A Question-Answering System Using Unit Estimation and Probabilistic Near-Terms IR

Masaki Murata, Masao Utiyama, and Hitoshi Isahara
Communications Research Laboratory
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{murata, mutiyama, isahara}@crl.go.jp

Abstract

Our question-answering system incorporates several new methods. One is unit estimation, which is useful when the answer is a numerical expression and the question sentence does not include any unit expression. This method can be used to estimate a unit expression for the answer by using a statistical test and corpus data and thus improves the results for such questions. Another new method is called probabilistic near-terms information retrieval. This method enables us to use full-size documents in document retrieval without dividing documents into passages and is useful when the answer is not within the passage where relevant terms occur. We confirmed the effectiveness of the unit estimation by using a statistical test (a T-test) and found that the probabilistic near-terms information retrieval improved the performance in Task-1 and Task-2.

Keywords: *Statistical test, Unit estimation, Probabilistic near-terms IR*

1 Introduction

We expect question-answering systems to become increasingly important as a more convenient alternative to systems designed for information retrieval and to be a basic component of future artificial intelligence systems. We have investigated the potential of question-answering systems [10] and have participated in the Question-Answering Challenge (QAC) of NTCIR 3. Our question-answering system reported at NTCIR 3 incorporates two new methods. One is unit estimation which is useful when the answer is a numerical expression and the question sentence does not include any unit expression. This method can be used to estimate a unit expression for the answer by using a statistical test and corpus data and thus improves the results for such questions. Another new method is probabilistic near-terms information retrieval. This enables us to use full-size documents in document retrieval without dividing documents into passages. This is useful when the answer is not within the paragraph where relevant terms occur.

2 Importance of a question-answering system

We have investigated question-answering systems with regard to various aspects [10, 11, 7]. Our experience has led us to believe that question-answering systems are particularly important in two respects:

- Question-answering systems are more convenient for users than information retrieval systems.

  Since a question-answering system shows users direct answers to their questions, users can easily acquire the information they want without having to read the pertinent documents. An information-retrieval system only identifies such documents, thus making it necessary for users to read through the documents to find the information they seek.

- Question-answering systems can be applied as a component of other intelligent systems.

  A question-answering system that can automatically extract answers can be used as a component of other intelligent systems. We think that a question-answering system will be a fundamental component of future artificial intelligence systems.

3 Explanation of our system

3.1 Outline of our system

Our system uses three ordinary algorithms as follows:

1. Estimation of answer type

The system estimates the answer to be a particular type of expression based on the expressions of an interrogative pronoun, adjective, or adverb of an input question sentence. For example, if the input question sentence is "How large is the area of Japan?", the expression "How large" suggests that the answer will be a numerical expression.

2. Document retrieval

   The system extracts terms from a question sentence and retrieve documents by using these terms. The retrieval thus gathers documents that are likely to contain the answer. For example, for the input question "How large is the area of Japan?", we extract 'large', 'area', and 'Japan' as terms and retrieve documents containing these terms.

3. Answer detection

   The system extracts linguistic expressions that match the estimated expression type of the answer as estimated in 1 from the documents retrieved in 2. It then outputs extracted expressions as the answer. For example, for the question "How large is the area of Japan?", the system extracts numerical expressions as the answer from documents containing 'large', 'area', and 'Japan'.

3.2 Estimation of answer type

Estimation regarding the answer type in our system is created based on manually created heuristic rules. Some of the heuristic rules are shown here.

