Proceedings of NTCIR-4, Tokyo, April 2003 - June 2004

GDQA : Graph Driven Question Answering System
– NTCIR-4 QAC2 Experiments –

Gakuto Kurata Naoaki Okazaki Mitsuru Ishizuka
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{kurata, okazaki, ishizuka}@miv.t.u-tokyo.ac.jp

Abstract

This paper is a detailed presentation of a question answering system developed for NTCIR-4. Our question answering system, GDQA, employs both a new text retrieval algorithm that is specialized for QA and a new algorithm for sorting the answer candidates. Using our new algorithm for text retrieval, articles containing the answer for the question can be retrieved with high precision. Our algorithm for sorting answer candidates uses a graphic structure based on the result of the dependency analysis of retrieved articles. Using this sorting algorithm, GDQA can present the correct answer in a higher rank than other candidates.

1 Introduction

Question Answering (QA) is a task to present an appropriate answer for a question written in natural language from a large corpus that is available as computerized forms. QA is widely recognized as a complex of Information Retrieval, Information Extraction and other natural language processing techniques. Numerous researchers are addressing QA tasks and participating in evaluation workshops such as TREC[9] and NTCIR[11], which are intended to enhance research.

Tab. 1 shows the difference between QA and Information Retrieval (IR).

Using an IR system like Google, we can only retrieve documents that are related to the terms that we input. For that reason, we must read the documents through and search again for the information we need. In contrast, we can get an answer to a query directly using a QA system. A QA system saves our labor in the exemplary IR system shown below.

- To determine terms from our information need
- To read documents to seek the information we need

The ubiquitous computing era is coming. One can easily imagine a situation in which people converse with distributed computers and ask some questions. Thereupon, the computers could return an answer. Question Answering and Speech Recognition are essential technologies to realize this vision.

According to this vision, QA is a promising field. However, the performance of current QA systems is very poor. This paper presents new techniques to improve that performance.

2 Related works

Numerous systems have been established. Most of those systems employ four steps. Fig. 1 shows the common architecture of conventional QA systems.

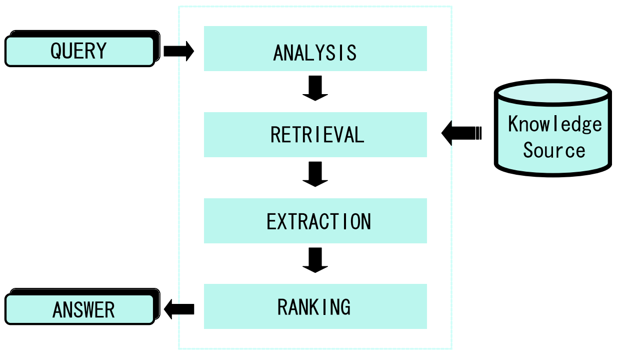


Figure 1. Common Architecture of QA systems

We will briefly describe the approach used in conventional QA systems to realize the four procedures shown in Fig. 1. At the same time, we will point out problems of the former approaches.

2.1 Common Architecture of QA system

2.1.1 ANALYSIS of the query

The first step analyzes queries that are written in natural language.

For example, a query like “Who is the president of the United States?” indicates that the answer is the name of a person. Interrogative words like who, where, and when suggest the answer category. Tab. 2 shows the interrogative words in Japanese and categories they suggest.

Some inconsistency exists between Japanese and English correspondence listed above. Nevertheless, the fact that the interrogative word suggests the category of the answer never changes across languages. Most systems classify queries according to this truism. The number of classifications is very important and varies from small to large. For example, SQAIQA, which NTT Communication Science Laboratories developed, contains 80 categories with hand-made rules[12] and SAIQA-2 into about 160 categories with Support Vector Machines.

Pasca et al. classifies queries into 22 categories and more subcatogories[6]. Describing their approach in de-

Table 1. Difference between QA and IR

	QA	IR
Input	Queries written in natural language	Query terms
Output	Answers for the query	Related documents

Table 2. Correspondence between interrogative words and categories of answers

Interrogative words		Categories
誰，だれ (Who)	$\rightarrow$	Person
何処，何所 (where)	$\rightarrow$	Place
どこ (where, which)	$\rightarrow$	Location, Company, Organization
いつ (When)	$\rightarrow$	Time, Date

tail is beyond the scope of this paper. Briefly, their approach is based on a lot of pairings of a query and its category. Their system performs well[16]. However, constructing the dataset of pairings like this takes a lot of time and money.

