Japanese Question-Answering System Using Decreased Adding with Multiple Answers

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}@crl.go.jp

Abstract

We propose a new method of using multiple documents as evidence with decreased adding 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 carried out experiments using the QAC1 test collection and confirmed the effectiveness of our method through a statistical test. Our method produced a large improvement, with values of 0.05 to 0.14 for the evaluation scores (MRR/MF). We also obtained relatively good results in the experiments using the QAC2 test collection. These results, and the fact that our method is very simple and easy to use, demonstrate its feasibility and utility for question-answering systems.

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 a document containing a sentence, 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 an online text, such as a website, a newspaper article, or an encyclopedia. The system can then output "Tokyo" as the correct answer. We expect question-answering systems to become increasingly important as a more convenient alternative to 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]. Evaluated conferences, or contests, on question-answering systems have been held in both the U. S. A. and Japan. In the U. S. A., an evaluated conference has been held as the Text REtrieval Conference (TREC) [17], while in Japan, a conference called the Question-Answering Challenge (QAC) has been conducted [13]. These evaluated conferences aim to improve question-answering systems. Researchers make their question-answering systems and use them to solve the same questions, and each system's performance is then examined to glean possible improvement. We have investigated the potential of question-answering systems [10] and studied their construction by participating in the QAC [13] at NTCIR 3 [11].

In this paper, we propose a new method using multiple documents as evidence with decreased adding to improve the performance of question-answering systems. 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 for question answering systems [1, 2, 5, 16]. In our method, information from multiple documents is employed by adding the scores for the candidate answers extracted from the various documents [2, 16]. 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 adding. 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

Table 1. Candidate answers with 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 with 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
...	...	...	...

"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.

To evaluate this method, we experimented using the QAC test collection [13] and confirmed the effectiveness of our method through a statistical test. In our experiments, we constructed a question-answering system and employed four variations of it to confirm that our method was effective. We also confirmed that simply adding the scores from multiple documents, without employing decreasing weights, degraded the performance of the question-answering systems in some cases. Our method is very simple and easy to use, and it improves the performance of these systems, thus demonstrating its feasibility and utility.

2 Use of Multiple Documents as Evidence with Decreased Adding

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 method based on simply adding the scores of candidate answers was used previously [2, 16]. For our current example question, this produces the results shown in Table 2. In this case, the system outputs the correct answer, "Tokyo", as the first answer. The method can thus 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. It is a serious problem from a performance standpoint, in particular. In the case of a system with good inherent performance, the original scores that it outputs are often more reliable than the simply added scores, so the use of this method often degrades the system performance.

To overcome this problem, we developed our new method of using multiple documents as evidence with decreased adding. Instead of simply adding the scores of the candidate answers, the method adds the scores with decreasing weights. This approach reduces the negative effect of a question-answering 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 proposed 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 apply the method of simply adding scores in this system, however, we obtain the results shown in Table 4. In this case, the incorrect answer, "Tokyo",

Table 5. Candidate answers obtained by decreased adding, 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 decreased adding, where when "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
...	...	...	...

achieves the highest score.

To overcome this problem, we can try to apply our proposed method of adding candidate scores with decreasing weights. Suppose that we implement our method by multipling the score of the i-th candidate by a factor of $0.3^{(i-1)}$ before adding 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.

As described here, our method of adding scores for candidate answers with decreasing weights successfully obtained the correct answers to each of the example questions. This suggests the feasibility of the method 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. We thus confirmed the effectiveness of our method experimentally, as described in Section 4.

3 Question-answering Systems Used in 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 we used 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 document retrieval method operates 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 [14, 15]. This approach is very effective, and we have used it extensively for information retrieval [9, 12, 8]. In the question answering system, we use 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.

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

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

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 question-answering system can determine whether terms occur 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 only use candidates consisting only of nouns, unknown words, and symbols. Also, 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 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.

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

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

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 used 45 heuristic rules to award points to candidates and utilized the total points as the score. Some of the heuristic rules are listed below:

1. Add 1000 to 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 [18]. We used only five named entity as same as in our previous system [11].