1. When '何 (what) X1 何 (what) X2' is in a question sentence, WH-WH is given as an answer type. (e.g. "何年何月" (what year what month))
2. When 'X1 を何と言う' (what do we call X1) is in a question sentence, expressions corresponding to X1 are extracted as the NOUN-FOCUS.
3. When 'X1 はどこ' (where is X1) is in a question sentence, expressions corresponding to X1 are extracted as the NOUN-FOCUS.
4. When '誰' (who) is in a question sentence, PERSON is given as the answer type.
5. When 'どこ' (where) is in a question sentence and the NOUN-FOCUS is not '地域' (area), '場所' (location), or so on, ORGANIZATION is given as the answer type.
6. When 'どこ' (where) is in a question sentence and the NOUN-FOCUS is not '会社' (company), '組織' (organization), or so on, LOCATION is given as the answer type.
7. When '何 (what) + particles' is in a question sentence and NOUN-FOCUS is not '会社' (company), '地域' (area), or so on, ARTIFACT is given as the answer type.
8. When '何 (what) + particles' is in a question sentence and the NOUN-FOCUS is '地域' (area), or so on, LOCATION is given as the answer type.
9. When '何 (what) + suffix' is in a question sentence, the suffix is extracted as a unit expression.
10. When 'いつ' (when) is in a question sentence, TIME is given as the answer type.
11. When 'どのくらいの X1' (how many X1) is in a question sentence, the expressions corresponding to X1 are extracted as the UNIT-FOCUS.
12. When 'どのくらいの' (how many) is in a question sentence, NUM-EXP is given as the answer type.
13. When 'どこの国' (what country) is in a question sentence, SEM-COUNTRY is given as the answer type.
14. When '読みは何' (what spellings) is in a question sentence, SEM-SPELL is given as the answer type.

Our system uses a new method, which we call *unit estimation*, to obtain a correct unit expression for the answer. (This method is one main topic of this paper.) In this method, we gather sentences containing expressions like "UNIT-FOCUS + 'は' + 'numerical expressions' + 'unit expressions'" and extract the unit expressions.[1] We then eliminate unnecessary unit expressions by applying a statistical test based on a binomial distribution. Eliminated expressions are as follows:

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

where $P(e)$ is calculated by the following equation[2] and k_p is a constant that is set based on experimental results.

[1] Extracting unit expressions from the corpus was proposed by us in our previous paper [11].

[2] In this study, we used the summation of 0 to k, but the summation of 0 to $k-1$ could also be used. When the summation of 0 to k is used, an expression having a lower value for $P(e)$ is judged to be an expression that occurs less often than the average occurrence when UNIT-FOCUS co-occurs and it is eliminated. When the summation of 0 to $k-1$ is used, an expression having a higher value for $P(e)$ is judged to be an expression that occurs more often than the average occurrence when UNIT-FOCUS co-occurs and the expressions other than such an expression are eliminated. We plan to patent these methods.

Table 1. An example of using unit estimation

e	k	n	P(e)
メートル (meter)	50	128175	1.000000
センチ (centimeter)	28	47050	1.000000
ミリ (millimeter)	11	25897	1.000000
キロ (kilometer)	11	99618	0.999996
光年 (light-year)	2	538	1.000000
分 (minute)	2	955808	0.000000
ヤード (yard)	1	2744	0.998205
インチ (inch)	1	1865	0.999160
本 (piece)	1	1625073	0.000000
尺 (*shaku*)	1	2146	0.998892

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

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 + 'は' + 'numerical expressions' + 'unit expressions'" in the corpus, and $p(u)$ is calculated by

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

where 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 newspaper issues from over a 10-year period [5] as the corpus for the unit estimation calculation.

An example of using unit estimation is as follows. Consider the question sentence 'Xの長さはどのくらいですか' (How much is the length of X?). In this case, we extract a noun '長さ' (length) as the UNIT-FOCUS and gather candidate unit expressions using "長さ + は + 'numerical expressions' + 'unit expressions'". We obtain 'メートル' (meter), 'センチ' (centimeter), 'ミリ' (millimeter), 'キロ' (kilometer), '光年' (light-year), '分' (minute), 'ヤード' (yard), 'インチ' (inch), '本' (piece), and '尺' (shaku) as candidates. We calculate $P(e)$ for each candidate and obtain the results shown in Table 1. In this case, N is 533,366,720, the frequency of '長さ' is 11,887, and $p(u) = \frac{11,887}{533,366,720} = 0.000022289$. As shown in Table 1, our method can correctly eliminate '分' (minute) and '本' (piece). When using our unit estimation, we do not need a dictionary for unit expressions. Another good point regarding the unit estimation is that it gives 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.

