Japanese Question-Answering System Using Decreased Adding with Multiple Answers at NTCIR 5

Masaki Murata, Masao Utiyama, and Hitoshi Isahara
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{murata, mutiyama, isahara}@nict.go.jp

Abstract

We propose a new method of using multiple documents with decreasing weights as evidence to improve the performance of a question-answering system. Sometimes, the answer to a question may be found in multiple documents. In such cases, using multiple documents for prediction would generate better answers than using a single document. Thus, our method employs information from multiple documents by adding the scores of the candidate answers extracted from the various documents. Because simply adding scores degrades the performance of question-answering systems, we add scores with decreasing weights to reduce the negative effect of simple adding. We used this method at NTCIR 4 and NTCIR 5, where it obtained very good results. The three systems that we submitted to NTCIR 5 obtained the highest scores among the 16 systems that participated in the conference.

Keywords: *Multiple Documents, Decreased Adding, Combined Method*

1 Introduction

A question-answering system is an application designed to produce the correct answer to a question given as input. For example, when "What is the capital of Japan?" is given as input, a question-answering system may retrieve text containing sentences like "Tokyo is Japan's capital and the country's largest and most important city. Tokyo is also one of Japan's 47 prefectures." from websites, newspaper articles, or encyclopedias. The system then outputs "Tokyo" as the correct answer. We expect question-answering systems to become a more convenient alternative to other systems designed for information retrieval and as a basic component of future artificial intelligence systems. Recently, many researchers have been attracted to this important topic. These researchers have produced many interesting studies on question-answering systems [4, 3, 1, 2, 5, 7]. Evaluation conferences, or contests, on question-answering systems have also been held. In particular, the U.S.A. has held the Text REtrieval Conferences (TREC) [19], and Japan has hosted the Question-Answering Challenges (QAC) [15]. These conferences aim to improve question-answering systems. For such conferences and contests, researchers make question-answering systems and use them to answer the same questions, and each system's performance at the conference is then examined to glean possible improvements. We have investigated the potential of question-answering systems [10] and studied their construction by participating in the QAC [15] at NTCIR 3 [11].

At NTCIR-4, we proposed a new method of using multiple documents with decreased weightings as evidence. Sometimes, the answer to a question may be found in multiple documents. In such cases, using multiple documents for prediction would generate a better answer than using only one document [1, 2, 5, 18]. In our method, information from multiple documents is employed by adding the scores for the candidate answers extracted from the various documents [2, 18]. Because simply adding the scores degrades the performance of a question-answering system, our method adds the scores with decreasing weights to overcome the problems of simple addition. More concretely, our method multiplies the score of the i-th candidate answer by a factor of $k^{(i-1)}$ before adding the score to the running total. The final answer is then determined based on the total score. For example, suppose that "Tokyo" is extracted as a candidate answer from three documents and has scores of "26", "21", and "20", and assume that k is 0.3. In this case, the total score for "Tokyo" is "34.1" (= 26 + 21 $\times$ 0.3 + 20 $\times$ 0.3^2). Thus, we calculate the score in the same way for each candidate and take the answer with the highest score as the correct answer. We also used this method at NTCIR-5, and it obtained very good results at both conferences. In fact, it obtained the highest scores among the participants at NTCIR 5.

Table 1. Candidate answers according to the original scores, where "Tokyo" is the correct answer.

Rank	Candidate answer	Score	Document ID
1	Kyoto	3.3	926324
2	Tokyo	3.2	259312
3	Tokyo	2.8	451245
4	Tokyo	2.5	371922
5	Tokyo	2.4	221328
6	Beijing	2.3	113127
...	...	...	...

Table 2. Candidate answers with simply added scores, where "Tokyo" is the correct answer.

Rank	Cand. ans.	Score	Document ID
1	Tokyo	10.9	259312, 451245, ...
2	Kyoto	3.3	926324
3	Beijing	2.3	113127
...	...	...	...