Overall, classifying queries into many categories is costly and time consuming. The following process depends on the results of classification. Therefore, an error in the classification step engenders errors in subsequent processes and renders the QA system incapable of presenting a correct answer. From this viewpoint, classification precision is vitally important. Notwithstanding, a certain error with classification must exist with machine learning.

2.1.2 RETRIEVAL of documents

In this step, documents which may contain the answer from large corpora are retrieved according to the query.

Differently from the IR system, in which query terms are given, terms to retrieve the documents must be selected from the query. In this step, a Japanese full-text search engine like Namazu[8] and techniques for IR are available.

One salient problem is the retrieval algorithm. Among conventional QA systems which use newspaper articles as a knowledge source, some only use Namazu, which adopts a simple algorithm or the $tf \cdot idf$ algorithm, others establish original search engines which employ the $tf \cdot idf$ algorithm, the $Okapi$ algorithm, or other[3].

Another problem in this step is which phonemes, terms or compound terms should be used as the index of the corpora and query terms. Briefly speaking, a full-text search engine is realized in the manner shown below.

1. The index file contains correspondence between the document and the phonemes, terms or compound terms in it.

2. The search engine calculates the similarity between the query terms set and the index of all documents by checking whether the document contains the query term or not; then, it outputs the documents that have high similarity.

Therefore, the form of the terms from phonemes to compound words must be unified between the index and the query terms. Consistency of the dictionary is extremely important for morphological analysis.

The system established by Takaki et al. uses the junction of the phonemes as query terms[15].

2.1.3 EXTRACTION of the answer candidates

In this step, answer candidates are extracted from the retrieved documents. What are called "answer candidates" are expressions which belong to the category determined in the ANALYSIS phase.

For example, when the query asks the name of a person using "who", expressions which may mean the name of person are extracted in this step. This kind of procedure has been investigated in recent years and implemented as Named Entity Extraction. $NExT$[7] is well known as the named entity extraction tool. $NExT$ extracts the seven kinds of named entities shown below from raw texts.

7 named entities of $NExT$

ORGANIZATION, PERSON, LOCATION, DATE, TIME, MONEY, PERCENT

The answer candidates for the query which suggest that the answer is the specific named entity can be extracted using the named entity extraction tool. However, answer candidates for a query which does not suggest the category of the named entity extraction tool can not be extracted. The query "What does DVD stand for?" is a proper example for this case. In this case, compound words and unknown words are extracted as answer candidates in most systems.

The system developed by Nomoto defines 29 named entity categories and establishes their original named entity extraction tool[10]. Their named entity extraction tool is based on 178 hand-made rules.

2.1.4 RANKING the candidates

In this step, answer candidates are ranked from the viewpoint of suitability as an answer to the query. Probably, there exist plural answer candidates. For that reason, this procedure is important.

However, it is readily apparent that it is difficult for a computer to distinguish a correct answer from other answer candidates that belong to the same category as the correct answer. For that reason, this step is the most difficult procedure in QA systems.

Numerous approaches are proposed for this step. Conventional systems have ranked answer candidates according to the simple distance between the answer candidates and the query terms in the retrieved documents. This approach is based on the assumption that the answer appears in the corpora near the terms in the query. This assumption is appropriate. Nevertheless, no conventional system performs very well.

Another system uses tree structure[14]. This system is based on Collin's tree kernel[13]. However, this does not perform well.

Other systems use a lot of rules. Typical example of this kind is Site Q/J[5]. It performs very well[1]. But, making a lot of rules is costly and time consuming.

2.2 Problems of conventional approaches

We will briefly point out some problems of conventional approaches.

2.2.1 Number of classifications

It is expensive and time consuming to classify queries into many categories. Additionally, if the classification is based on a method with machine learning, the classification accuracy can not be 100%. Nevertheless, the accuracy of the classification is very important.

If many categories are defined, named entity extraction tools are necessary to classifiy the answer candidates into one category from many. It's obvious that the more the categories increase, the more difficcult this procedure becomes. The F-measure of $NExT$ is about 75 for 7 categories. Defining more categories markedly degrades the accuracy of named entity extraction. Consequently, the system performance will worsen.

