Japanese-to-English and English-to-Japanese Cross-Language 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 describe a method of using multiple documents with decreasing weights as evidence to improve the performance of a question-answering system and how it was employed in cross-language question answering (CLQA) tasks at NTCIR-5. 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 in the CLQA part of NTCIR-5. It was incorporated in a commercially available translation system that performed the cross-language question-answering tasks. Our method obtained relatively good CLQA results.

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].

At NTCIR-4, we proposed 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 "34.1" (= 26 + 21 $\times$ 0.3 + 20 $\times$ 0.3^2). Thus, we

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
...	...	...	...

calculate the score in the same way for each candidate and take the answer with the highest score as the correct answerWhen this method was used at CLQA (NTCIR-5), it obtained relatively high scores among those of the participants.

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", 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"

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
...	...	...	...

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.

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

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

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.

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

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

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 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.

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

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

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 listedbelow:

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 How we handle cross-language question-answering

We used commercially available translation software to translate the questions and documents. Our monolingual question answering system can only handle the Japanese language. Therefore, we translated the questions into Japanese, to perform the English-to-Japanese question-answering, tasks and translated the documents into English, to perform the Japanese-to-English tasks.[1]. We also used the same translation system to translate the answers.[2]

5 Experiments

In this section, we show the experimental results in CLQA of NTCIR-5. Tables 7 to 11 show the results of CLQA at NTCIR-5. We submitted one official run (NICT-E-J-01) and four unofficial runs (NICT-E-J-u-01, NICT-E-J-u-02, NICT-J-E-u-01, and NICT-J-E-u-02). After the formal run, we made additional two runs (NICT-J-J--01, NICT-J-J--02). We used the decreasing weights method with $k = 0.3$ in NICT-E-J-01, NICT-E-J-u-01, NICT-J-E-u-01, and NICT-J-J- $\times$ -01. We did not use it in NICT-E-J-u-02, NICT-J-E-02, and NICT-J-J--02. In the tables, "top 1" in the left most column indicates that only one answer was evaluated for each question, while "5 ans." indicates that five answers were evaluated for each question. Regarding "5 ans.", we used the top five answers. "Acc", "MRR", and "Top5" are evaluation metrics. "Acc" indicates the accuracy rate of the first answer. "MRR" indicates a score of $1/r$ when the r-th submitted answer is correct. "Top5" indicates the ratio when one of the top five answers was correct. "*+U" indicates answers that were not supported by a relevant document were judged to be correct. No "*+U" indicates only the answers that were supported were judged to be correct. Tables 7 and 8 show the results for the English-to-Japanese question answering tasks, and Tables 9 and 10 show the results for the Japanese-to-English tasks. Table 11 shows the result for the Japanese-to-Japanese task.[3] Tables 7 10 and 11 show the evaluations of the Japanese answers, and Tables 8 and 9 show the evaluations of the English answers.

The following findings are indicated by the experimental results.

- The decreasing weights were effective (compare "NICT-J-E-u-01" and "NICT-J-E-u-02", and "NICT-E-J-u-01" and "NICT-E-J-u-02").
- The Japanese-to-English question-answering tasks were more difficult for our methods than the English-to-Japanese tasks were (compare "NICT-J-E-u-01" and "NICT-E-J-u-01", and "NICT-J-E-u-02" and "NICT-E-J-u-02").
- When the translated answers were used, the evaluation scores were very low, as shown in Tables 8 and 10.
- The Japanese monolingual question-answering tasks were easier than the Japanese-to-English or English-to-Japanese question-answering tasks (compare "NICT-J-E-u-01", "NICT-E-J-u-01", and "NICT-J-J--01", and "NICT-J-E-u-02", "NICT-E-J-u-02", and "NICT-J-J--02").

[1] In the English-to-Japanese tasks, the questions were written in English and the documents were written in Japanese. In the Japanese-to-English tasks, the questions were written in Japanese and the documents were written in English

[2] For example, to output English answers in the English-to-Japanese tasks, we translated the Japanese answers extracted from the Japanese documents into English, and to output English answers in the Japanese-to-English tasks, we translated the Japanese answers extracted from the Japanese documents into English. We used Japanese documents in the Japanese-to-English tasks because we only used a Japanese question-answering system.)

[3] The Japanese-to-Japanese task is not considered in NICIR-5.

6 Conclusions

We described a new method of using multiple documents with decreasing weights as evidence 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 used this method for the CLQA part of NTCIR-5. We incorporated it into a commercially available translation system that performed the cross-language question-answering tasks. Our method obtained relatively good results at CLQA.

Acknowledgement

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

Table 7. Evaluation of Japanese answers in the English-to-Japanese question-answering tasks

System ID	Acc	MRR	Top5	Acc+U	MRR+U	Top5+U
NICT-E-J-01 (top 1)	0.090	0.090	0.090	0.120	0.120	0.120
NICT-E-J-u-01 (top 1)	0.090	0.090	0.090	0.120	0.120	0.120
NICT-E-J-u-01 (5 ans.)	0.090	0.095	0.105	0.120	0.155	0.210
NICT-E-J-u-02 (top 1)	0.075	0.075	0.075	0.100	0.100	0.100
NICT-E-J-u-02 (5 ans.)	0.075	0.086	0.105	0.100	0.128	0.175

Table 8. Evaluation of English answers in the English-to-Japanese question-answering tasks

System ID	Acc	MRR	Top5	Acc+U	MRR+U	Top5+U
NICT-E-J-u-01 (top 1)	0.000	0.000	0.000	0.040	0.040	0.040
NICT-E-J-u-01 (5 ans.)	0.000	0.000	0.000	0.040	0.052	0.070
NICT-E-J-u-02 (top 1)	0.000	0.000	0.000	0.035	0.035	0.035
NICT-E-J-u-02 (5 ans.)	0.000	0.000	0.000	0.035	0.042	0.055

Table 9. Evaluation of English answers in the Japanese-to-English question-answering tasks

System ID	Acc	MRR	Top5	Acc+U	MRR+U	Top5+U
NICT-J-E-u-01 (top 1)	0.010	0.010	0.010	0.030	0.030	0.030
NICT-J-E-u-01 (5 ans.)	0.010	0.016	0.025	0.030	0.041	0.065
NICT-J-E-u-02 (top 1)	0.025	0.025	0.025	0.030	0.030	0.030
NICT-J-E-u-02 (5 ans.)	0.025	0.029	0.035	0.030	0.037	0.050

Table 10. Evaluation of Japanese answers in the Japanese-to-English question-answering tasks

System ID	Acc	MRR	Top5	Acc+U	MRR+U	Top5+U
NICT-J-E-u-01 (top 1)	0.000	0.000	0.000	0.020	0.020	0.020
NICT-J-E-u-01 (5 ans.)	0.000	0.000	0.000	0.020	0.031	0.050
NICT-J-E-u-02 (top 1)	0.000	0.000	0.000	0.010	0.010	0.010
NICT-J-E-u-02 (5 ans.)	0.000	0.000	0.000	0.010	0.019	0.035

Table 11. Evaluation of Japanese monolingual question-answering tasks

System ID	Acc	MRR	Top5	Acc+U	MRR+U	Top5+U
NICT-J-J--01 (top 1)	0.170	0.170	0.170	0.265	0.265	0.265
NICT-J-J--01 (5 ans.)	0.170	0.239	0.370	0.265	0.386	0.605
NICT-J-J--02 (top 1)	0.190	0.190	0.190	0.240	0.240	0.240
NICT-J-J--02 (5 ans.)	0.190	0.261	0.380	0.240	0.362	0.565

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.