3.3 Document retrieval

We extracted terms from a question sentence by using a morphological analyzer, ChaSen[6]. We eliminated terms whose parts of speech were postpositional particles or something, then performed retrieval using the extracted terms.[3] We developed a method that enables us to use a full-size document for document retrieval. In general, we divide each document into small passages before document retrieval. For example, we divided each document into sets of three sentences in our previous study [10, 11]. This overcomes the shortcoming of common information retrieval methods that cannot handle long documents for retrieval. In a question-answering system, the answer often occurs in the terms extracted from a question sentence. In this case, the condition that terms occur near each other is very important, but existing information retrieval methods have not taken this condition into consideration. As a result, divided passages have generally been used for document retrieval. The information retrieval method that we use for document retrieval in our question-answering system, however, considers the condition that terms occur near each other by using a probability theory. Our method is as follows.

1. We first retrieve the top k_{dr1} documents that have the highest scores from the equation

$$Score(d) = \sum_{\text{term } t} \left(\frac{tf(d,t)}{tf(d,t) + k_t \frac{length(d) + k_+}{\Delta + k_+}} \times log\frac{N}{df(t)} \right) \quad (4)$$

where d is a document, t is a term extracted from a question sentence, $tf(d,t)$ is the frequency of t occurring in a document d, $df(t)$ is the number of documents in which t appears, N is the total

[3] We used down-weighting methods [3, 9, 8] for the terms in Eqs. (5), (9), and (12) which allowed us to use compound terms. We also used synonyms in the EDR dictionary [2] to expand terms. When we used synonyms as terms in Eqs. (5), (9), and (12) instead of terms in a question sentence, we used the total frequency of the original term and its synomyms as $df(t)$.

number of documents, $length(d)$ is the length of a document d, and Δ is the average length of the documents. k_{dr1}, k_t, and k_+ are constants that are set according to experimental results.

This equation was based on Robertson's equation [13, 14]. This method is very effective and we have used it for information retrieval [9, 12, 8].

2. Next, we re-rank the extracted documents using the following equation and extract the top k_{dr2} documents. The k_{dr2} extracted documents are used in the answer extraction phase which comes next.

$$Score(d) = -min_{t1\in T} log \prod_{t2\in T3} (2dist(t1,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \quad (5)$$

$$= max_{t1\in T} \sum_{t2\in T3} w_{dr2}(t2) log \frac{N}{2dist(t1,t2) * df(t2)} \quad (6)$$

$$T3 = \{t | t \in T, 2dist(t1,t)\frac{df(t)}{N} \leq 1\} \quad (7)$$

where d is a document, T is a set of terms, and $dist(t1,t2)$ is the distance between $t1$ and $t2$ (using the number of characters as the distance) where $dist(t1,t2) = 0.5$ when $t1 = t2$. $df(t)$ is the number of times t occurs in all documents, N is the total number of characters of all documents, and $w_{dr2}(t2)$ is a function of $t2$ which is adjusted according to experimental results.

Equation (5) can be deduced when $w_{dr2}(t2) = 1$ through a probability theory. We consider the probability of a pattern of the occurrence of terms. First, we select $t1$. The occurrence probability of $t1$ is $\frac{df(t1)}{N}$. Next, we consider the other terms $t2$. The probability of each of these terms occurring within the distance $dist(t1,t2)$ to $t1$ is $2dist(t1,t2)\frac{df(t2)}{N}$[4] Therefore, the probability of an occurrence pattern is

$$\frac{df(t1)}{N} \prod_{t2\in T3, t2\neq t1} (2dist(t1,t2)\frac{df(t2)}{N}) \quad (8)$$

If we suppose that $dist(t1,t2) = 0.5$ when $t1 = t2$, the above equation becomes similar to Eq. (5). When the probability of pattern occurrence is low, the pattern is more valuable, so we extract documents that have low scores from Eq. (8). Since Eq. (5) is multiplied by -1, we extract documents having high scores from Eq. (5).

We call this information retrieval method that uses the pattern of nearby terms based on probability theory *probabilistic near-terms information retrieval*.[5]

3.4 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 led to the generation of too many candidates. Therefore, we now use only candidates that consist of only nouns, unknown words, and symbols. Also, we use ChaSen so that we can use morphemes and their parts of speech. Hiragana characters in personal names having the suffix 'さん' (Mr.) were sometimes incorrectly recognized as particles, so we added expressions whose first word are nouns, whose last word is 'さん' (Mr.), and whose length is less than 10 characters to the candidates as exceptions. We call this addition *use of exceptional person candidates*. Also, we added expressions surrounded by Japanese brackets '「' and '」' to the candidates because such expressions were often correct answers [15] and these expressions contain words such as particles.

