NAIST QA System for QAC2

Tetsuro Takahashi Kozo Nawata Kentaro Inui Yuji Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama, Ikoma, Nara, 630-0192, Japan
{tetsu-ta,kozo-n,inui,matsu}@is.aist-nara.ac.jp

Abstract

The system we presented for subtask1 and subtask2 in QAC2 is based on our previous one [12], which utilized a greedy answer seeking model using paraphrasing. We incorporated into the previous system a re-ranking model for matching questions and passages. In this model, we integrated a proximity-based scoring function with the original structural-based scoring function. Unfortunately, the result of evaluation showed that our proposed model did not work well. Based on error analysis, we conclude that structural matching-based approaches to answer seeking require technologies for the large-scale acquisition of paraphrase patterns. We are now investigating a variation of paraphrasing which is expected to be more helpful for question answering.

Keywords: *structural matching, paraphrasing, paraphrase space*

1 Introduction

An important issue in question answering is how to match an input question with a document or a passage that includes a candidate answer (hereafter referred to as a *passage*). Languages have redundancies, so that the same piece of information can often be linguistically realized as more than one expression. These redundancies make it hard to match questions and passages.

Paraphrasing is one approach to resolve this problem. If we have enough knowledge to allow paraphrasing to cover the redundancies, identification can be a simple task. Here is an example.

(1) *Q.* Who invented dynamite?

P. Alfred Nobel, the inventor of dynamite, was also a great industrialist.

In (1) , *Q* is a question and *P* is a matching passage. This question and passage cannot be matched exactly in their original form. If these expressions can be paraphrased as in (2) , they can be identified exactly.

(2) *Q'.* *X(NE:PERSON)* invented dynamite.

P'. Alfred Nobel invented dynamite. He was also a great industrialist.

In previous work, we proposed a greedy answer seeking model using paraphrasing [12]. The current system is based on this algorithm. The system also incorporates a re-ranking model for matching a question and a passage.

In QAC2 [5] we participated in subtask1 and subtask2. We describe an overview of the system, the results and discussion in the following sections.

2 System Overview

An overview of the system is shown in Figure 1. In the current system, the overall question answering process has three steps: 1) question analysis, 2) passage retrieval and 3) answer selection. We describe preprocessing first and then the above three steps.

2.1 Preprocessing

Questions in QAC2 are factoid questions. The required answers are short answers consisting of a noun or noun phrase. The answers are basically represented as named entities (hereafter NE) in source texts. Furthermore, keywords in a question are NE in most cases. Hence, NE tagging plays a very important role in finding an answer. We utilized an NE tagger Bar [1] to annotate the documents with NE tags. Bar annotates using the eight types of NE tags defined in IREX [7]. The tagging F-measure is about 87% for newspaper text.

Our question answering system paraphrases both questions and passages using a lexico-structural paraphrase engine KURA [11]. Since KURA requires the dependency structure of a sentence as its input, input questions and passages have to be parsed into their dependency structures. For sentence parsing, we use CaboCha [10]. We use dependency structures, with *bunsetsu*-phrasal units as the nodes, to represent parsed questions and passages.

