Question Answering System Based on Expanded Answer Types and Multi-Scores

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

Abstract

In this paper, we describe our Question Answering System. We proposed to use 200 answer types to abate answer types' ambiguity in Query Analysis. And as the score of extracting correct answers, we proposed TF・IDF and a word distance between an answer candidate and a weighty word from a query.

Comparative experiments were conducted with QAC1 and QAC2 formal run test set. The result showed that 200 answer types are more effective than 5 answer types to decide the correct answer type and reduce the number of answer candidates. Also, the score of TF・IDF and a word distance was more effective than one score to extract many correct answers on higher rank.

Keyword: *200 answer types, TF・IDF, Word Distance*

1 Introduction

Recently, many researchers are interested in the study of Question Answering (QA). As conferences on QA systems, the Question Answering Challenge (QAC)[1][2][3] and the Text Retrieval Conference (TREC)[13] has been held in Japan and the U.S.A respectively.

Generally, Question Answering (QA) systems consist of several techniques such as Query Analysis, Document Retrieval, and Answer Selection. In Query Analysis, most QA systems decide answer types. However, the number of answer types is different on each system. Utilizing a large number of answer types have been proposed[4][6]. These answer types are mostly based on Named Entity types or dictionaries. In Answer Selection, Several scores to detect correct answers have been proposed. The scores are based on their frequency[5], distance between a word from a query and answer candidates[9] or the similarity between the dependency structures of queries and answer candidates [4][8][9].

Our QA system that proposed in previous work[5] had also standard architecture. It was composed of three main modules: Query Analysis, Document Retrieval and Answer Selection. In the Query Analysis, answer types were decided with 5 answer types called Type A. But on QA, detecting correct answers with only these 5 answer types was very difficult because of a wide range of the answer candidates. To solve this problem, we propose to increase answer types to restrict the range of answer candidates.

In the Answer Selection, answer candidates had been ranked by score of *TF・IDF*. We assumed that topic words in the retrieved documents were likely to become correct answers. But the *TF・IDF* method couldn't detect correct answers for some kinds of queries. To solve this problem, we combine the *TF・IDF* method with the method based on word distance between answer candidates and weighty words extracted from the query.

In this paper, our QA system that proposed in previous work[5] and its problems are explained in Section 2. We propose the method to improve our previous system in Section 3. We conduct experiments to show the superiority of the system in Section 4. Then some topics of results are discussed in Section 5. At the end, we describe the conclusions.

2 Our Previous System and its Problems

This section explains the previous QA system that we implemented and its problems. We utilized 5 answer types called Type A in Query Analysis and a *TF・IDF* method in Answer Selection. Our system consists of three main modules: Query Analysis, Document Retrieval and Answer Selection. The outline of the system is illustrated in Figure 1.

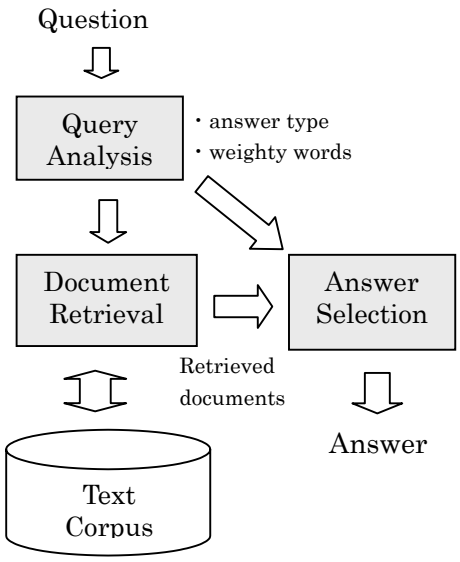

Figure1. The outline of the system

2.1 Answer type's problem in Query Analysis

Query Analysis extracts answer types to detect the answer candidates in articles.

Answer types are based on Named Entity (NE) types. We utilized 5 answer types (Type A). The answer types of Type A are "Person", "Location or Organization", "Date", "Numeral" and "Others". Type A depends solely on the interrogatives. For example, when the interrogative is "where", "Location or Organization" as the answer type is decided.

We evaluated the Query Analysis of Type A with 200 queries of QAC1 formalrun test data[1]. The number of correct answer types is 178 out of 200 queries. The result is shown in Table 1.