Judging from these discussions, an ideal system would need a method with a small number of categories and good accuracy of classification.

2.2.2 The RETRIEVAL algorithm

In information retrieval systems like Google and full-text search engines like Namazu, all query teams are assumed to be equal. However, there are difference between terms in the query in the respect that the documents must contain the term or not. Therefore, it is necessary to deal with query terms separately and properly according to the parts of speech or some other measure.

Similarly, which parts of speech are used as query terms and the index for the document is also important. Namazu can not properly deal with this issue because it functions as a black box.

2.2.3 The simple distance between the answer candidate and the terms in the query

The assumption, "the answer appears in the corpora near the terms in the query" is reasonable. Conventional systems use simple distance between them. However, the distance between terms which have a close relation can be greater than those having little or no relation. An example of this fact is shown below.

Problem of the simple distance

Prince Charles met with earthquake survivors in the flattened city on Bam after talks talks with President Mohammad Khatami earlier Monday in the first visit to Iran by a member of the British royal family in 33 years.

Simple distance in sentences like this, which consist of nested clauses, does not reflect the relation. Such instances are much more common in Japanese texts.

2.3 Necessary techniques

Techniques listed below are necessary to establish a Japanese Question Answering system.

- A Japanese morphological analyzer
- A Japanese dependency analyzer
- A Japanese full-text search engine
- A Japanese Named Entity Extraction tool

3 Proposed method

This section explains the proposed method.

3.1 Retrieval Algorithm

First, we establish our original search engine, the Geta Based Search Engine (GBSE). GBSE is based on GETA[4]. GBSE characteristics are shown below.

- Its response is faster than Namazu.
- Documents are scored according to the $tf \cdot idf$ algorithm.
- Priority can be set for each retrieval term. The priority consists of two values: high and low. Documents without retrieval terms with high priority can not be retrieved in GBSE.

Our retrieval algorithm using GBSE is as follows.

Indexing:1 Newspaper articles are segmented into paragraphs.

Indexing:2 Each paragraph is analyzed morphologically.

Indexing:3 The paragraph id and phonemes in it are recorded in the index file. In this phase, phonemes in the specific POS are used for the index.

Query:1 The query is also analyzed morphologically and phonemes in the specific POS(Part Of Speech) are chosen for the retrieval phonemese.

Query:2 Phonemes in certain POS such as proper nouns are categorized into essential query phonemese; others are categorized into optional query phonemes.

Retrieval:1 The priority of all retrieval phonemes is set as high. GBSE tries to find paragraphs. If retrieval succeeds, the retrieved paragraphs are used in the next step. [INITIAL RETRIEVAL]

Retrieval:2 The priority of the optional query phoneme with the highest TF in all newspaper articles is set to low. Then GBSE tries again. If retrieval fails again, the priority of the optional query phonemes are set to low one by one according to the TF and GBSE; then retrieval is attempted repeatedly. In this phase, the priority of the essential retrieval phonemes is kept high. [REDUCED RETRIEVAL]

Retrieval:3 When all optional query phonemes are set to low, retrieval stops and GDQA does not return the answer. (During the retrieval, at least one optional query phoneme is set to high.)

Benefits shown below are gained using this algorithm.

- Using a phoneme in the index and analysis of the query, compound words are managed easily while maintaining the consistency of the dictionary.
- It is flexible in which POS are used for the index.
- System tuning is easy because it indexes articles and retrieves the paragraph more quickly than Namazu.

3.2 Classification of queries according to the expected answer type

Queries are classified into four categories in our system. We will explain these categories briefly.

Type 1 Queries which suggest the suffix of the answers are classified into this category. The number of the answer candidates with a suggested suffix is very small. Therefore, queries in this category are easy to answer for GDQA.

Type1

QAC2-10002-01 Which prefecture does Katakura Kunio, the Japanese ambassador in Iraq at the time of the Gulf War, come from?

Type 2 Queries using interrogative words which suggest the type of the answers are classified into this category. For example, if the query has the interrogative word, "who", the answer for this query is the name of the person. GDQA extracts the answer candidates for the queries in this category with the named entity extraction tools.