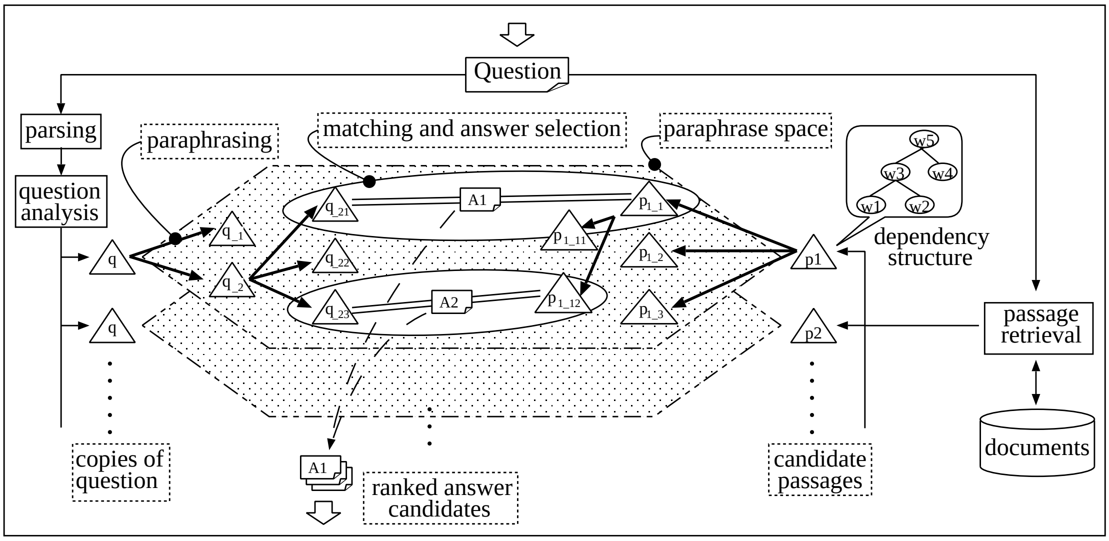

Figure 1. System overview

All of the sentences in the corpus used for the QAC2 task were parsed into their dependency structures and tagged with NE tags before the formal run to conduct entire process in practical time.

2.2 Question analysis

The system first analyzes an input question. In our system, an input question is first paraphrased into a regularized expression for the purpose of question analysis. The regularized expression contains a word variable which is to be matched with the answer. The knowledge for question analysis is implemented as paraphrasing patterns. We implemented about 130 paraphrasing patterns for the current system. The following is an example of paraphrasing for question analysis.

(3) *S.* “クローン羊のドリーが誕生したのはいつですか。“
(When was the cloned sheep Dolly born?)

T. “クローン羊のドリーは *X(NE:DATE)* 誕生した。”
(The cloned sheep Dolly was born on *X(NE:DATE)*)

S is the original question and *T* is the paraphrased question. These paraphrases are generated by paraphrasing patterns such as (4)

(4) a. いつ → *X(NE:DATE)*
(when → *X(NE:DATE)*)

b. *VP* するのは *X(NE:DATE)* だ → *X(NE:DATE) VP* する
(the day *VP* is *X(NE:DATE)* → *VP* on *X(NE:DATE)*)

2.3 Passage retrieval

For passage retrieval, the system first submits the set of keywords contained in a given question to the IR tool [14] to retrieve the 20-best documents. The passage retrieval module then summarizes the 20 retrieved documents, and produces a set of passages. A passage is a sequence of sentences selected according to the following factors: question keywords, answer type, NE tags and the proximity of the sentences. In the current system, the length of passages is limited to five sentences. The system also calculates the score of each passage in order to rank them, and the 10-best passages are then passed on to the answer selection module.

2.4 Answer selection

2.4.1 Matching

The roles of matching a question with a passage are 1) to give a score to every word in a passage and 2) to calculate the similarity between the question and passage. 1) is necessary for selection of answer candidates, and 2) is necessary for greedy answer seeking and ranking of answer candidates.

Since the system must extract only one set of answers from the documents, the similarity is especially important for subtask2 in QAC2 . The current system depends completely on the similarity measure to detect the relevance threshold.

Our previous system used only structural matching based on the Tree Kernel [2] for matching between a question and a passage. However structural matching is too strict for matching between questions and passages because of variation in natural language expressions. We therefore extended the matching in two directions. First, we integrated a proximity-based scoring function with the structural-based scoring function. The system first calculates similarity using a proximity-based scoring function. The function gives a score to every *bunsetsu*-phrase in a passage based on question keywords, the answer type, NE tags and the proximity of the *bunsetsu*-phrases in *bunsetsu*-phrase sequences. This score is then multiplied by the structural similarity score (0〜1) calculated by a structural-based scoring function. In other words, our system verifies and re-ranks answer candidates using structural information. The system calculates the score of the ith *bunsetsu*-phrase in a passage ($Score_i$) using scoring function (1).

$$Score_i = ((W + P) \times C_w + S) \times C_s \qquad (1)$$

where W is a keyword matching score, P is a proximity score, S is a structural matching score, and C_w and C_s are both confidence measure. C_w measures the amount of keywords in the original question that are matched in the passage. Similarly C_s is used to evaluate the amount of the original question's dependency structure that is preserved in the passage. The function (1) attaches a great deal of importance to the value of structural confidence. We discuss issues concerning variations of the function in Section 4.2.

Second, we made the structural matching looser. Generally, it seems to be rare that the structure of a question sentence matches that of passages. The current system produces a bag of bigrams on a dependency tree. This can be thought of as a loose approximation of strict structural matching.

The system returns the top five answer candidates for subtask1, and answer candidates which have a higher score than a given threshold for subtask2. The parameters for the matching and the threshold were tuned manually using the QAC1 data [4].

