type set that is defined by the dictionary and NExT output "Dictionary Type Set"

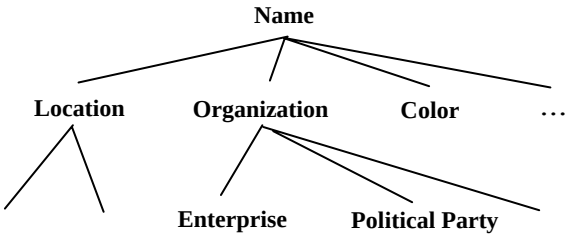

Figure 2. The tree structure of NE 200 types

These 200 answer types are a tree structure. The example of the tree structure is illustrated in Figure 2. That's why the fineness of answer type can be controlled. For example, If the answer type is "組織名 (organization)", the answer types include some answer type like "企業名 (enterprise)" and "政党名 (political party).

As the weighty words, four kinds of words are extracted. There are, in order of priority, named entity, the stem of the Sa-hen verb or adverb and noun.

2.2 Document Retrieval

Utilizing document retrieval, articles related with a set of weighty words such as named entity and noun extracted from query on Query Analysis are retrieved. Articles including weighty words are retrieved from two year newspaper articles of Mainichi and Yomiuri newspapers, about 0.6 million articles. To retrieve, the Namazu system ver2.0.12 [7] with chasen index is utilized. The reason of utilizing chasen index is why we use ChaSen system to extract weighty words.

There are some heuristic rules to solve problems. In case of retrieving no article, first, kakasi index with the same set of weighty words is utilized. Then if there's no retrieved article again, it is tried to retrieve articles with the set of weighty words from which one word having the lowest priority are erased. For the experiments, the retrieved articles ranked less than sixth were employed to be processed.

2.3 Answer Selection

Answer selection module makes answer candidates ranked by score. The score is calculated by TF･IDF and word distance between answer candidates and weighty words extracted from query. It is possible that topic words in a sentence that is similar with query expression become correct answer. The reason of using word distance is based on the concept that answer candidates near weighty words from query can be correct answers.

Then to find topic words which characterizes a retrieved document, we use TF・IDF. TF・IDF is calculated by the following formula:

$$TF(p,t)\cdot IDF(p) = TF(p,t) * \log\left(\frac{N}{df(p)} + 1\right) \quad (1)$$

where,

TF(p,t): frequency of answer candidate p in a document t

N: frequency of all document in corpus

df(p): frequency of documents containing a answer candidate p

It can be considered that answer candidates having the higher score of TF・IDF more than other answer candidate are topic words in the retrieved documents.

The score based on word distance between a answer candidate and weighty words from query in a sentence is calculated by the following formula:

$$Score(p) = \sum \frac{1}{dis(p,w)} \quad (2)$$

where,

dis(p,w): word distance between a possible word p and a weighty word w from query

For using formula(2), The score of an answer candidate with many weighty words and closer weighty words in a sentence is higher than other candidates. If answer candidates appear with no weighty word in a sentence, its score is regarded as zero. The example is shown in Figure 3. For a question of “ドイツの首都はどこですか(What is the capital of German?”, its weighty words are “ドイツ(German)” and “首都(capital)” and the answer type is “市区町村名(City)”. When one sentence in a retrieved document is “ドイツの首都ベルリンで～(In Berlin, which is the capital of German,)”, “ベルリン(Berlin)” is included as a answer candidate. And each word distance between a answer candidate and weighty words is 3 and 1. And the score of “ベルリン” is 1.333.