Table 3. Candidate answers according to the original scores, where "Kyoto" is the correct answer.

Rank	Cand. ans.	Score	Document ID
1	Kyoto	5.4	926324
2	Tokyo	2.1	259312
3	Tokyo	1.8	451245
4	Tokyo	1.5	371922
5	Tokyo	1.4	221328
6	Beijing	1.3	113127
...	...	...	...

Table 4. Candidate answers with simply added scores, where "Kyoto" is the correct answer.

Rank	Cand. ans.	Score	Document ID
1	Tokyo	6.8	259312, 451245, ...
2	Kyoto	5.4	926324
3	Beijing	1.3	113127
...	...	...	...

2 Use of Multiple Documents as Evidence with Decreased Weighting

Suppose that the question, "What is the capital of Japan?", is input to a question-answering system, with the goal of obtaining the correct answer, "Tokyo". A typical question-answering system would output the candidate answers and scores listed in Table 1. These systems also output a document ID indicating the document from which each candidate answer was extracted.

For the example shown in Table 1, the system outputs an incorrect answer, "Kyoto", as the first answer.

A previous method based on simply adding the scores of candidate answers has been evaluated [2, 18]. For our current example question, this method produces the results shown in Table 2. It outputs the correct answer, "Tokyo", as the first answer, and thus it can obtain correct answers by using multiple documents as evidence.

The problem with this method, however, is that it is likely to select candidate answers with high frequencies. This is a serious problem from a performance standpoint. That is, when a method has good inherent performance, the original scores that it outputs are often more reliable than the simply added scores; hence simply adding scores often degrades the method's performance.

To overcome this problem, we developed a method of using multiple documents with decreased weightings as evidence. Instead of simply adding the scores of the candidate answers, the method adds scores by assigning decreasing weights to them. This approach reduces the negative effect of the system being likely to select candidate answers with high frequencies, while still improving the accuracy of the system by adding the scores.

We can demonstrate the effect of our method by giving an example. Suppose that a question-answering system outputs Table 3 in response to the question, "What was the capital of Japan in A.D. 1000?". The correct answer is "Kyoto", and the system outputs the correct answer as the first answer.

When we simply add scores, however, we obtain the results shown in Table 4. In this case, the incorrect answer, "Tokyo", achieves the highest score.

To overcome this problem, we can try to apply our proposed method of adding candidate scores with decreasing weights. Now suppose that we implement our method by multiplying the score of the i-th candidate by a factor of $0.3^{(i-1)}$ before adding up the scores. In this case, the score for "Tokyo" is 2.8 (= $2.1 + 1.8 \times 0.3 + 1.5 \times 0.3^2 + 1.4 \times 0.3^3$), and we obtain the results shown in Table 5. The correct answer, "Kyoto", achieves the highest score, while the score for "Tokyo" is notably lower.

We can also apply our method to the first example question, "What is the capital of Japan?". When we use our method, the score for "Tokyo" is 4.3 (= $3.2 + 2.8 \times 0.3 + 2.5 \times 0.3^2 + 2.4 \times 0.3^3$), and we obtain the results shown in Table 6. As expected, "Tokyo" achieves the highest score.

Our method of adding candidate answer scores multiplied with decreasing weights successfully ob-

Table 5. Candidate answers obtained by adding scores with decreasing weights, where "Kyoto" is the correct answer.

Rank	Cand. ans.	Score	Document ID
1	Kyoto	5.4	926324
2	Tokyo	2.8	259312, 451245, ...
3	Beijing	1.3	113127
...	...	...	...

Table 6. Candidate answers obtained by adding scores with decreasing weights, where "Tokyo" is the correct answer

Rank	Cand. ans.	Score	Document ID
1	Tokyo	4.3	259312, 451245, ...
2	Kyoto	3.3	926324
3	Beijing	2.3	113127
...	...	...	...

tained the correct answers to each of the example questions. This suggests its feasibility for reducing the effect of a question-answering system being likely to select candidate answers with high frequencies, while at the same time improving the system's accuracy.