2. When a country name is one of the predicted answer types, add 1000 to 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 candidates having the Noun X.

Our system has an additional function that are used after answers are selected based on the scores. It is the compiling of similar answers. Our system compiles answers that are part of other answers and the difference in their scores is less than 90% of the best score. The compiling is done by eliminating answers other than the longest one. We call this method *rate-based answer compiling*.

4 Experiments using the QAC1 data collection

4.1 Data Used in the Experiments

We used the Task-1 data in the QAC1 data collection from the QAC at NTCIR 3 for our first experiments [13]. This data collection contains 200 pairs consisting of a question and its answer. It was developed in the Japanese language and is based on articles in the Japanese Mainichi newspaper from 1998 and 1999. The mean reciprocal rank (MRR) is used for evaluation. By applying the MRR, we then obtain a score of $1/r$ when the r-th output answer is correct. Our system outputs its top five answers.

4.2 Methods of Adding Scores

In the experiments, we employed each of the following methods of adding scores.

Original Method

This method simply outputs the answers generated by the question-answering system as they are, without adding the scores of each candidate answer.

Simple Adding Method

This method adds the scores of each candidate answer as extracted from multiple documents. It then outputs the answers according to their added scores.

In our question-answering system, the significance of a candidate answer is greatly changed by a score of 1000. Therefore, we do not want to skew scores by adding the thousands and higher-order digits from each score. Instead, for scores with the same values for the thousands and higher-order digits, the method extracts only the digits representing values below 1000 (i.e., hundreds, tens, and units) from each score. It then adds all the extracted values to give a subtotal, which is combined with the values of the thousands and higher digits shared by the scores.

On the other hand, the method does not add the scores of candidates with different values for the digits representing values of 1000 or greater, but instead simply takes the higher score as the total.

For example, suppose that a candidate answer X appears twice with scores of 1025 and 1016. In this case, 25 and 16 are extracted as the digits representing values below 1000; these scores are added to obtain 41; and finally, 41 is added to 1000, and 1041 is obtained as the total score. As another example, suppose that a candidate answer X appears twice with scores of 2025 and 1016. Here, the scores have different values for the thousands digit. In this case, the scores are not added, and 2025 is obtained as the total score.

Decreased Adding Method

This method adds the scores for each candidate answer extracted from multiple documents in almost the same way as the simple adding method. It also handles the digits representing values of 1000 or more by using the same approach.

The actual method of adding, however, is different. The decreased adding method multiplies the score of the i-th candidate answer by a factor of $k^{(i-1)}$ before adding the scores. This is expressed by the following equation:

$$Score_{decreased} = \sum_{1 \leq i \leq n} k^{i-1}\ score_{original}(i) \qquad (5)$$

Here, $Score_{decreased}$ is the value of the final score obtained by the decreased adding method for the digits representing values below 1000, while $score_{original}(i)$ is the value of the original score obtained by the question-answering system for these digits. n is the number of occurrences of the same candidate answer extracted from multiple documents with the same values for the thousands and higher-order digits. Finally, k is a constant set according to experimental results.

For example, suppose that a candidate answer X appears twice with scores of 2025 and 2016, and that $k = 0.3$. First, 25 and 16 are extracted as the values for the digits below 1000. These values are added by using Eq. 5, so that $25 + 16 \times 0.3$ gives a result of 29.8. Then, 29.8 is added to 2000, and 2029.8 is obtained as the total score.

In this study, we used 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9 as values of k.

Combined Method

This method is a combination of the original method, the simple adding method, and the decreased adding method (with the same twelve possible values of k, i.e., 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9). First, the combined method identifies the method obtaining the best performance (as measured by the MRR/MF) for a set of training data. Then, it uses the best method to output answers.

In this study, we did not use any data other than the QAC test collection, so we performed 10-fold cross validation for training purposes.

The combined method involves two important considerations.

One is the combination of multiple methods. It can select a good method for each case and improve system performance.

