English-to-Japanese Cross-Language Question-Answering System using Weighted Adding with Multiple Answers

Masaki Murata, Masao Utiyama, Toshiyuki Kanamaru, and Hitoshi Isahara

Abstract—**We describe a method of using multiple documents with decreasing weights as evidence to improve the performance of a question-answering system. We also describe how it was used in cross-language question answering (CLQA) tasks. Sometimes, the answer to a question may be found in multiple documents. In such cases, using multiple documents for prediction generates better answers than using a single document. Therefore, our method uses information from multiple documents by adding the scores of 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 simply adding. We used this method in the CLQA part of NTCIR-5. It was incorporated into a commercially available translation system that carries out cross-language question-answering tasks. Our method obtained relatively good CLQA results.**

Index Terms—**Machine translation, cross-language question-answering, decreased adding, multiple documents, NTCIR.**

I. 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 [1], [2], [3], [4], [5], [6]. Evaluation conferences or contests on question-answering systems have been held in both the U. S. A. and Japan. In the U. S. A., one evaluation conference was called the Text REtrieval Conference (TREC) [7], while in Japan, another conference was called the Question-Answering Challenge (QAC) [8]. These evaluation conferences aim to improve question-answering systems by having researchers use their question-answering systems to solve the same questions, and then examining each system's performance to glean possible methods of improvement. We investigated the potential of question-answering systems [9] and studied their construction by participating in the QAC [8] at NTCIR workshop [10].

Manuscript received October 19, 2008. Manuscript accepted for publication August 3, 2009.

M. Murata, M. Utiyama, T. Kanamaru, and H. Isahara are with the National Institute of Information and Communications Technology 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan (e-mail: murata@nict.go.jp).

We proposed a new method that uses multiple documents as evidence but decreases adding to improve performance. Sometimes, the answer to a question may be found in multiple documents. In such cases, question answering systems that use multiple documents for prediction generate better answers than those that use only one document [3], [4], [5], [11]. In our method, information from multiple documents is used by adding the scores for the candidate answers extracted from the various documents [4], [11]. 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 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. When this method was used at CLQA (NTCIR-5), it scored higher than most participants' methods.

II. 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 I. These systems also output a document ID indicating the document from which each candidate answer was extracted.

TABLE I
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 II
CANDIDATE ANSWERS CHOSEN BY SIMPLE ADDITION 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 III
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 IV
CANDIDATE ANSWERS CHOSEN SIMPLY BY ADDING 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
...	...	...	...

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

A method based on simple addition of the scores of candidate answers was used previously [4], [11]. For our current example question, this produces the results shown in Table II. In this case, the system outputs the correct answer, "Tokyo", as the first answer. The method can thus obtain correct answers 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. In the case of a system with good inherent performance, the original scores that it outputs are often more reliable than the simple addition scores, so using this method often degrades system performance.

To overcome this problem, we developed our new method of using multiple documents with decreased adding as evidence. Instead of simply adding the scores of the candidate answers, the method adds the scores with decreasing weights. This approach reduces the likelihood that a question-answering system will 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 with an example. Suppose that a question-answering system outputs Table III 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 use a method that simply adds scores in this system, however, we obtain the results shown in Table IV. In this case, the incorrect answer, "Tokyo", scores the highest.

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 multiplying 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 V. 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 VI. As expected, "Tokyo" scores the highest.

As shown 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 that the method reduces the likelihood that a question-answering system will select candidate answers with high frequencies, while at the same time improving the system's accuracy.

III. QUESTION-ANSWERING SYSTEMS USED IN THIS STUDY

The system has 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 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

TABLE V
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 VI
CANDIDATE ANSWERS OBTAINED BY DECREASED ADDING, 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
...	...	...	...

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 people's names as candidate answers from documents containing the terms "prime", "minister", and "Japan".

A. Prediction of Answer Type

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 *doko* "where" is in a question sentence and the focus word in a question is not *chiiki* (area), *basho* (location), or so on, an organizational expression is given as the answer type.
4) When *doko* "where" is in a question sentence and the focus word in a question is not *kaisha* (company), *soshiki* (organization), or so on, a location expression is given as the answer type.
5) When *doko no kuni* "what country" is in a question sentence, a country expression is given as the answer type.
6) When '*nani* (what) + suffix' is in a question sentence, the suffix is extracted as a unit expression.
7) When *donokurai* "how many" occurs in a question, a numerical expression is given as the answer type. The unit expression is estimated using the following method.