Table1. The result of Type A

Answer type	Correct
Person	39/42
Location or Organization	43/44
Date	15/18
Numeral	14/26
Others	67/70
Total	178/200

As the reason of incorrect answer types, these answer types couldn't be decided by just the interrogative. The example Figure 2shows the example. As you can see from Table 1, many correct answer types were extracted, but some answer types have ambiguity. When there is answer type's ambiguity, the answer candidates increase. For example, the answer type of "What

Q: 明石海峡大橋の全長は何メートルですか
(How many meters is the Akashi Kaikyo Bridge?)
↓ answer type
Others
(decided incorrectly by "何(what)" on Type A)

Figure2. The example of incorrect answer type

is the capital of Germany?" is "Location or Organization". This answer type is correct, but NE words of Location and Organization are included as answer candidates. Then it is difficult to extract a correct answer because of many answer candidates. If this answer type is only "Location" or more detailed type of "Location" like "City", it should be easier to find correct answers. Then we propose to increase answer types to restrict the range of answer candidate.

2.2 Document Retrieval

Document Retrieval is based on an IR system. By utilizing document retrieval, articles related with a set of weighty words such as named entity and noun extracted from a query on Query Analysis are retrieved. To retrieve the articles, the Namazu system ver2.0.12 [12] were used.

2.3 TF・IDF Score on Answer ranks

Answer Selection makes answer candidates ranked by score. The score was calculated based on *TF・IDF*. We assumed that topic words in a document are likely to become correct answers.

To find topic words, we use *TF・IDF*. *TF・IDF* is calculated by the following formula:

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

Here,

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

N: frequency of all documents in corpus

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

Answer candidates having the higher *TF・IDF* score than other candidates must be topic words in the retrieved documents.

To evaluate a performance of the *TF・IDF* method itself, queries that couldn't retrieve documents including correct answers are eliminated in QAC1 formalrun 200 queries[1]. Then, 103 queries remained. The result of the *TF・IDF* method is shown in Table 2.

Table 2. The result of the TF・IDF method

Questions	Correct	MRR
103	52	0.280

As the reason of incorrect answers, when there are some topics in retrieved documents, there are many answer candidates and *TF* score for each answer candidate is low. Then, it's difficult to extract correct answers. And since the *TF・IDF* method has nothing to do with a query expression, answer candidates that don't relate with a query were extracted. Also, answer type's ambiguity would affect Answer Selection.

To solve this problem, we needed to think about the relationship with a query expression. Then, we propose a word distance between answer candidates and a word in a query as the score to extract correct answers.

3 Improvement of Our System

In this section, we propose the method to improve our previous QA system that utilized 5 answer types called Type A and the *TF・IDF* method. We propose to increase answer types to solve the problem of answer types' ambiguity in Type A in Section 3.1. And we propose to utilize a word distance to think about the relationship with a query expression in Section 3.2

3.1 Expanded Answer Types in Query Analysis

To abate the answer type's ambiguity, we utilized 200 NE answer types (Type B). The answer types of Type B are based on the Extended Named Entity Definition ver 6.1[10]. Figure 3 shows Type B has a hierarchy structure.

In Type B, there are 3 steps to decide the answer type. First, an interrogative such as "Where, When, Who, What or How much" is extracted. Second, according to the pattern of interrogatives, the answer types can be decided from the neighbor noun of the interrogatives or the suffix behind the interrogatives. There are 53 patterns of interrogatives like "□□□□□□□ (Where is ~)" or "~□□□□□ (What is ~)". These patterns were created by human-handed work

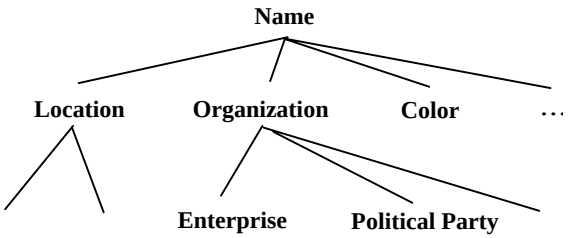

Figure3. The hierarchy structure of NE 200 types

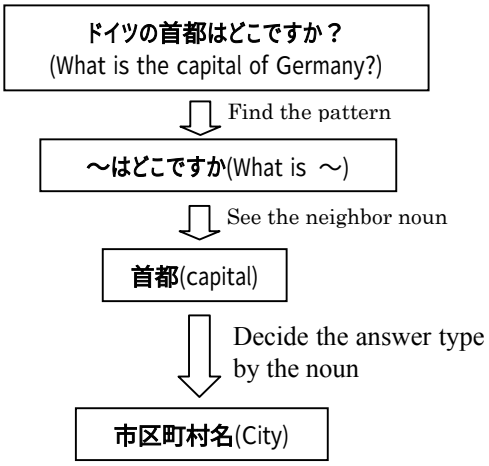

Figure4. The example of deciding answer type

with QAC1 formalrun data. Finally, the answer type is judged by the noun or suffix. If the noun or suffix that distinguishes answer types can't be found, the answer type is decided by the solely interrogatives. The example of deciding answer type is shown in Figure 4.