Type2

QAC2-10060-01 Who was the Minister of Finance of the Obuchi Cabinet?

Type 3 Queries which request numeric expressions for answers are classified into this category. Interrogative words and adjectives, such as "how long", "how tall" and "how many", suggest the numeric expressions as answers. GDQA extracts expressions that consist of numbers and units as answer candidates for queries in this category.

Type3

QAC2-10032-01 How tall was Giant Baba?

Type 4 Queries which are not classified into the categories above are classified into this category. No apparent clues for the answer are found in the query. Therefore, GDQA extracts all nouns and conjunctions of the nouns as answer candidates. Consequently, the number of the answer candidates is very big: it is difficult for GDQA to present the correct answer.

Type4

QAC2-10067 What kind of recycled material was used to make the solar boat Mr. Horie Kennich used to cross the Pacific Ocean alone without making any port calls for the first time ever?

3.3 Ranking the answer candidates with a graphic structure

Answer candidates are ranked on the assumption that the terms in the query and its answer appear near the retrieved texts.

3.3.1 Construction of the graphic structure from retrieved documents

1: Dependency Analysis Each sentence in the retrieved paragraphs was analyzed with CaboCha. The result of this dependency analysis comprises of clauses.

2: Simplification of the result The result of the dependency analysis is a directed graph. First, it is transformed into an undirected graph[Transformation]. Then, the clauses are transformed with the following rules for simplification.

- Particles and auxiliary verbs are eliminated.
- Punctuation marks and case arcs are eliminated; other symbols are assumed as dependent nouns.
- Unknown words are inferred to be independent nouns.
- If an independent verb or an independent adjective is included, its root form is made into a stand-alone node.
- If independent nouns are included, their junction is made into a stand-alone node.
- Clauses only with dependent nouns, dependent verbs, dependent adjectives and stopwords are made into dummy nodes. Dummy nodes only present the path in the graph structure.

3: Graph Structure A single graph structure was made from simplified results with identical clauses merged into one node.

4: Shrinking Plural nodes with identical meanings exist in the graph structure. A thesaurus is necessary to solve this problem. However, a typical thesaurus which is constructed so that all words cover a broad meaning, merges nodes with different meanings in context. For that reason, a thesaurus that is specialized for retrieved documents is necessary to solve the problem. Constructing a thesaurus is a very difficult task. In this phase, we attempt to deal with the problem partly.

- English Abbreviation

ex JT $\Longleftrightarrow$ 日本たばこ産業 (Japan Tabaco, Inc)

- Birth names and common names

ex イチロー (Ichiro) $\Longleftrightarrow$ 鈴木一朗 (Ichiro Suzuki)