The second consideration is fair evaluation. For example, the decreased adding method had twelve possible values of k in our experiments. Because the variation in k is large, even if a system using a certain value for k obtains a good evaluation score, the score may be a rare fluke found only in the test data. In general, to avoid such unfair evaluation and to calculate appropriate evaluation scores, the 10-fold cross validation is used.

4.3 Tested Systems

As noted above, we used the question-answering system described in Section 3 for the experiments, but we employed four variations of it, as listed here. Note that the designation, Sys-3, refers to the base system exactly as described previously.

We used these four systems to confirm whether our proposed method of using multiple documents as evidence with decreased adding would be effective in various question-answering systems.

Sys-1

Unlike Sys-3, which uses Eq. 4, this system uses the following equation.

$$Score_{near2}(c) = \sum_{t2 \in T} w_{dr2}(t2) log \frac{N}{df(t2)} \quad (6)$$

Sys-1 thus does not use the distances between a candidate answer and the terms extracted from the input question.

Sys-2

This system divides documents into paragraphs during document retrieval without re-ranking based on Eq. 2, and it also uses Eq. 6 instead of Eq. 4.

Sys-2 thus also does not directly use the distances between a candidate answer and the extracted terms. Because it divides documents into paragraphs during document retrieval, however, it does confirm whether a candidate answer appears in the same paragraph as each term. This enables Sys-2 to utilize a little more information related to the distances between the candidate answer and the extracted terms, as compared to Sys-1.

Sys-3

Sys-3 is the base question-answering system as described in Section 3.

4.4 Experimental Results

We conducted experiments with the QAC test collection for Task-1 by using the methods described in Sections 4.2 and 4.3. The results are shown in Table 7. In the table, the leftmost column indicates the adding method, while the top line indicates the question-answering system. We used the two-sided t-test as a statistical test to recognize significant differences, with the original method as the baseline method. When a method performed better than the baseline method at the 0.05 or 0.01 significance level, it was tagged with "+" or "++", respectively. Likewise, when a method performed worse than the baseline method at the 0.05 or 0.01 significance level, it was tagged with "-" or "–", respectively.

Table 7. Results for Task-1 in QAC1 (MRR)

	Sys-1	Sys-2	Sys-3
Original	0.294	0.405	0.541
Simple	0.387^{++}	0.474^{+}	0.449^{--}
Combined	0.437^{++}	0.506^{++}	0.597^{++}
Decreased			
k=0.01	0.432^{++}	0.498^{++}	0.551
k=0.02	0.433^{++}	0.502^{++}	0.561
k=0.05	0.440^{++}	0.510^{++}	0.563
k=0.1	0.449^{++}	0.516^{++}	0.570
k=0.2	0.446^{++}	0.509^{++}	0.590^{++}
k=0.3	0.450^{++}	0.504^{++}	0.597^{++}
k=0.4	0.434^{++}	0.504^{++}	0.580
k=0.5	0.428^{++}	0.509^{++}	0.565
k=0.6	0.414^{++}	0.505^{++}	0.544
k=0.7	0.411^{++}	0.498^{++}	0.537
k=0.8	0.399^{++}	0.489^{++}	0.492
k=0.9	0.390^{++}	0.480^{++}	0.472^{-}

4.5 Discussion

From Table 7, we found the following: The simple adding method obtained lower performance than the original method in some cases (Sys-3). The combined method, which includes our proposed method, always obtained higher performance than the original method and the simple adding method. Our method produced a large improvement, with values of 0.05 to 0.14 for the evaluation scores (MRR). In the decreased adding method, 0.2 and 0.3 were good values for k. We actually counted the frequency of using each method in the combined method and confirmed that the frequency of using 0.2 or 0.3 was very large. With our best question-answering system (Sys-3), the combined method obtained scores of 0.597 for Task-1, while the original method obtained scores of 0.541. The best score was 0.608 in the QAC contest. We have tried many other ways to improve question answering, but we could not improve question answering easily. Although our proposed method in this paper is very easy and feasible, it could make quite a large improvement and obtain almost the same precision as the best score in the QAC contest.