To utilize these answer types, all nouns in articles must be defined the same kinds of type as the answer type. Then the type's definition of a named entity tagger, NExT ver0.82[11] is utilized. We increased the NE types from 7 NE types at the default to 71 NE types. For example, the NE type of "~ art museum" is "Museum" and the NE type of "～cm" is "Length".

To prepossess Named Entity extraction, part of speech tagging is required. As a part of speech tagger, we used ChaSen ver2.3.3[7].

There are some answer types that can't be decided by NExT output, for example, "Color" like "red" and "Animal" like "giraffe". Then we used a type dictionary and detected 155 kinds of type. In this dictionary, there are some nouns that have plural NE types. Then we can't judge which NE types are right at this point, so we

allow filling all the plural types.

Since Type B is composed of hierarchy structure, the fineness of answer type can be controlled. For example, if the answer type is "Organization", the answer types include some answer type like "Enterprise" and "Political party".

3.2 Multi-Scores on Answer Selection

To improve the *TF・IDF* method, we also utilized the word distance between answer candidates and weighty words extracted from a query expression as the score. We assumed that a sentence including correct answers and a query expression are similar. The score that based on the word distance between an answer candidate and weighty words from a query in a sentence is calculated by the following formula:

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

Here,
p: an answer candidate
w: a weighty word from a query expression
dis(p,w): the word distance between *p* and *w*

WordDis(p) of an answer candidate that exists nearer weighty words in a sentence is higher than that of other candidates. If answer candidates appear with no weighty word in a sentence, the score is decided as zero. The example is shown in Figure 3. For a query, "What is the capital of Germany?", "Germany" and "capital" are extracted as the weighty words and "City" is decided as the answer type. When one sentence in a retrieved document is "In Berlin, which is the capital of Germany, ~", "Germany" and "capital" are realized as the weighty words and "Berlin" is realized as an answer candidate. Then, each word distance between an answer candidate and weighty words is calculated as 3 and 1. Also, the *WordDis(p)* of an answer candidate "Berlin" is calculated as 1.333.

To rank the answer candidates more precisely, utilizing the both score of *TF(p,t)・IDF(p)* and *WordDis(p)* would be better than using one score independently. As the product of *TF(p,t)・IDF(p)* and *WordDis(p)*, *Score(p)* is used. And the formula is given below:

$$Score(p) = TF(p) \cdot IDF(p,t) \times WordDis(p) \quad (3)$$

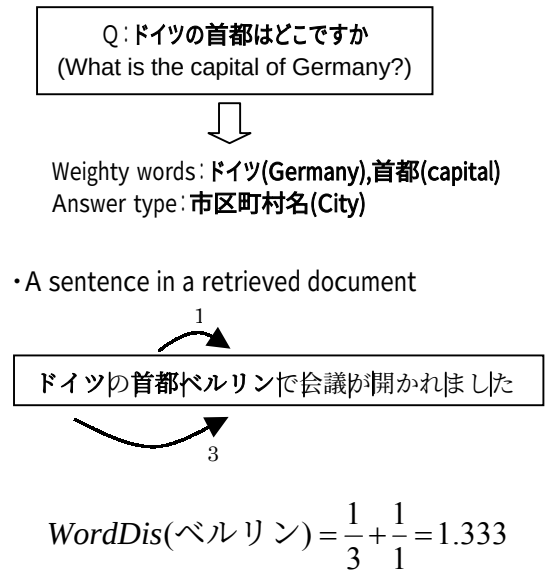

Figure5. The example of the word distance

4 Experiments

To evaluate the improved system, we conducted experiments with QAC1[1] and QAC2 formal run test set [2].

On Query Analysis module, we utilized two kinds of answer types, Type A and Type B. The number of the correct answer types is shown in Table3.The number of answer types that detailed more than Type A are shown in Table3. The number in parentheses shows the number of correct answer types in Type B that was incorrect answer types in Type A in Table4.

In Answer Selection module, we utilized three systems that are ranked by *TF(p,t)・IDF(p)* only, *WordDis(p)* and *Score(p)*. Also Type B answer types were used. To evaluate only answer selection, queries that couldn't retrieve documents including correct answers are eliminated in QAC1 and QAC2 formalrun 200 queries. Then, 103 queries remained on QAC1 and 117 queries remained on QAC2. The number of correct answers and MRR are shown in Table 5.

5 Discussions

In Query Analysis, the results of Table3 and Table4 show that Type B distinguished answer types correctly more than Type A. Concerning with the incorrect answer types on Type A, the answer types were decided as "Others" because of the interrogative "何(what)" incorrectly. For example, the correct answer type of "若乃花(花

Table3. The Correct answer type

	Type A	Type B
QAC1	178	195
QAC2	178	194

Table4. The Detailed type in Type B

	QAC1	QAC2
Person	(3)	(6)
Location or Organization	37(1)	25
Date	5(1)	9(4)
Numeral	21(13)	18(9)
Others	21	21(1)
Total	87(18)	77(20)

田勝)は第何代の横綱ですか(What number yokozuna is Wakanohana (Hanada Masaru)?" is "Numeral" and the correct answer type of "小野寺章太郎って本名は何ですか？(What is Onodera Shotaro's real name?)" is "Person" in Type B. But both answer types were "Others" in Type A and it's incorrect.