3 Question-answering Systems of This Study

The system utilizes three basic components:

1. Prediction of answer type

   The system predicts the answer to be a particular type of expression, based on whether the input question is indicated by an interrogative pronoun, an adjective, or an adverb. For example, if the input question is "Who is the prime minister of Japan?", the expression "Who" suggests that the answer will be a person's name.

2. Document retrieval

   The system extracts terms from the input question and retrieves documents by using these terms. The retrieval process thus gathers documents that are likely to contain the correct answer. For example, for the input question "Who is the prime minister of Japan?", the system extracts "prime", "minister", and "Japan" as terms and retrieves documents accordingly.

3. Answer detection

   The system extracts linguistic expressions that match the predicted expression type, as described above, from the retrieved documents. It then outputs the extracted expressions as candidate answers. For example, for the question "Who is the prime minister of Japan?", the system extracts person's names as candidate answers from documents containing the terms "prime", "minister", and "Japan".

3.1 Prediction of answer type

3.1.1 Heuristic rules

The system applies manually defined heuristic rules to predict the answer type. There are 39 of these rules. Some of them are listed here:

1. When *dare* "who" occurs in a question, a person's name is given as the answer type.

2. When *itsu* "when" occurs in a question, a time expression is given as the answer type.

3. When *donokurai* "how many" occurs in a question, a numerical expression is given as the answer type.

3.2 Document retrieval

Our system extracts terms from a question by using a morphological analyzer, ChaSen [6]. The analyzer first eliminates terms whose part of speech is a preposition or a similar type; it then retrieves by using the extracted terms.

The documents are retrieved as follows:

We first retrieve the top k_{dr1} documents with the highest scores calculated from the equation

$$Score(d) = \sum_{\text{term } t} \left(\frac{tf(d,t)}{tf(d,t) + k_t \frac{length(d) + k_+}{\Delta + k_+}} \times log \frac{N}{df(t)} \right) \quad (1)$$

where d is a document, t is a term extracted from a question, $tf(d,t)$ is the frequency of t occurring in document d, $df(t)$ is the number of documents in which t appears, N is the total number of documents, $length(d)$ is the length of d, and Δ is the average length of all documents. k_t and k_+ are constants defined according to experimental results. We based this equation on Robertson's equation [16, 17]. This approach is very effective, and we have used it extensively for information retrieval [9, 14, 8]. The question answering system uses a large number for k_t.

Next, we re-rank the extracted documents according to the following equation and extract the top k_{dr2} documents, which are used in the ensuing answer extraction phase.

$$
\begin{aligned}
& Score(d) \\
= \; & -min_{t1 \in T} log \prod_{t2 \in T3} (2dist(t1,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \\
= \; & max_{t1 \in T} \sum_{t2 \in T3} w_{dr2}(t2) log \frac{N}{2dist(t1,t2) * df(t2)}
\end{aligned}
\tag{2}
$$

$$
T3 = \{t | t \in T, 2dist(t1,t)\frac{df(t)}{N} \leq 1\}, \tag{3}
$$

where d is a document, T is the set of terms in the question, and $dist(t1, t2)$ is the distance between $t1$ and $t2$ (defined as the number of characters between them) with $dist(t1, t2)$ = 0.5 when $t1 = t2$. $w_{dr2}(t2)$ is a function of $t2$ that is adjusted according to experimental results.

Because our system can determine whether terms are near each other by re-ranking them according to Eq. 2, it can use full-size documents for retrieval. In this study, we extracted 20 documents for retrieval. The following procedure for answer detection is thus applied to the 20 extracted documents.