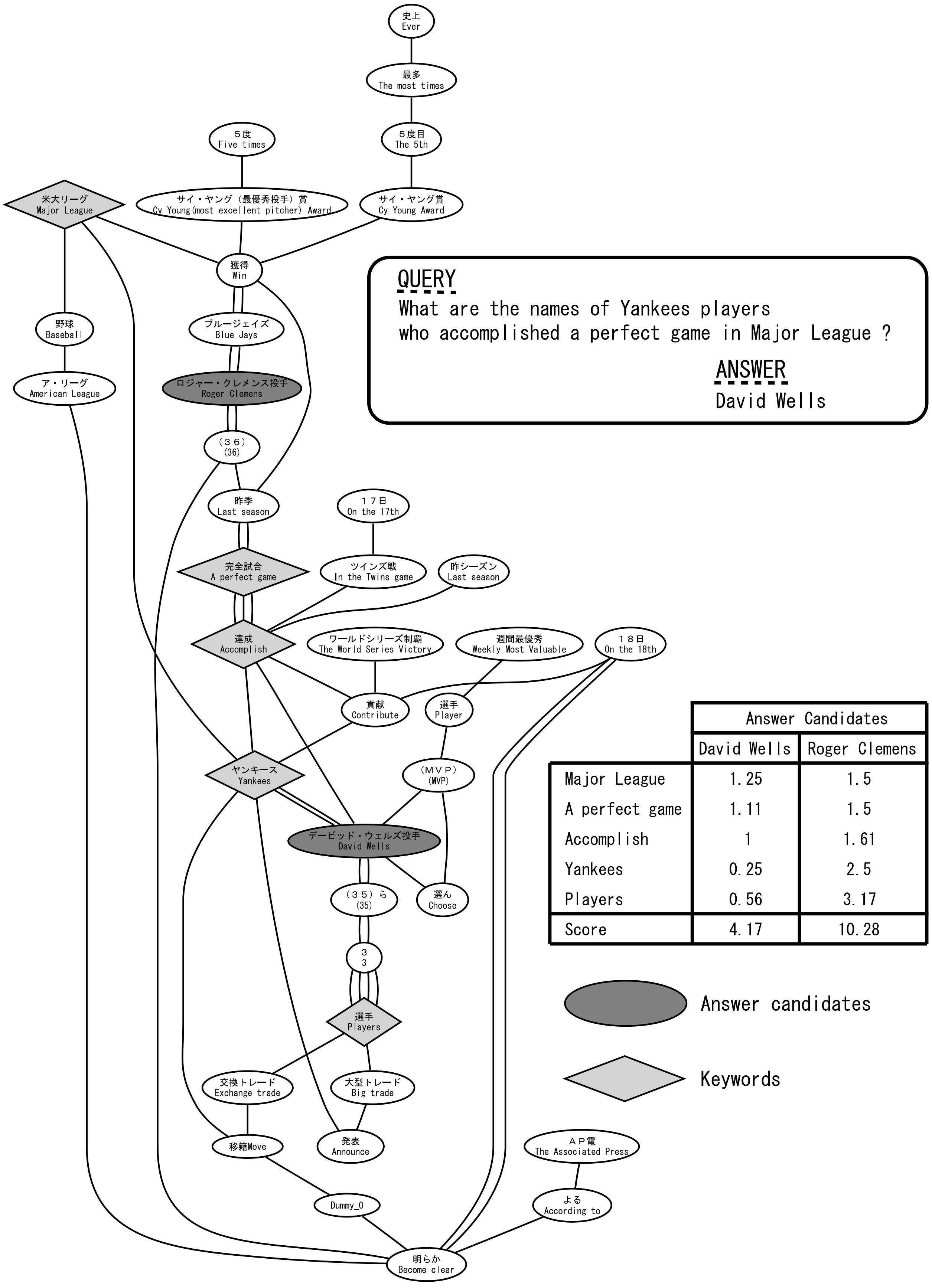
- Name Abbreviation

ex 小泉首相 (Prime Minister Koizumi) $\Longleftrightarrow$ 小泉純一郎首相 (Prime Minister Junichiro Koizumi)

You can see the example of the graph structure in Fig. 2

3.3.2 Definition of the graph structure distance

To define the distance between any two nodes in the graph structure, the distance between two connected nodes must be defined first. The distance between two connected nodes is defined as Equ. (1)



QUERY
What are the names of Yankees players
who accomplished a perfect game in Major League ?

ANSWER
David Wells

	Answer Candidates	
	David Wells	Roger Clemens
Major League	1.25	1.5
A perfect game	1.11	1.5
Accomplish	1	1.61
Yankees	0.25	2.5
Players	0.56	3.17
Score	4.17	10.28

Figure 2. Graph Structure Example

$$distance(node_1, node_2) = \frac{1}{{N_{12}}^2} \tag{1}$$

In Equ. (1), N_{12} indicates the number of links between $node_1$ and $node_2$. The stronger the relation between two connected nodes, the smaller the distance. This is a reasonable definition.

After the distance between any two connected nodes is defined with Equ. (1), the smallest distance between any two nodes in the graph structure is calculated with Dijkstra's algorithm.

3.3.3 The score of the answer candidate

According to that assumption, the score of the answer candidates should express the relation with the keywords in the query. Here, the conjunction of the phonemes in the certain POS shown bwelow are selected as the keywords from the query sentece.

Keywords

nouns(proper nouns, general nouns, numbers and Japanese specific nouns), suffixes, verbs, adjectives, alphabets, unkown words

The scores of the answer candidates are defined as Equ. (2)

$$Score(Candidate) = \sum_{All\ keywords} Min(A\ keyword, Candidate) \tag{2}$$

In Equ. (2), Min expresses the smallest distance.