Our system uses a new method, which we call *unit estimation*, to obtain a correct unit expression answer. With this method, we gather sentences containing expressions like "UNIT-FOCUS + *wa* (be) + 'numerical expressions' + 'unit expressions"' and extract the unit expressions. We then eliminate unnecessary unit expressions by applying a statistical test based on a binomial distribution. Eliminated expressions are as follows:

$$\text{Unnecessary expressions} = \{t | P(e) \leq k_p\}, \quad (1)$$

where $P(e)$ is calculated by the following equation and k_p is a constant identified based on experimental results.

$$P(e) = \sum_{r=0}^{k} C(n,r) p(u)^r (1 - p(u))^{n-r}, \quad (2)$$

where $C(x, y)$ is the number of combinations when we select y items from x items, n is the number of times expression e occurs in the corpus, k is the number of times the unit expression e occurs in the pattern of "UNIT-FOCUS + *wa* (be) + 'numerical expressions' + 'unit expressions"' in the corpus, and $p(u)$ is calculated by

$$p(u) = \frac{\text{freq}(u)}{N}, \quad (3)$$

where $\text{freq}(u)$ is the frequency of the UNIT-FOCUS u appearing in the corpus and N is the number of all characters in the corpus. In this study, we used articles from newspapers issued over a 10-year period [12] as the corpus for the unit estimation calculation.

TABLE VII
AN EXAMPLE OF USING UNIT ESTIMATION

e	k	n	P(e)
meetoru (meter)	50	128175	1.000000
senchi (centimeter)	28	47050	1.000000
miri (millimeter)	11	25897	1.000000
kiro (kilometer)	11	99618	0.999996
kounen (light-year)	2	538	1.000000
hun (minute)	2	955808	0.000000
yaado (yard)	1	2744	0.998205
inchi (inch)	1	1865	0.999160
hon (piece)	1	1625073	0.000000
shaku (*shaku*)	1	2146	0.998892

An example of using unit estimation is as follows. Consider the question sentence 'X *no nagasa wa dono kurai desuka?*' (What is the length of X?). In this case, we extract a noun *nagasa* (length) as the UNIT-FOCUS and gather candidate unit expressions using "*nagasa* + *wa* + 'numerical expressions' + 'unit expressions'". We obtain *meetoru* (meter), *senchi* (centimeter), *miri* (millimeter), *kiro* (kilometer), *kounen* (light-year), *hun* (minute), *yaado* (yard), *inchi* (inch), *hon* (piece), and *shaku* (a measure unit for a length, equal to about 30.3 cm) as candidates. We calculate $P(e)$ for each candidate and obtain the results shown in Table VII. In this case, N is 533,366,720, the frequency of *nagasa* is 11,887, and $p(u) = \frac{11{,}887}{533{,}366{,}720} = 0.000022289$. As shown in Table VII, our method can correctly eliminate *hun* (minute) and *hon* (piece). When using our unit estimation, we do not need a dictionary for unit expressions. Another valuable feature of unit estimation is that it presents various expressions that appear in the corpus. Unit estimation can also be used to construct a dictionary of unit expressions. Thus, our unit estimation method offers various benefits.

B. Document Retrieval

Our system extracts terms from a question using a morphological analyzer called ChaSen [13]. The analyzer first eliminates terms that are prepositions or similar parts of speech and then retrieves 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

$$\begin{aligned} &Score(d) \\ &= \sum_{\substack{\text{term}\\ t}} \left(\frac{tf(d,t)}{tf(d,t) + k_t \dfrac{length(d)+k_+}{\Delta + k_+}} \times \ log \frac{N}{df(t)} \right) \end{aligned} \tag{4}$$

where d is a document, t is a term extracted from a question, $tf(d,t)$ is the occurrence frequency of t 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 identified based on 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 [16], [17], [18]. 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.

$$\begin{aligned} &Score(d) \\ = \quad &-min_{t1 \in T} log \prod_{t2 \in T3} (2dist(t1,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \\ = \quad &max_{t1 \in T} \sum_{t2 \in T3} w_{dr2}(t2) log \frac{N}{2dist(t1,t2) * df(t2)} \end{aligned} \tag{5}$$

$$T3 = \{t | t \in T, 2dist(t1,t)\frac{df(t)}{N} \leq 1\}, \tag{6}$$

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 based on experimental results.

Because our question-answering system can determine whether terms occur near each other by re-ranking them according to Eq. 5, 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.

C. Answer Detection

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

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 with the highest total points as correct answers.

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

$$\begin{aligned} Score_{near}(c) &= -log \prod_{t2 \in T3} (2dist(c,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \\ &= \sum_{t2 \in T3} w_{dr2}(t2) log \frac{N}{2dist(c,t2) * df(t2)} \end{aligned} \tag{7}$$

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

where c is a candidate for the correct answer, and $w_{dr2}(t2)$ is a function of $t2$ that is adjusted based on 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 used 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 [19]. We used only five named entities as in our previous system [10].