3.3 Answer detection

To detect answers, our system first generates candidate expressions for the answer from the extracted documents. We initially used morpheme n-grams for the candidate expressions, but this approach generated too many candidates. Instead, we now use candidates consisting only of nouns, unknown words, and symbols. Moreover, we use the ChaSen analyzer to determine morphemes and their parts of speech.

Our approach to judging whether each candidate is a correct answer is to add the score ($Score_{near}(c)$) for the candidate, under the condition that it is near an extracted term, and the score ($Score_{sem}(c)$) based on heuristic rules according to the answer type. The system then selects the candidates having the highest total points as the correct answers.

We used the following method to calculate the score for a candidate c under the condition that it must be near the extracted terms.

$$
\begin{aligned}
Score_{near}(c) &= -log \prod_{t2 \in T3} (2dist(c,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \\
&= \sum_{t2 \in T3} w_{dr2}(t2) log \frac{N}{2dist(c,t2) * df(t2)}
\end{aligned}
\tag{4}
$$

$$
T3 = \{t | t \in T, 2dist(c,t)\frac{df(t)}{N} \leq 1\}
$$

Table 7. Results of Formal Run

System ID	Total	First	Rest
NICT1	0.236	0.403	0.209
NICT2	0.250	0.450	0.218
NICT3	0.208	0.403	0.177

Table 8. Results of Reference 1 Run

System ID	Total	First	Rest
NICT1	0.305	0.403	0.289
NICT2	0.314	0.450	0.292
NICT3	0.305	0.403	0.289

where c is a candidate for the correct answer, and $w_{dr2}(t2)$ is a function of $t2$, which is adjusted according to experimental results.

Next, we describe how the score ($Score_{sem}(c)$) is calculated based on heuristic rules for the predicted answer type. We use 45 heuristic rules to award points to candidates and utilized the total points as the score. Some of these rules are listed below:

1. Add 1000 to the candidates when they match one of the predicted answer types (a person's name, a time expression, or a numerical expression). We use named entity extraction techniques based on the support-vector machine method to judge whether a candidate matches a predicted answer type [20]. We use only five named entities, as in our previous system [11].

2. When a country name is one of the predicted answer types, add 1000 to the candidates found in our dictionary of countries, which includes the names of almost every country (636 expressions).

3. When the question contains *nani* Noun X "what Noun X", add 1000 to the candidates having the Noun X.

An additional function of compiling similar answers may be used after the answers have been selected based on their scores. This function compiles those answers that are part of other answers and whose difference is less than 90% of the best score. The answers are compiled by eliminating answers other than the longest one. We distinguish the method when it uses this additional function by calling it *rate-based answer compiling.*

4 Experiments using the QAC3 data collection

The experimental results are listed in Tables 7 and 8. The methods used $k = 0.3$. QAC 3 was a series of questions having contexts. We used as the current question the concatenation of all the questions from the first question to the current question sentence in which an interrogative pronoun, adjective, or adverb in the first sentence is changed to dummy symbols. We also used the *select-by-rate method* proposed by us in the QAC2 contest. The *Select-by-rate method* outputs the answers having a score more than a certain rate (Rate for selection) of the highest score. NICT 1 and NICT 3 used 0.95 as the rate for selection, whereas NICT 2 used 0.9. NICT 3 used an additional function that added the answers to previous questions in the same series to the current question.

Table 7 shows the results of the formal run. Table 8 shows the results of the reference 1 run. The reference 1 run is a special situation where the omitted expressions (e.g. pronouns) were supplemented in all the question sentences; hence, it is easier than the formal run.

The three systems that we submitted obtained the highest scores among the 16 systems that participated in NTCIR 5. They were thus very effective question-answering systems.[1]

5 Conclusions

We proposed a method of using multiple documents with decreased weighting as evidence to improve the performance of question-answering systems. The method multiplies the score of the i-th candidate by $k^{(i-1)}$ before adding the score to the running total. We experimentally found that 0.2 and 0.3 were good values for k. Our method is simple, and it produced large score improvements. These results demonstrate the feasibility and utility of our method.