2.4.2 Greedy answer seeking using paraphrasing

We previously proposed an answer seeking algorithm for question answering that integrates matching and paraphrasing [12]. In this method, paraphrasing is responsible for making matching more exact. Matching and paraphrasing are repeated until the improvement in the matching score levels off. The best matching pair and the corresponding answer candidate string are then returned. In Figure 1, the system generates a paraphrase space between q and p to seek better matches. Here the paraphrase space is a search space consisting of paraphrases generated from questions and passages. Since it can be intractably large, we restrict the paraphrase generation in a greedy search-like manner.

The current system also utilizes this algorithm. Knowledge for paraphrasing is basically the same as in our previous system.

3 Results

3.1 Passage retrieval

We first examined the accuracy of the passage retrieval module. The system retrieved passages which contains a correct answer for 163 questions out of 197 questions in subtask1. This means that the system failed to find an answer for 34 (17.0%) questions. The accuracy is variable by the parameter which sets the number of passages. We need to set this at an appropriate value.

3.2 Answer seeking

The results of the overall question answering task are shown in Table 1 for subtask1 and Table 2 for subtask2. We conducted experiments on four types of model which are combinations of two sets of alternatives: with (+) or without (−) re-ranking using structural information (re-ranking), and with or without greedy answer seeking using paraphrasing (paraphrasing). We compared these results to analyze the effects of the re-ranking model and paraphrasing. The values are MRR in Table 1 and mean F-value in Table 2

Table 1. MRR in subtask1

	− re-ranking	+ re-ranking
− paraphrasing	0.340	0.311
+ paraphrasing	0.341	0.310

Table 2. Mean F-value in subtask2

	− re-ranking	+ re-ranking
− paraphrasing	0.219	0.185
+ paraphrasing	0.220	0.185

4 Discussion

4.1 Effects of re-ranking by structural matching

As Table 1 and Table 2 show, re-ranking using structural matching had a negative rather than a positive effect. The main reason why re-ranking did not work is scattered keywords. Question keywords often appear in positions syntactically isolated from the answer. Furthermore they tended to be scattered beyond sentence boundaries. In such cases, without deeper analysis of discourse including coreference resolution, structural matching is ineffective. We have shown previously the importance of coreference resolution in question answering [13]. For 41% of questions in QAC1, matching passages contain more than

Table 3. Experimental results using various scoring functions

scoring function	s1	s2	s2’
(2)	**0.356**	**0.211**	0.211
(3)	0.351	0.210	0.154
(4)	0.353	**0.211**	**0.212**
(5)	0.296	0.167	0.108

one coreference which has to be resolved in order to match with the question sentence exactly.

Even in cases in which there is no coreference, structural information did not work in most cases. We expected that paraphrasing would help structural matching, however, the size of our current paraphrasing knowledge was too small to do so. We discuss the effects of paraphrasing in Section 4.3.

4.2 Variations of scoring function

The system calculated the score of a *bunsetsu*-phrase using the function (1). As previously stated, the function may attache a great deal of importance to the value of structural confidence to excess. We then conducted experiments using various scoring functions as follows.

$$Score_i = (W + P) \times C_w \quad (2)$$

$$Score_i = (W + P) \times C_w + S \quad (3)$$

$$Score_i = (W + P) \times C_w + S \times C_s \quad (4)$$

$$Score_i = ((W + P) \times C_w + S) \times C_s \quad (5)$$

The function (2) corresponds to “without re-ranking” and the function (5) corresponds to “with re-ranking” denoted in Section 3.2.

The results of experiments are shown in Table 3. Column s1 and s2 show the results for subtask1 and subtask2 respectively. Column s2’ shows the result for subtask2 using the modified Tree Kernel based scoring measure for structural matching which we proposed in our previous system [12]. Each value in Table 3 is MRR for subtask1 and mean F-value for subtask2. We manually tuned some system parameters for the experiments in order to concentrate variation of scoring functions. This caused slight differences between the result in Table 3 and that in Table 1 and Table 2.

The function (2) gave the top performance for subtask1. This means that structural matching could not bring desired effect for subtask1. The result of s2’ using the function (4) gave slightly higer mean F-value than that using the function (2). This shows availability of structural matching.

4.3 Effects of paraphrasing

On comparison between with $(+)$ paraphrasing and without $(-)$ paraphrasing in Table 1 and Table 2, it becomes clear that the effects of paraphrasing are extremely small.