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 is used after answers are selected based on the scores. Our system compiles answers that are part of other answers and whose score is less than 90% of the best score. The system compiles answers by retaining the longest one and eliminating the others. We call this method *rate-based answer compiling*.

IV. How We Handle Cross-Language Question-Answering

We used commercially available translation software to translate questions and documents. We translated the questions into Japanese, to carry out the English-to-Japanese question-answering tasks. In the English-to-Japanese tasks, the questions were written in English and the documents were written in Japanese. We output Japanese answers in response to English queries.

V. Experiments

In this section, we show the experimental results for CLQA of NTCIR-5. Tables VIII to IX show these results. We did one official run (NICT-E-J-01) and two unofficial runs (NICT-E-J-u-01, NICT-E-J-u-02). After the formal run, we made two additional 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, and NICT-J-J--01. We did not use it in NICT-E-J-u-02, and NICT-J-J--02. 200 questions were given for each run. In the tables, "top 1" in the leftmost column indicates that only one answer was evaluated for each question, while "5 ans." indicates that five answers were evaluated for each question, in which case 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 and were judged to be correct. No "*+U" indicates only the answers that were supported and were judged to be correct. Tables VIII shows the results for the English-to-Japanese question answering tasks. Table IX shows the results for the Japanese-to-Japanese task. The Japanese-to-Japanese task is not relevant to NTCIR-5. We did the experiments with NTCIR-5 to compare the results for English-to-Japanese and for Japanese-to-Japanese tasks.

The experimental results indicate the following.

– The method of weighted adding was effective (compare "NICT-E-J-u-01" and "NICT-E-J-u-02", or "NICT-J-J-u-01" and "NICT-J-J-u-02").). In every case, the accuracy of the method that uses weighted adding was higher than that of methods that do not use weighted adding.
– The Japanese monolingual question-answering tasks were easier than the Japanese-to-English or English-to-Japanese question-answering tasks (compare "NICT-E-J-u-01" and "NICT-J-J--01", and "NICT-E-J-u-02", and "NICT-J-J--02").
– Our cross-language (English-to-Japanese) question-answering obtained about half the accuracy of single-language (Japanese-to-Japanese) question-answering (0.09/0.170 or 0.120/0.265). We found that use of commercial translation software answering in cross-language question answering obtained about half the accuracy of single-language question.

VI. Conclusion

We described a new method of using decreasingly weighted multiple documents 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 0.3 were good values for k. Our proposed method is simple and easy to use, and scored much better than methods that did not use decreased adding. These results demonstrate the effectiveness 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 carries out cross-language question-answering tasks. Our method obtained relatively good results at CLQA.

References

[1] 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.
[2] A. Ittycheriah, M. Franz, W.-J. Zhu, and A. Ratnaparkhi, "IBM's Statistical Question Answering System," in *TREC-9 Proceedings*, 2001.
[3] 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.
[4] 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.
[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] D. Moldovan, M. Pasca, S. Harabagiu, and M. Surdeanu, "Performance issues and error analysis in an open-domain question answering system," *ACM Transactions on Information Systems*, vol. 21, no. 2, pp. 133–154, 2003.
[7] TREC-10 committee, "The tenth text retrieval conference," 2001, http://trec.nist.gov/ pubs/trec10/t10_proceedings.html.
[8] National Institute of Informatics, *Proceedings of the Third NTCIR Workshop (QAC)*, 2002.

TABLE VIII
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 IX
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

[9] M. Murata, M. Utiyama, and H. Isahara, “Question answering system using syntactic information,” 1999, http://xxx.lanl.gov/abs/cs.CL/9911006.
[10] ——, “A question-answering system using unit estimation and probabilistic near-terms IR,” 2002.
[11] T. Takaki and Y. Eriguchi, “NTT DATA question-answering experiment at the NTCIR-3 QAC,” 2002.
[12] Mainichi Publishing, “Mainichi Newspaper 1991-2000,” 2000.
[13] 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.
[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] M. Murata, K. Uchimoto, H. Ozaku, Q. Ma, M. Utiyama, and H. Isahara, “Japanese probabilistic information retrieval using location and category information,” 2000, pp. 81–88.
[17] M. Murata, M. Utiyama, Q. Ma, H. Ozaku, and H. Isahara, “CRL at NTCIR2,” 2001, pp. 5–21–5–31.
[18] M. Murata, Q. Ma, and H. Isahara, “High performance information retrieval using many characteristics and many techniques,” 2002.
[19] H. Yamada, T. Kudoh, and Y. Matsumoto, “Japanese named entity extraction using support vector machine,” *Transactions of Information Processing Society of Japan*, vol. 43, no. 1, pp. 44–53, 2002.