All answer candidates can be sorted according to this score. The smaller the score of the candidate is, the heigher the rank becomes. The system presents the answers in ascending order.

4 Experiments

This section presents the results of experiments on NTCIR-4 QAC2 Subask 1 . Overview of NTCIR-4 QAC2 Subtask 1 is beyond of the scope of this paper[2].

4.1 Evaluation method

The system extracts five answers from documents in some order. The inverse of the order of the correct answer, the Reciprocal Rank (RR), represents the score of the question. For example, if the second answer is correct, the score will be 1/2. The highest score of the five answers will be the score of the question. If there are several correct answers of a question, the system might return one of them, but not all of them. The Mean Reciprocal Rank (MRR) is used for evaluation of Subtask 1. If n sets of answers are correct, the Mean Reciprocal Rank can be calculated as the following.

$$MRR = \frac{\sum_i^n RR_i}{N}$$
$$RR_i = \frac{1}{Rank}$$

4.2 Experimental Conditions

Experimental conditions are listed in Tab. 3.

Table 3. Experimental conditions

knowledge source	newspaper articles of four years
	Mainichi('98, '99) # of articles : 220078 115522('98), 104556('99) size : 284 MB 147 MB('98), 137 MB('99)
	Yomiuri('98, '99) # of articles : 375980 132995('98), 242985('99) size : 496 MB 184 MB('98), 312 MB('99)
# of questions	200 NTCIR-4 QAC2 Subtask 1 Formal Run

4.2.1 Maximum number of the paragraphs

The relation between the MRR and the maximum number of the paragraphs used in the phase of answer candidate extraction and the graph structure is shown in Fig. 3.

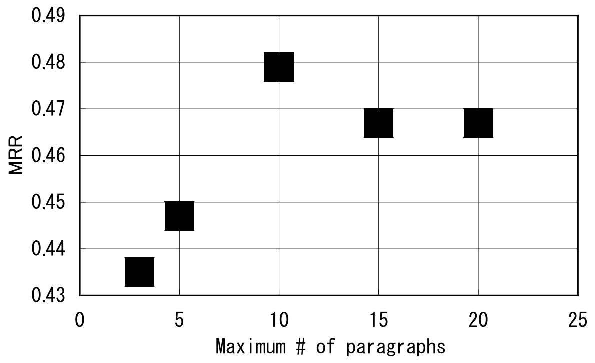


Figure 3. MRR and the maximum number of paragraphs

This pilot experiment was based on the QAC1 dataset[1]. This figure suggests that too many paragraphs have no effect on performance because paragraphs are ordered according to the relation to the query. For that reason, we set the maximum number of paragraphs to 10.

4.3 Results

The system performance is evaluated with the measure, MRR. The result given by the common scoring tool is shown below.

RESULTS

Task1 Results
83.8 marks out of 195.0 in TASK1
Average Score: 0.430

Question	Answer	Output	Correct
197	385	632	116
Recall	Precision	F-Value	MRR
0.301	0.184	0.228	**0.430**

The MRR measure is good for comparison among systems. However, we can not judge performance directly from it . Rate.1st, the rate at which the system presented the correct answer with 1st rank, was **0.349**. Rate.5, the rate at which the correct answer was found in the first five answers of the system's output, was **0.538**.

We will discuss further these results in section 6.

5 Additional Experiments

This section explains two additional experiments. The former shows the effect of the transformation of the graph structure from a directed graph to an undirected graph. The latter shows the advantage of GBSE and ranking with a graph structure.

5.1 Transformation of the graph structure

The dependency analysis result is transformed into an undirected graph in our approach. Sentences using active voice and the passive voice can be expressed in the same graph structure with this transformation. However, information can be decreased to a certain degree with this transformation.

Therefore, we made experiments with two conditions to ensure the transformation. One was with the transformation (undirected graph); the other was without transformation (directed graph). Results are shown below in 4.

Table 4. MRR with and without transformation

	Undirected Graph	Directed Graph
MRR	0.430	0.384

This table shows the effectiveness of the transformation.

5.2 GBSE and ranking with graph structure

These experiments compare the two retrieval algorithm and the two ranking algorithm. Results are shown in Tab. 5. This comparison was also based on the QAC1 dataset.