The proposed method using multiple documents as evidence with decreased adding has the problem that when only one document includes answers, the method cannot add the scores of answers in multiple documents, and the performance of the system deteriorates. We examined this problem by using Sys-3, which offered the best performance. We calculated the system performance of Sys-3 with various numbers of documents (x) including answers among the top 20 documents obtained during document retrieval. As we expected, in the case of $0<x\leq1$, the combined method obtained lower performance than the original method.

Table 8. Results for Subtask-1 in QAC2

System ID	MRR
CRL1	0.566
CRL2	0.577
Base line	0.541

Table 9. Results for Subtask-2 in QAC2

System ID	MF	Rate for select
CRL1	0.321	0.95
CRL2	0.302	0.97
CRLX	0.363	0.90
CRLY	0.358	0.85
CRLZ	0.334	0.80
Base line	0.293	0.95

Otherwise, however, the combined method obtained higher performance. To solve this problem, we should increase the size of the document sets to obtain multiple documents containing answers or identify whether the number of documents including answers for an input question is one or more.

5 Experiments using the QAC2 data collection

In this section, we show the experimental results in the QAC2 data collection. The results are shown in Tables 8 to 10. We used Sys-3 with the decreased adding method and $k = 0.3$. In Subtask-1, our system outputs its top five answers. In CRL1 of Subtask-2 or Subtask-3, our system outputs the answers having a score that is more than 95% of the highest score. In CRL2 of Subtask-2 or Subtask-3, our system outputs the answers having a score that is more than 97% of the highest score. In CRL3 of Subtask-3, our system outputs the top five answers. Although we used *select-one method* in the QAC1 contest [11], we used *select-by-rate method* in the QAC2 contest. *Select-by-rate method* outputs the answers having a score more than a certain rate (Rate for selection) of the highest score. In Subtask-2, we made experiments of changing the rate for selection. CRLX, CRLY, and CRLZ are the results after the contest. In Subtask-3, the automatic evaluation system has not been given by the organizers, so we could not make additional experiments. "Base line" is the system using Sys-3 and not using the decreased adding method. Sys-2 of Subtask-1 uses the following equation [11] instead of Equation 4.[1]

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

Table 10. Results for Subtask-3 in QAC2

System ID	MF
CRL1	0.224
CRL2	0.223
CRL3	0.153

$$Score_{near2}(c) = \sum_{t2 \in T3} k_{near2}^{dist(c,t2)} w_{dr2}(t2) log \frac{N}{df(t2)} \quad (7)$$

$$T3 = \{t | t \in T, 2dist(t1,t) \frac{df(t)}{N} \leq 1\} \quad (8)$$

Our systems obtained the second-best score, the best score, and the best score, in Subtask-1, Subtask-2, and Subtask-3 of QAC2, respectively. This indicates that our methods for question-answering systems are very effective.

In Subtask-1 and Subtask-2, CRL1 which uses the decreased adding method outperformed the baseline method which does not use the decreased adding method. The effectiveness of the decreased adding method was also confirmed in the QAC2 data collection.

In this paper, we could not show the results of more detailed experiments in the case using the QAC2 data collection, because the schedule for writing is very tight and our system needs a lot of time. (Our system is very slow.) In the future studies, we plan to make more experiments and show them.

6 Conclusions

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

Our participation team (CRL) 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 also obtained the second-best score, the best score, and the best score, in Subtask-1, Subtask-2,

and Subtask-3 of QAC2, respectively. Thus, we obtained very good results constantly in the series of the QAC contest. This indicates the effectiveness of our question-answering system. Our question-answering system uses many kinds of effective methods that can be used easily. Thus, the paper describing those methods will be very useful.

Our question-answering system has not yet used 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.

Acknowledgement

We are grateful to all of the organizers of NTCIR 4, who gave us a chance to pariticpate in the NTCIR 4 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, 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.

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

[14] 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.

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

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

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

[18] 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.