Our method of judging whether each candidate is a correct answer is to add up the score for each candidate under the condition that the candidate is a nearby term with the score based on heuristic rules according to the answer type. Our system selects the candidates having the highest total points as correct answers.

We developed two methods for calculating the score under the condition that the candidate be near the terms of each candidate c. These are as follows.

$$Score_{near1}(c) = -log \prod_{t2\in T3} (2dist(c,t2)\frac{df(t2)}{N})^{w_{dr2}(t2)} \quad (9)$$

$$= \sum_{t2\in T3} w_{dr2}(t2) log \frac{N}{2dist(c,t2) * df(t2)} \quad (10)$$

$$T3 = \{t | t \in T, 2dist(c,t)\frac{df(t)}{N} \leq 1\} \quad (11)$$

[4] Strictly speaking, the probability is $1 - (1 - \frac{df(t2)}{N})^{2dist(t1,t2)}$. When we use Taylor's formula ($(1-x)^n = 1 - nx$ $(x \ll 1)$), the probability becomes $2dist(t1,t2)\frac{df(t2)}{N}$.

[5] We plan to patent this method also.

$$Score_{near2}(c) = \sum_{t2 \in T3} k_{near2}^{dist(c,t2)} w_{dr2}(t2) log \frac{N}{df(t2)} \quad (12)$$

$$T3 = \{t | t \in T, 2dist(t1, t) \frac{df(t)}{N} \leq 1\} \quad (13)$$

where c is a candidate for the answer, T is a set of terms, and $dist(c, t2)$ is the distance between c and $t2$. $df(t)$ is the number of times t occurs in all the documents, N is the total number of characters in all the documents, and $w_{dr2}(t2)$ is a function of $t2$ which is adjusted according to experimental results. k_{near2} is a constant set according to experimental results.

$Score_{near1}(c)$ is a very similar method to that of Eq. 5. This method also has a function to select a candidate by using nearby terms.

$Score_{near2}(c)$ is a heuristic function using the condition that the answer is near certain terms.

Next, we describe how the score is calculated based on heuristic rules using the estimated answer type. We used many heuristic rules to award points to candidates and used the total score as the deciding value. Some of the heuristic rules are shown below.

1. We add -1,000,000 to candidates that consist of unnecessary characters such as '→', '＊', and 'に関連記事' (has a related document in).

2. We add 1000 to candidates that match one of the estimated answer types (LOCATION, ORGANIZATION, PERSON, ARTIFACT, or TIME) when they are numerical expressions + UNIT-EXP or estimated unit expressions; when they are 'X1 年 (year) X2 月 (month) X3 日 (day)' where the question sentence contains '生年月日はいつ' (what is your birthday); when they have the meaning of jobs where the NOUN-FOCUS is '仕事' (work), '職業' (job), or so on; and when they have Noun X where the question sentence contains '何 (what) + Noun X', and so on.

   This is the most fundamental rule, and we adjust it so that a candidate that is a good answer will almost always satisfy the rule.

   Named entity (NE) extraction techniques are used to judge whether a candidate matches LOCATION, ORGANIZATION, PERSON, ARTIFACT, or TIME. We use a machine learning approach for our NE extraction system, and CRL (Communications Research Laboratory) NE data was used for training in the machine learning [16]. Only LOCATION, ORGANIZATION, PERSON, ARTIFACT, TIME, and DATE were used as categories in the CRL NE data. We made six NE extraction systems for these six categories. Each system judges whether a candidate expression belongs to a certain category. For example, the NE system for LOCATION judges whether a candidate is a LOCATION. In our system, one candidate can belong to more than one NE category. We used the support vector machine for machine learning [1, 4, 17]. When a candidate is judged to be a DATE, we classify it as a TIME because our question-answering system does not use a DATE category.

   Our method for judging whether a candidate matches LOCATION, ORGANIZATION, PERSON, ARTIFACT, or TIME uses two other powerful techniques. One improves the recall rate and the other improves the precision rate. The first technique is that we did not use the morphemes previous to and succeeding the analyzed NE as features in the NE judgment. This improves the recall rate.[6] The second technique is that we used heuristic rules to eliminate any wrong NEs after the NE judgment by machine learning. For example, we used the following rules for elimination.