Table 5. GBSE and Graph

Retrieval	Ranking	MRR
Namazu	Frequency	0.305
Namazu	Graph	0.375
GBSE	Frequency	0.431
GBSE	**Graph**	**0.479**

First, Tab. 5 illustrates the advantages of GBSE over Namazu.

The algorithm of "frequency" is a very simple method to rank the answer candidates. It sorts the answer candidates according to their own frequency in the retrieved documents. Our graph structure approach is superior to this method.

5.3 Comparison between INITIAL RETRIEVAL and REDUCED RETRIEVAL

Fig. 4 shows the percentage of cases in which the top N(N=1,2,3,4,5) candidates contain the correct answer to the query when INITIAL RETRIEVAL succeeds and REDUCED RETRIEVAL succeeds after INITIAL RETRIEVAL fails.

From Fig. 4, the difference between INITIAL and REDUCED RETRIEVAL is apparent. Additionally, if

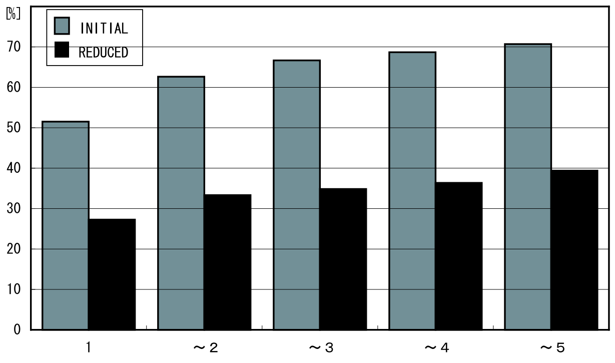


Figure 4. Comparison between INITIAL RETRIEVAL and REDUCED RETRIEVAL

retrieval were initiated with looser conditions than REDUCED RETRIEVAL, the result would be very poor.

6 Discussion

6.1 Comparison with other systems

The MRR of the top system of NTCIR-4 QAC2 was 0.607. The MRR of GDQA was lower. However, GDQA has two characteristics.

- Minimum number of hand-made rules
- No output to difficult queries

At first, the number of hand-made rules is very small. A larger number of hand-made rules might improve the QA system performance. However, producing many hand-made rules is costly and time consuming. Moreover it is difficult to add new modules into a QA system with too many rules.

Therefore, almost all systems return 1000 answer candidates to 200 queries. This means that they return answer candidates to all queries blindly. In marked contrast, GDQA does not return answers beyond the limits of reason when GBSE does not work well. Considering 5.3, this is reasonable. Additionally, it is better to be shown "NO ANSWER FOUND" than presented the wrong answer in use by the general public. GDQA returns 632 outputs to 156 queries; the MRR for these queries was 0.537.

Considering these characteristics, we infer that GDQA is adequately competitive and promising.

6.2 MRR with each query category

Respective MRRs for queries in each category defined in 3.2 are shown in Tab. 6.

Table 6. MRR with each query category

	Type 1	Type 2	Type 3	Type 4
MRR	0.49	0.50	0.56	0.28

The MRR of Type 4 is much lower than the others, as we expected. In our classification method, no clue for the answer can be extracted from queries. Therefore, the number of answer candidates becomes big. Additionally, answer candidates include meaningless terms. New approaches to extract more information from queries in Type 4 must be established.

6.3 Response Time

The relation between the response time and the maximum number of paragraphs used in the phase of answer candidates extraction and the graph structure are shown in Fig. 5.

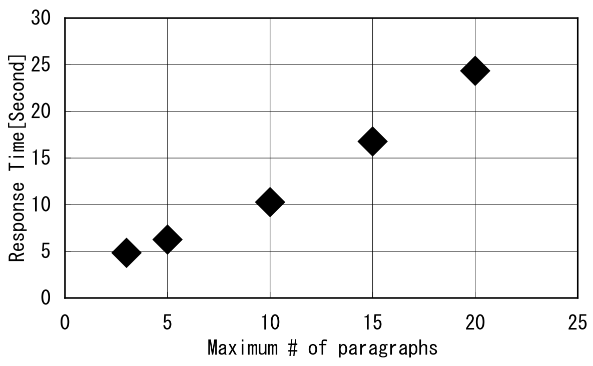


Figure 5. Response time and maximum number of paragraphs

We set the maximum number to 10. Therefore, it takes about 10 seconds to get an answer from GDQA. This is too long for use by the general public. However, for experimental use, it is not prohibitively long.