For 200 questions there were 2000 pairs of question and passage. The system generated 7829 (3.91 on average) paraphrases. Of those 7829 paraphrases, 6668 paraphrases were of passages and 1161 paraphrases were of questions. The paraphrases of questions did not include paraphrases produced by the question analysis module which regularizes from interrogative sentences to affirmative sentences.

Greedy answer seeking repeated 1.08 times on average. This does not mean that the similarity between a question and a passage levels off, but that there is often insufficient paraphrasing knowledge to gain a matching score. The number of paraphrasing rules which were used to gain a matching score was 592. In other words the system generated effective paraphrases 0.296 times on average for each pair.

5 Related work

Hermjakob et al. [6] and Dumais et al. [3] report that using paraphrase patterns resulted in considerable improvements when using the web as an information source, but did not work effectively when the information source was limited to a closed document collection. When resources are limited such as in QAC2, large scale paraphrasing knowledge is required.

Ittycheriah et al. [8] and Kiyota et al. [9] used syntactic structure information as a score to be appended. In our approach, however, this information was used as a penalty. The penalty was too strict, since question key words appeared in positions isolated from the answer in many passages.

6 Conclusion

The characteristics of our question answering system are 1) a re-ranking model using structural information and 2) greedy matching using paraphrasing. Unfortunately, the result of evaluation shows that the re-ranking model did not work. It could be used as an approach which avoids using structural information for matching between a question and a passage, however, an answer found by shallow information such as proximity of keywords does not have a reasonable level of certainty. We must therefore investigate the effective use of structural information. Though paraphrasing seemed to be a solution for the problem, the result also shows that the knowledge used for paraphrasing was still insufficient.

We are now investigating a variation of paraphrasing which is expected to be more helpful for question answering.

7 Acknowledgments

We thank Satoshi Sekine (New York University) for allowing us to use his named entity ontology and dictionary, and Masao Utiyama (Communications Research Laboratory) for his IR engine ruby-ir. We also thank the NTT Communication Science Laboratories for their case frame dictionary and thesaurus, which we used for paraphrase generation.

References

[1] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *HLT-NAACL*, 2003.

[2] M. Collins and N. Duffy. Convolution kernels for natural language. In *Neural Information Processing Systems (NIPS)*, 2001.

[3] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, 2002.

[4] J. Fukumoto, T. Kato, and F. Masui. Question answering challenge (qac1): Question answering evaluation at NTCIR workshop 3. In *Working Notes of the Third NTCIR Workshop Meeting: QAC1*, 2002.

[5] J. Fukumoto, T. Kato, and F. Masui. Question answering challenge for five ranked answers and list answers - an overview of ntcir4 qac2 subtask 1 and 2. In *Working Notes of the Third NTCIR Workshop Meeting: QAC2*, 2004.

[6] U. Hermjakob, A. Echibahi, and D. Marcu. Natural language based reformulation resource and web exploration for question answering. In *the 2002 edition of the Text REtrieval Conference (TREC)*, 2002.

[7] e. IREX Committee. In *IREX workshop*, 1999.

[8] A. Ittycheriah, M. Franz, and S. Roukos. IBM's statistical question answering system–trec-10. In *the 2001 edition of the Text REtrieval Conference (TREC)*, page 258, 2001.

[9] Y. Kiyota, S. Kurohashi, and F. Kido. "dialog navigator" : A questions answering system based on large text knowledge base. In *The 19th International Conference on Computational Linguistics (COLING 2002)*, 2002.

[10] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[11] T. Takahashi, T. Iwakura, R. Iida, A. Fujita, and K. Inui. KURA: A revision-based lexico-structural paraphrasing engine. In *The Natural Language Processing Pacific Rim Symposium (NLPRS-2001) Workshop on Automatic Paraphrasing: Theories and Applications*, 2001.

[12] T. Takahashi, K. Nawata, S. Kouda, and K. Inui. Seeking answers by structural matching and paraphrasing. In *Working Notes of the Third NTCIR Workshop Meeting: QAC1*, 2002.

[13] T. Takahashi and S. Sekine. Analysis of effects of paraphrasing in question answering (in Japanese). In *Information Processing Society of Japan NL-157*, pages 99–106, 2003.

[14] M. Utiyama and H. Isahara. Tools for exploring natural language. In *The Natural Language Processing Pacific Rim Symposium (NLPRS-2001)*, pages 779–780, 2001.