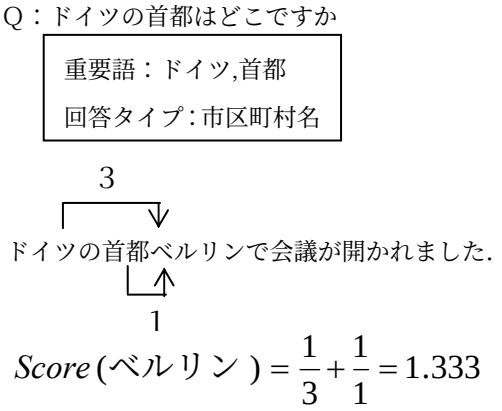

Figure 3. The example of the word distance

There are the processes to detect answer candidates. First, one document from retrieved documents is processed by part of speech tagging, named entity extraction and type definition are used. Second, all the answer candidates in the document which has the same type with answer type are extracted. The same words with words in a query are removed from answer candidates because they can’t be correct answers. Third, For all the answer candidates, The scores of TF・IDF and word distance are measured. The score of word distance for every answer candidate is given a maximum score for every answer candidate in a document. Then, answer candidates ranked in the top 10 in order of TF・IDF*Score(p) in a document are output as document’s rank data. It can be thought that correct answer for query is included with in the top 10 and other answer candidates become noises. This process is done for all the retrieved documents and output every document’s rank data. Finally, the document’s rank data are integrated, and answer candidates ranked in the top five in order of the score are extracted as the system’s output. If there’s no answer candidate in all retrieved documents, a answer type goes to upper class and the same process is done.

3 Experiments

To learn our system's performance, we conducted experiments with QAC2 formal run subtask 1, subtask 2 and subtask 3 test set [4].

As the methods for subtask 2, we use the threshold for the integrated rank data on the

Table 1. Results of evolution

	Question	Answer	Output	Correct	Recall	Precision	F-value	MRR	MF
MAIQA1-1	197	385	881	64	0.166	0.073	0.101	0.217	-
MAIQA1-2	197	385	798	46	0.119	0.058	0.078	0.152	-
MAIQA2	200	647	472	84	0.130	0.178	0.150	-	0.154
MAIQA3-1	251	539	1152	65	0.121	0.056	0.077	-	0.082
MAIQA3-2	251	539	1135	61	0.113	0.054	0.073	-	0.078

system. The threshold is based on the highest value that divided the score of the n in rank by the score of the n+1 in rank. The threshold is the n in rank that extracted the highest value and answer candidates ranked in top n are output as system's answer for subtask 2.

The methods for subtask 3 are explained as follows. To extract correct answer f of the related question, we use the same retrieved documents for related questions as that of the main question. As the weighty words for each related question, both weighty words from the main question and weighty words from the related question. And the answer type for each related question is extracted from each related question.

Both test sets of subtask 1 and subtask 2 include 200 queries and test set of subtask 3 include 36 main queries and 4-8 related queries for each main query, total 251 queries. As measures, mean reciprocal rank (MRR) was used for subtask1 and mean F-measure (MF) was used for subtask2 and subask3.

We utilized two systems for subtask1. One is the system that using suffix type set as a type set (MAIQA1-1), the other is the system that using dictionary type as type set (MAIQA1-2). We utilized one system for subtask1 that using Suffix Type Set as a type set (MAIQA2). We utilized two systems for subtask3, both systems use Suffix Type Set as a type. One is the system using the method for subtask3 (MAIQA3-1), the other system is the same system for subtask1 (MAIQA3-2).

The result of evaluation is shown in Table 1. For subtask1, 64 of 881 MAIQA1-1 system's outputs were correct answer and 46 of 798 MAQA1-2 system's outputs were correct answer. The total performances of the MAIQA1-1 and MAIQA1-2 systems are 0.217 and 0.152 MRR points. For subtask2, the total performance of the MAIQA2 system is 0.154 MF points. For subtask3, the total performances of the MAIQA3-1 and MAIQA3-2 system are 0.071 and 0.068 MF points.

Ranking our system in participated systems results of subtask1, MRR of our system ranks 20 th (MAIQA1-1) and 24 th (MAIQA1-2) out of 25 systems.

4 Discussions

We describe every module's performance with our system's (MAIQA1-1) outputs.