6.4 Advantage of the transformation

The result of 5.1 shows that the transformation to express the sentences with the active voice and passive voice in the same graph structure has a positive effect on the performance of GDQA despite the loss of information based on the transformation.

The same relation is expressed in various ways in the corpus. Equating various sentences and detecting important relations is extremely difficult. Dependency analysis has some effect to this problem. The result of 5.1 indicates that the transformation into an undirected graph has an additional effect.

7 Conclusion and future work

7.1 Conclusion

This paper introduced our Question Answering System, GDQA.

Our experiments demonstrated that GDQA is not sufficiently developed for use by the general public. However, our new algorithm, which uses a graph structure and search engine with GETA, is effective.

System performance differs according to the category of queries.

7.2 Future work

Future work to realize the Question Answering System is listed below.

More information from queries New techniques to extract more information from queries are necessary, especially from those in the Type 4 category in our framework.

Thesaurus If we can construct a thesaurus specialized to the retrieved documents, we can consolidate nodes with identical meanings.

Response Time A quicker response is essential to introduce a Question Answering System that is useful by the general public. We must produce a more efficient algorithm.

Retrieval INITIAL RETRIEVAL was very effective. We should improve REDUCED RETRIEVAL. Using a thesaurus, replacing the optional query terms with synonyms and setting their priority as high in REDUCED RETRIEVAL might facilitate retrieval .

References

[1] J. FUKUMOTO, T. KATO, and F. MASUI. “Question Answering Challenge(QAC-1) An Evaluation of Question Answering Task at NTCIR Workshop 3”. *Proceedings of the Third NTCIR Workshop*, 2003.

[2] J. FUKUMOTO, T. KATO, and F. MASUI”. “Question Answering Challenge for Five ranked answers and List answers – Overview of NTCIR4 QAC2 Subtask 1 and 2”. *Working Notes of the Fourth NTCIR Workshop Meeting*, 2004.

[3] IEICE. “The Frontier of QA System – Current Situation and Possibility of QA(in Japanese)”. In *Seminar of IEICE SIG-NLC*, 2003.

[4] IPA. *GETA*. http://geta.ex.nii.ac.jp/.

[5] S. Lee and G. G. Lee. “SiteQ/J: A Question Answering System for Japanese”. *Proceedings of the Third NTCIR Workshop*, 2003.

[6] M.A.Pasca and S.M.Harabagiu. “High Performance Question/Answering”. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 366–374, 2001.

[7] F. MASUI, S. SUZUKI, and J. FUKUMOTO. “Development of the named entity extraction tool: NExT(in Japanese)”. *Proceedings of the 8th annual conference of NLP*, pages 176–179, 3 2002.

[8] NamazuProject. *“Namazu”*. http://www.namazu.org/.

[9] NIST. *TREC*. http://trec.nist.gov/.

[10] M. NOMOTO, M. SATO, and H. SUZUKI. “NTCIR-3 QAC Experiments at Matsushita”. *Proceedings of the Third NTCIR Workshop*, 2003.

[11] RCIR/NII. *NTCIR*. http://research.nii.ac.jp/ntcir/index-en.html.

[12] 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(in Japanese)”. *Proceedings of IPSJ SIG-Fundamental Infology*, (No.064-12), 2001.

[13] T. TAKAHASHI, K. INUI, and Y. MATSUMOTO. “Methods for Estimating Syntactic Similarity(In Japanese)”. *IPSJ SIG-NL NL-150-7*, 2002.

[14] T. TAKAHASHI, K. NAWATA, and K. INUI. “Applying Structural Mathcing and Paraphrasing”. *Proceedings of the Third NTCIR Workshop*, 2003.

[15] T. TAKAKI and Y. ERIGUCHI. “NTT DATA Question-Answering Experiment at the NTCIR-3 QAC”. *Proceedings of the Third NTCIR Workshop*, 2003.

[16] E. M. Voorhees. “Overview of the TREC 2001 Question Answering Track”. *Proceedings of the Tenth Text REtrieval Conference*, 2001.