Moreover, by utilizing Type B, about 84 answer types out of 200 queries were decided more precisely than utilizing Type A. Then, it would be easy to extract correct answers. Therefore, Type B is more effective than Type A.

In Type B, most of answer types can be decided correctly, but some answer types are not necessary because of the shortage of the number of words that filled with the answer types. Then, we would need to find the proper number of answer types.

In Answer Selection, Table5 shows that the number of queries that extracted correct answers on QAC1 were 64, 73 and 79 respectively out of 103 queries on *TF(p,t)･IDF(p)*, *WordDis(p)* and *Score(p)*. The number of queries extracted correct answers on QAC2 is 58, 71 and 74 respectively out of 117 queries on *TF(p,t)・IDF(p)*, *WordDis(p)* and *Score(p)*. Judging from this result, utilizing *Score(p)* could extract more correct answers than utilizing *TF(p,t)･IDF(p)*. Also, when *Score(p)* is compared to *WordDis(p)*, both of the number of queries including correct answers don't have a big difference, but Score could get a higher MRR score than *WordDis(p)*.

Table5. The number of correct answers

QAC1

	TF・IDF	WordDis	Score
Questions	64	73	79
MRR	0.473	0.534	0.572

QAC2

	TF・IDF	WordDis	Score
Questions	58	71	74
MRR	0.328	0.437	0.479

This result shows that utilizing Score can extract correct answers on higher rank than utilizing *WordDis(p)*. Therefore, Utilizing *TF(p,t)･IDF(p)* and *WordDis(p)* at the same time is more effective than utilizing one score.

This time, we used a word distance to measure the similarity of answer candidates to a query expression, but there is a problem on some queries. When a correct answer and weighty words appear in different sentences in retrieve documents, *WordDis(p)* of the correct answer is zero. Thus, the correct answer can't be extracted on high rank. From this result, to improve this problem, we need to use a sentence distance or a passage distance to measure the proximity on this kind of query.

6 Conclusion

In this paper, we proposed to utilize 200 answer types to abate answer types' ambiguity and the score of *TF・IDF* and a word distance between answer candidates and weighty words from a query to think about the relation ship with a query expression.

The result of experiments showed that 200 answer types are more effective than 5 answer types to decide the correct answer type and reduce the number of answer candidates. Also, the score of TF・IDF and a word distance was more effective than to extract many correct answers on higher rank.

As the future work, we will find the proper number of answer types on Query Analysis. And we will think to use a sentence distance or a passage distance to measure the proximity on this kind of query on Answer Selection. And as the whole performance of our QA system, we also need to improve the low performance of the Document Retrieval module.

References

[1] J. Fukumoto, T. Kato and F. Masui. Question Answering Challenge(QAC-1): Question answering evaluation at ntcir wokshop 3. In Working Notes of the Third NTCIR Workshop Meeting Part IV: Question Answering Challenge(QAC-1), pages 1-10,2002.

[2] J. Fukumoto, T. Kato and F. Masui. Question Answering Challenge for Five ranked answers and List answers –Overview of NTCIR4 QAC2 Subtask 1 and 2-: In Working Notes of the Fourth NTCIR Workshop Meeting: Question Answering Challenge(QAC-2), pages 283-290,2004.

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

[4] S. M. Harabagiu, D. I. Moldovan, M.Pasca, R. Mihalcea, M. Surdeanu, R. C. Bunescu, R. Girju, V. Rus and P. Morarescu. Falcon: Boosting Knowledge for answer engines: In Proceedings of ninth Text Retrieval Conference (TREC9), pages479-488, 2000.

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

[6] H. Isozaki. NTT's Question Answering System for NTCIR QAC2: In Working Notes of the Fourth NTCIR Workshop Meeting: Question Answering Challenge(QAC-2), pages 326-332,2004

[7] 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

[8] M. Murata, M. Utiyama and H. Isahara. Question answering system using similarity-guided reasoning. SIG Notes 2000-NL-135, Information Processing Society of Japan, Jan. 2000.

[9] Y. Sasaki, H. Isozaki, T. Hirao, K. Kokuryou and E. Maeda. NTT's QA Systems for NTCIR QAC-1. In Working Notes of the Third NTCIR Workshop Meeting Part IV: Question Answering Challenge(QAC-1), pages 63-70,2002.

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

[11] 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

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

[13] TREC Home pape. http://trec.nist.gov/,2003