   (a) A candidate whose last character is '系' (system), '社' (company), or so on is eliminated in the case of PERSON classification.

   (b) A candidate whose first characters is 'ため' (because), 'こと' (thing), or so on, is eliminated in the case of PERSON classification.

   (c) A candidate containing words other than words we selected in advance is eliminated in the case of TIME classification. (We first selected 'すぎ' (past), '半ば' (half), '夏' (summer), and so on as good words for TIME and eliminated candidates containing other words.)

   The rules shown here represent a very small sample of our heuristic rules. These rules were made by manually checking a large body of results from our NE system using a large amount of raw data[7] as input data.

3. When WH-WH is one of the estimated answer types, we add 1000 to candidates that match a

[6] We also used other methods to improve the recall rate. Katakana expressions whose parts of speech are unknown words are classified as LOCATION, ORGANIZATION, PERSON, or ARTIFACT. A kanji character + any character + 'さん' (Mr.) is classified as PERSON. Candidates that are in our dictionary of companies are classified as ORGANIZATION.

[7] This raw data had no NE tags, and was part of the documents that were used for the QAC (Mainichi newspaper 1998, 1999).

pattern of 'X1 P1 X2 P2' which uses the interrogative pronoun '何 (what) P1 何 (what) P2'.

4. When the NOUN-FOCUS is '仕事' (work), '職業' (job), or so on, we add 1000 to candidates that have the meaning of jobs. The judgment as to whether a candidate has the meaning of a job is made by using the EDR thesaurus [2].

5. When SEM-COUNTRY is one of the estimated answer types, we add 1000 to candidates that are in our dictionary of countries which includes the names of almost all countries.

6. When SEM-SPELL is one of the estimated answer types, we add 1000 to candidates that consist of only hiragana characters and the symbol '・'.

7. When a question sentence includes '都道府県はどこ' (what prefecture), we add 1000 to candidates that are location expressions and whose suffix is '都', '道', '府', or '県' (prefecture).

8. When the NOUN-FOCUS is '名作' (masterpiece), '作品' (literary work), or so on, we add 1000 to candidates that are surrounded by brackets '「' and '」'.

9. When NUM-EXP is one of the estimated answer types, we add 1000 to candidates that consist of only numerical expressions and unit expressions.

10. When a question sentence contains 'の記号は何' (what symbol), we add 1000 to candidates that consist of symbols.

11. When a question sentence contains '電話番号は何番' (what is the telephone number), we add 1000 to candidates that consist of numerical expressions and the symbol '・'.

12. When a question sentence contains '何 (what) + Noun X', we add 1000 to candidates having the Noun X.

13. When a question sentence contains '何 (what) + 曜日 (day of the week)', we add 10,000 to candidates that are '日曜日' (Sunday), '月曜日' (Monday), or so on.

Our system has additional functions that are used after answers are selected based on the scores. The first one 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 1000. The compiling is done by eliminating answers other than the longest one. We call this method *answer compiling*. The second function expands foreign personal names. For example, when we obtain XXXXX (katakana characters) as an answer for PERSON, we gather 'YYYYY・XXXXX' and the 'YYYYY・XXXXX' that has the highest frequency is selected as the answer. We call this method *name expansion*.

Now, we will describe our strategy for the QAC. In task-1, where the MRR is used in the evaluation, our system outputs the top five answers.[8] In task-2 and task-3, where the mean of the f-measure (MF) is used in the evaluation, our system outputs the top answer.[9] In task-3, we use as the second question sentence the connection of the first and second question sentence where an interrogative pronoun, adjective, or adverb in the first sentence is changed to dummy symbols.

