NAIST QA System for QAC2

TAKAHASHI Tetsuro NAWATA Kozo INUI Kentaro MATSUMOTO Yuji
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 the subtask1 and subtask2 in QAC2 is based on our previous one [12], which utilized a greedy answer seeking model using paraphrasing. We incorporate a re-ranking model for matching questions and passages into the previous system. In the model, we integrated a proximity-based scoring function with the structural-based scoring function. Unfortunately, the result of evaluation shows 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 large-scale acquisition of paraphrase patterns. We are now investigating a variation of paraphrasing which is expected to be 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 an answer candidate (hereafter referred to as *passage*). Languages have redundancies, so that the same piece of information can often be linguistically realized in more than one expression. The redundancies make it hard to match questions and passages.

Paraphrasing is one approach to resolve this problem. If we have enough knowledge for 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 the 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 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 can annotate 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 and translated 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.

All of the sentences in the corpus which are used for the QAC2 task were parsed into the dependency

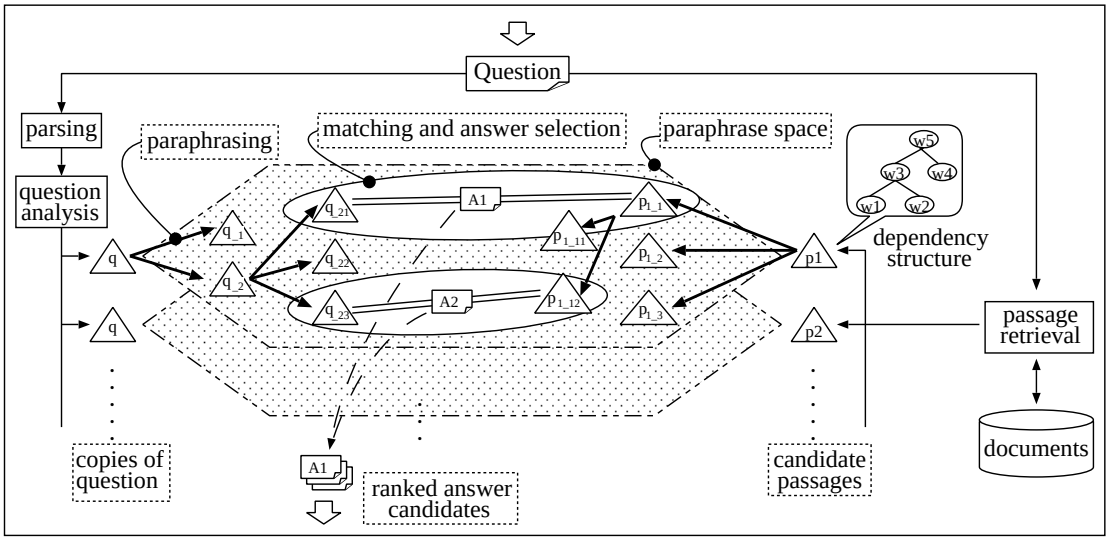

Figure 1. System overview

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 variable word which is to be matched with the answer. The knowledge for question analysis is implemented as paraphrasing patterns. We implemented about 100 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 paraphrasing are generated by paraphrasing patterns as in (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 system then summarizes the 20 retrieved documents, and produces a set of passages. The passage is a sequence of sentences which is selected according to the factors that take into account question keywords, the answer type, NE tags and their proximity. In the current system, the length of passages is limited by five sentences. The system also calculates the score of each passage to rank them, and the 10-best passages are used for the answer selection module.

2.4 Answer selection

2.4.1 Matching

The roles of matching a question and 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. For subtask2 in QAC2 the similarity is especially important. In subtask2, the system must extract only one set of answers from the documents. 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 a question with a passage. However the structural matching is too strict for matching questions and passages because of variation in natural language expression. 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 score to every word in a passage according to the factors that take into account question keywords, the answer type, NE tags and their proximity. The 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. Second, we made the structural matching looser. Generally, it seems to be rare that a structure of a question sentence matches with that of passages. The current system produces a bag of bigram. This can be thought of as a loose approximation of strict structural matching.

The system returns top five answer candidates for subtask1, answer candidates which have higher score than a 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

The results 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 and with or without 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, structural matching is ineffective, without deeper analysis of discourse including coreference resolution. We have shown previously the importance of coreference resolution in question answering [13]. For 41 % of questions in QAC1, matching passages contain more than one coreference which has to be resolved 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 the next section.

4.2 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 for regularization of interrogative sentences. This type of paraphrase was conducted by the question analysis module.

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 are few 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) re-ranking model using structural information and 2) greedy matching using paraphrasing. Unfortunately, the result of evaluation shows that re-ranking model did not work. It can be an approach to use structural information as a score to be appended, however, we would like to stick to use of structural information for verification. The result shows that if we used structural information to give score to answer candidates, in many cases it was useless. In such cases paraphrasing can be a solution. However the result also shows that knowledge for paraphrasing was still too small.

We are now investigating a variation of paraphrasing which is expected to be 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] IREX Committee, editor. 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.