Our team (CRL/NICT)[2] obtained the second-best precision, the best precision, and the second-best precision in Task-1, Task-2, and Task-3 of QAC1, respectively. It obtained the second-best score, the best score, and the best score in Subtask-1, Subtask-2, and Subtask-3 of QAC2, respectively. It also obtained the best score in QAC3. The results of the QAC contest series indicate the effectiveness of our question-answering system.

Our question-answering system has yet to use a large ontology for the named entity and has used only a few kinds of named entities. In future studies, we would like to use more kinds of named entities to improve the performance of our system.

[1] The experiments on the comparison of use and no use of decreased weighting were done in NTCIR 4 to confirm the effectiveness of the method and the results were described in our previous papers [12, 13].

[2] CRL is an abbreviation of Communications Research Laboratory, which is the previous name of our institute, the National Institute of Information and Communications Technology (NICT).

Acknowledgements

We are grateful to all of the organizers of NTCIR 5, who gave us a chance to participate in the NTCIR 5 contest to improve and examine our question-answering system. We greatly appreciate the kindness of all those who helped us.

References

[1] C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[2] S. Dumis, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[3] A. Ittycheriah, M. Franz, W.-J. Zhu, and A. Ratnaparkhi. IBM's Statistical Question Answering System. In *TREC-9 Proceedings*, 2001.

[4] J. Kupiec. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[5] B. Magnini, M. Negri, R. Prevete, and H. Tanev. Is it the right answer? Exploiting web redundancy for answer validation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2002.

[6] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, and M. Asahara. Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition. 1999.

[7] D. Moldovan, M. Pasca, and M. S. Sanda Harabagiu. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, 21(2):133–154, 2003.

[8] M. Murata, Q. Ma, and H. Isahara. High performance information retrieval using many characteristics and many techniques. *Proceedings of the Third NTCIR Workshop (CLIR)*, 2002.

[9] M. Murata, K. Uchimoto, H. Ozaku, Q. Ma, M. Utiyama, and H. Isahara. Japanese probabilistic information retrieval using location and category information. *The Fifth International Workshop on Information Retrieval with Asian Languages*, pages 81–88, 2000.

[10] M. Murata, M. Utiyama, and H. Isahara. Question answering system using syntactic information. 1999. http://xxx.lanl.gov/abs/cs.CL/9911006.

[11] M. Murata, M. Utiyama, and H. Isahara. A question-answering system using unit estimation and probabilistic near-terms IR. *Proceedings of the Third NTCIR Workshop (QAC)*, 2002.

[12] M. Murata, M. Utiyama, and H. Isahara. Japanese question-answering system using decreased adding with multiple answers. *Proceedings of the NTCIR Workshop 4 (QAC)*, 2004.

[13] M. Murata, M. Utiyama, and H. Isahara. Use of multiple documents as evidence with decreased adding in a Japanese question-answering system. *Journal of Natural Language Processing*, 12(2), 2005.

[14] M. Murata, M. Utiyama, Q. Ma, H. Ozaku, and H. Isahara. CRL at NTCIR2. *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization*, pages 5–21–5–31, 2001.

[15] National Institute of Informatics. *Proceedings of the Third NTCIR Workshop (QAC)*. 2002.

[16] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[17] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC-3*, 1994.

[18] T. Takaki and Y. Eriguchi. NTT DATA question-answering experiment at the NTCIR-3 QAC. *Proceedings of the Third NTCIR Workshop (QAC)*, 2002.

[19] TREC-10 committee. The tenth text retrieval conference. 2001. http://trec.nist.gov/pubs/trec10/t10_proceedings.html.

[20] H. Yamada, T. Kudo, and Y. Matsumoto. Japanese named entity extraction using support vector machine. *Transactions of Information Processing Society of Japan*, 43(1):44–53, 2002.