4 Experiments

Our experimental results are shown in Table 2.[10] Experiments S2 to S11 were performed after the formal run to confirm the effectiveness of methods used in our system. S4 is the method by which documents are divided into paragraphs after using Eq. 4 and before using Eq. 5. S5 did not use unit estimation at all and did not use any estimated unit expression. S6 did not use a statistical test to eliminate unnecessary unit expressions and used all the candidate unit expressions gathered from the corpus as unit expressions. S10 is the method using as the second question the connection of the first question sentence, the highest-ranked answer of the first question sentence, and the second question sentence for task-3. S11 is a *select-two method* that outputs the two top answers. The experiments for S12 to S14 were done to obtain the highest score using the results of S2 to S10. We used a one-sided T-test to statistically confirm the effectiveness. We used S1 as the base method for comparison. The figures denoted by '+' were superior to the base method result at a significance level of 0.05 and the figures denoted by '−' were inferior at a significance level of 0.05. The parameters were $k_p = 0.9$, $k_t = 0.00001$, $k_+ = 20$, and $w_{dr2}(t) = 0.3$ when a term t was a verb and otherwise was 1. We used $Score_{near1}(c)$ for other than S7 and S8. The heuristic rules in our system were manually devised by referring to the questions in a dry run and about a hundred questions constructed by us. We used documents retrieved from Mainichi newspaper articles (taken from the issues over two years: 1998 and 1999) [5] which

[8] We explain MRR and MF in Section 4

[9] Our system has no function for selecting multiple answers, so we use only one answer to maintain good precision. We call this strategy the *select-one method*.

[10] Our submitted systems obtained the highest MF in Task-2. In Task-1 and Task-3, our systems obtained the second highest MRR and MF. (The highest MRR in Task-1 was 0.608. The highest MF in Task-3 was 0.187.) This indicated that our system worked reasonably well. Therefore, our results from the comparison of several methods in our system can be used for future studies regarding question-answering systems.

Table 2. Experimental results

	Method	Task-1 (MRR)	Task-2 (MF)	Task-3 (MF)
S1	Submitted system	0.524	0.364	0.167
S2	No use of name expansion	0.531	0.366	0.192
S3	No use of answer compiling	0.493^{-}	0.339	0.167
S4	Documents divided into paragraphs	0.513	0.355	0.250
S5	No use of unit estimation	0.501^{-}	0.341^{-}	0.167
S6	No use of statistical test for unit estimation	0.521	0.361	0.167
S7	Use of $Score_{near2}$ with $k_{near2} = 0.99$	0.521	0.378	0.225
S8	Use of $Score_{near2}$ with $k_{near2} = 0.999$	0.513	0.382	0.167
S9	No use of exceptional person candidates	0.526	0.361	0.167
S10	Use of the answer of the first question	—	—	0.225
S11	Use of the select-two method	—	0.341	0.213
S12	Combination of S2 and S9	0.532	—	—
S13	Combination of S2 and S8	—	0.385	—
S14	Combination of S2, S4, S7, and S10	—	—	0.325^{+}

were the documents that we extracted answers from in the QAC task. We did not use any other documents for document retrieval. The QAC had three tasks (Task-1, Task-2, and Task-3). Task-1 and Task-2 used the same 200 questions. Task-3 used 40 sets of two questions. The second question of each set was related to the first question. For example, the first sentence would be "日本の人口はいくらですか " (How many people are there in Japan?) and the second sentence would be "面積はいくらですか " (What is the size of the area?). In Task-3, only the second sentences were evaluated. In Task-1, we could submit five or less answers for each question and MRR was used for evaluation. In Task-1, the first correct answer among the five answers was used for the MRR calculation. In the MRR, we obtained a score of $1/r$ when the r-th submitted answer was correct. In Task-2 and Task-3, we could submit any number of answers for each question and the MF (mean of the f-measure) was used for evaluation. In Task-2 and Task-3, when the question had more than one answer, we had to submit the corresponding multiple answers to obtain full marks. The recall rate in the f-measure was the ratio of the number of correct answers submitted to the number of correct answers. The precision rate was the ratio of the number of correct answers submitted to the number of submitted answers.

We found the following from the experimental results.

- Our method of unit estimation was confirmed to be effective by statistical testing (the T-test) (c.f. S5). Our method of using statistical test for unit estimation slightly increased the performance of unit estimation (compare S1 and S6).[11]
- Answer compiling was also effective, as was confirmed by statistical testing in Task-1 (c.f. S3).
- Our method of probabilistic near-terms information retrieval sometimes provided better results (c.f. S1 and S4 in Task-1 and Task-2). However, we could not confirm its effectiveness through the T-test.
- $Score_{near1}(c)$ was good in Task-1. $Score_{near2}(c)$ was good in Task-2 and Task-3 (c.f. S7 and S8).
- Use of name expansion and exceptional person candidates lowered the scores (c.f. S2 and S9).
- In Task-3, we applied a new method of using the answer of the first question because our system obtained higher scores for the top answer than we estimated.[12] This addition greatly improved the score (c.f. S1 and S10).
- We created a tune-up system for each task using the results of S2 to S10 (c.f. S12, S13, and S14). The scores slightly increased in Task-1