On the query analysis module, the number of correct answer types that decided is 174 out of 200 queries for subtask.1. In the correct answer types, some answer types were decided in detail, others were decided widely. For example, the answer type of "スパイダーマンはどこの国の漫画ですか(Which country does the cartoon Spiderman come from?)" is "国名(country)" and it's the detail type. The answer type of "明石海峡大橋の愛称は何ですか(What is the nickname for the Akashi Kaikyo Bridge?)" is "名前(name)"and it's the wide type. As the main cause of incorrect answer types, it is thought that is a shortage of query pattern. When there's no pattern to match the query, the answer type is decided as "名前(name)".For example, the correct answer type of "新宿センタービルは何階建てですか(How many stories does Shinjuku Center Building have?)" is "numeral", but the answer type was decided as "名前" because there's no this kind of pattern and the answer type was defined from "何". The query analysis module will be better by increasing query patterns.

To correspond the answer type that can't be defined in suffix type, we utilized dictionary type in the MAIQA1-2 system. But the performance wasn't good.

On the retrieval document module, the number of the query that retrieved the documents include correct answers is 113 out of 200 queries. The

number of documents include correct answers is 222 out of 800 retrieved documents and the ratio is 27.8%. As the cause of this low ratio, it is considered that the choice of weighty words from query was wrong and the precision of Namazu system is bad. To improve the retrieval document module, it is thought that reconsidering the way to extract weighty words from query, changing the number of retrieved documents or utilizing other system to retrieve.

Some correct answers were not detected because the answers were divided into more than two words. For example, “パールブリッジ (Pearl Bridge)” were divided into “パール (Pearl)” and “ブリッジ(Bridge)”,”横浜ＦＣ (Yokohama FC)” were divided into “横浜 (Yokohama)” and “ＦＣ”.

The number of queries that done the correct operation on the query analysis module, the retrieval document module and named entity is 98. In the 98 queries, the number of correct answers that detected in the answer selection module is 64. It can be said that the performance of the answer selection is not bad. As the cause of incorrect answers, it can be considered that the number of retrieval document include correct answer in a query was small and the answer type was “名前(name)”. And the balance between TF・IDF and Score of word distance should be considered.

Considering the circumstances mentioned above, if the performance of every module is not good, it is difficult to have good performance of QAC system.

5 Conclusions

We described a question answering system. We implemented 200 NE types as the answer type. And our system extracts an answer by the score that TF・IDF and word distance between answer candidates and weighty words from query.

The Result of experiments we conducted show 0.217 and 0.152 MRR for subtask1, 0.154 MF for subtask2 and 0.071 and 0.068 MF for subtask3.

As the performance of each module of the system, Query Analysis module shows the high performance to decide answer type. But answer types are needed to fit NE types more. Document retrieval module’s performance is bad. Then we need to reconsider the way to retrieve documents. Answer Section module’s performance is not bad. To improve the module, the score based on sentence distance to detect answer is also thought.

References

[1] S. Sekine. Extended Named Entity Definition version 6.1. http://apple.cs.nyu.edu/~sekine/PROJECT /NEH/version6_1_0.html

[2] Information Retrieval and Extraction Exercise http://www.csl.sony.co.jp/person/sekine/IREX/

[3] N. Hidaka. F. Masui. A Comparison of Answer Ranking Methods in Question Answering. The 17th Annual Conference of the Japanese Society for Artificial Intelligence, 2003.

[4] J. Fukumoto, T. Kato and F. Masui. Qac task home page. http://www.nlp.cs.ritsumei.ac.jp/qac/, 2003.

[5] I. Watanabe, F. Masui and J. Fukumoto. NExT – a Named Entity Extraction Tool. http://www.ai.info.mie-u.ac.jp/~next/next.html, 2003

[6] Y. Matsumoto, H. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka and M. Asahara. User’s Manual for morphological analysis system “Chasen” version 2.3.3. Naist technical report, Nara Advanced Institute Science and Technology, 2003

[7] Namazu Project. Namazu: a full-text search engine. http://www.namazu.org/, 2003