[11] We examined the results of using unit estimation in Task 2 more minutely. There were fourteen questions where our system judged that a question has no unit expression and its answer type is a numerical expression. Among them, our system using unit estimation (S1) obtained more than zero score in eight quetions. When our system did not use unit estimation (S5), it obtained more than zero score in only three quetions. (S1 always obtained correct answers when S5 obtained correct answers.) When our system did not use statistical test for unit estimation (S6), it obtained more than zero score in seven quetions.(We compared S1 and S6 and found that there were two question when the answer of S1 was correct and the answer of S6 was incorrect and there was one question when the answer of S6 was correct and the answer of S1 was incorrect.)

[12] In the formal run, we thought our system's performance was low and adding the answer of the first question decreased the score.

and Task-2 and greatly increased in Task-3. The improvement in Task-3 was significant according to the T-test.

- In Task-3, we observed very few significant differences. This may have been due to the small number (40) of questions in Task-3.

- In S11, we used the *select-two method*. In Task-2, the score decreased as we estimated. However, we did not estimate the increased scores in Task-3. The performance of our system for Task-3 was not particularly good. We found that when the system performance was poor, the *select-two method* could be superior to the *select-one method*.[13]

5 Conclusion

Our question-answering system uses several new methods. One is unit estimation. This method is very useful when the answer is a numerical expression and a question sentence does not include any unit estimation. This method can be used to estimate a unit expression for the answer by using a statistical test and corpus data and thus improves the results for such questions. Another new method is the probabilistic near-terms information retrieval. This method enables us to use full-size documents in document retrieval without dividing the documents into passages. This method is useful when the answer is not within the paragraph where relevant terms occur. We confirmed the effectiveness of the unit estimation by using a statistical test (T-test) and found that probabilistic near-terms information retrieval improved the performance in Task-1 and Task-2.

In our previous question-answering system, we used other original methods based on syntactic structures and rewriting rules (paraphrasing) [10, 11, 7]. However, we did not use these techniques in the QAC because they are time consuming. In our future work, we hope to use these techniques to improve our system with regard to the QAC.

The QAC has drawn many researchers into the field of question-answering systems and has helped to improve the question-answering performance that can be attained. We hope to see further rapid progress in this field and the future development of artificial intelligence using such systems.

References

[1] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[2] EDR. *EDR Electronic Dictionary Technical Guide*. EDR (Japan Electronic Dictionary Research Institute, Ltd.), 1993.

[3] S. Fujita. Notes on phrasal indexing JSCB evaluation experiments at ntcir adhoc. *Proceedings of the NTCIR Workshop 1*, pages 101–108, 1999.

[4] T. Kudoh. TinySVM: Support Vector Machines. http://cl.aist-nara.ac.jp/ taku-ku// software/TinySVM/ index.html, 2000.

[5] Mainichi Publishing. Mainichi Newspaper 1991-2000, 2000.

[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] M. Murata and H. Isahara. Universal model for paraphrasing — using transformation based on a defined criteria —. In *NLPRS'2001 Workshop on Automatic Paraphrasing: Theories and Applications*, 2001.

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

[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. Question answering system using similarity-guided reasoning. *Information Processing Society of Japan, WGNL 2000-NL-135*, pages 181–188, 2000.

[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] 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.

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

[15] Y. Sasaki, H. Isozaki, H. Taira, T. Hirao, H. Kazawa, J. Suzuki, K. Kokuryo, and E. Maeda. SAIQA: A Japanese QA system based on a large-scale corpus. *IPSJ-WGNL 2001-NL-145*, 2001. (in Japanese).

[16] K. Uchimoto, M. Murata, H. Ozaku, Q. Ma, and H. Isahara. Named entity extraction based on maximum entropy model and transformation rules. *Proceedings of 33rd Annual Meeting of the Association of the Computational Linguistics*, 2000.

[17] 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.

[13] When we used the *select-two method* for S14 of Task-3, we obtained an MF of 0.279 which was lower than the score for S14.