Evaluating Question-Answering Techniques in Chinese

Xiaoyan Li and W. Bruce Croft

Computer Science Department

University of Massachusetts, Amherst, MA

{xiaoyan, croft}@cs.umass.edu

ABSTRACT

An important first step in developing a cross-lingual question answering system is to understand whether techniques developed with English text will also work with other languages, such as Chinese. The Marsha Chinese question answering system described in this paper uses techniques similar to those used in the English systems developed for TREC. Marsha consists of three main components: the query processing module, the Hanquery search engine, and the answer extraction module. It also contains some specific techniques dealing with Chinese language characteristics, such as word segmentation and ordinals processing. Evaluation of the system is done using a method based on the TREC question-answering track. The results of the evaluation show that the performance of Marsha is comparable to some English question answering systems in TREC 8 track. An English language version of Marsha further indicates that the heuristics used are applicable to the English question answering task.

Keywords

Question-Answering (QA); Search engine; multilingual retrieval, Chinese QA.

1. Introduction

A number of techniques for "question answering" have recently been evaluated both in the TREC environment (Voorhees and Harman, 1999) and in the DARPA TIDES program. In the standard approach to information retrieval, relevant text documents are retrieved in response to a query. The parts of those documents that may contain the most useful information or even the actual answer to the query are typically indicated by highlighting occurrences of query words in the text. In contrast, the task of a question-answering system is to identify text passages containing the relevant information and, if possible, extract the actual answer to the query. Question answering has a long history in natural language processing, and Salton's first book (Salton, 1968) contains a detailed discussion of the relationship between information retrieval and question-answering systems. The focus in recent research has been on extracting answers from very large text databases and many of the techniques use search technology as a major component. A significant number of the queries used in information retrieval experiments are questions, for example, TREC topic 338 "What adverse effects have people experienced while taking aspirin repeatedly?" and topic 308 "What are the advantages and/or disadvantages of tooth implants?" In question-answering experiments, the queries tend to be more restricted questions, where answers are likely to be found in a single text passage, for example, TREC question-answering question 11 "Who was President Cleveland's wife?" and question 14 "What country is the biggest producer of Tungsten?"

The TREC question-answering experiments have, to date, used only English text. As the first step towards our goal of cross-lingual question answering, we investigated whether the general approaches to question answering that have been used in English will also be effective for Chinese. Although it is now well known that statistical information

retrieval techniques are effective in many languages, earlier research, such as Fujii and Croft (1993, 1999), was helpful in pointing out which techniques were particularly useful for languages like Japanese. This research was designed to provide similar information for question answering. In the next section, we describe the components of the Chinese question answering system (Marsha) and the algorithm used to determine answers. In section 3, we describe an evaluation of the system using queries obtained from Chinese students and the TREC-9 Chinese cross-lingual database (164,779 documents from the Peoples Daily and the Xing-Hua news agencies in the period 1991-1995).

2. Overview of the Marsha Question Answering System

The Chinese question-answering system consists of three main components. These are the query processing module, the Hanquery search engine, and the answer extraction module. The query processing module recognizes known question types and formulates queries for the search engine. The search engine retrieves candidate texts from a large database. The answer extraction module identifies text passages that are likely to contain answers and extracts answers, if possible, from these passages. This system architecture is very similar to other question-answering systems described in the literature.

More specifically, the query processing module carries out the following steps:

(1) The query is matched with templates to decide the question type and the "question words" in the query. We define 9 question types. Most of these correspond to typical named entity classes used in information extraction systems. For each question type, there are one or more templates. Currently there are 170 templates. If more than one template matches the question, we pick the longest match. For example, a question may include "多少元" (how many dollars). Then both 多少元 (how many dollars) and 多少 (how many) will match the question. In this case, we will pick 多少元 and assign "MONEY" to the question type.

The following table gives examples for each question type:

TEMPLATE	QUESTION TYPE	TRANSLATION
哪个人	PERSON	which person
哪个城市	LOCATION	which city
什么组织	ORGANIZATION	what organization
哪一年哪一月哪一天	DATE	what date
什么时间	TIME	what time
多少元	MONEY	how many dollars
百分比是什么	PERCENTAGE	what is the percentage
多少	NUMBER	how many
什么意思	OTHER	what is the meaning of

(2) Question words are removed from the query. This is a form of "stop word" removal. Words like "哪个人" (which person) are removed from the query since they are unlikely to occur in relevant text.

(3) Named entities in the query are marked up using BBN's IdentiFinder system. A named entity is kept as a word after segmentation.

(5) The query is segmented to identify Chinese words.

(6) Stop words are removed.

(7) The query is formulated for the Hanquery search engine. Hanquery is the Chinese version of Inquery (Broglio, Callan and Croft, 1996) and uses the Inquery query language that supports the specification of a variety of evidence combination methods. To support question answering, documents containing most of the query words were strongly preferred. If the number of query words left after the previous steps is greater than 4, then the operator #and (a probabilistic AND) is used. Otherwise, the probabilistic passage operator #UWn (unordered window) is used. The parameter n is set to twice the number of words in the query.

Hanquery is used to retrieve the top 10 ranked documents. The answer extraction module then goes through the following steps:

(8) IdentiFinder is used to mark up named entities in the documents.

(9) Passages are constructed from document sentences. We used passages based on sentence pairs, with a 1-sentence overlap.

(10) Scores are calculated for each passage. The score is based on five heuristics:

· *First Rule:*

Assign 0 to a passage if no expected name entity is present.

· *Second Rule:*

Calculate the number of match words in a passage.

Assign 0 to the passage if the number of matching words is less than the threshold. Otherwise, the score of this passage is equal to the number of matching words (*count_m*).

The threshold is defined as follows:

threshold = *count_q* if *count_q*<4

threshold = *count_q*/2.0+1.0 if 4<=*count_q*<=8

threshold = *count_q*/3.0+2.0 if *count_q*>8

count_q is the number of words in the query.

· *Third Rule:*

Add 0.5 to score if all matching words are within one sentence.

· *Fourth Rule:*

Add 0.5 to score if all matching words are in the same order as they are in the original question.

· *Fifth Rule:*

score = score + *count_m*/(size of matching window)

(11) Pick the best passage for each document and rank them.

(12) Extract the answer from the top passage:

Find all candidates according to the question type. For example, if the question type is LOCATION, then each location marked by IdentiFinder is an answer candidate. An answer candidate is removed if it appears in the original question. If no candidate answer is found, no answer is returned.

Calculate the average distance between an answer candidate and the location of each matching word in the passage. Pick the answer candidate that has the smallest average distance as the final answer.

3. Evaluating the System

We used 51 queries to do the initial evaluation of the question-answering system. We selected 26 queries from 240 questions collected from Chinese students in our department, because only these had answers in the test collection. The other 25 queries were constructed by either reformulating a question or asking a slightly different question. For example, given the question “which city is the biggest city in China?” we also generated the questions “where is the biggest city in China?” and “which city is the biggest city in the world?”.

The results for these queries were evaluated in a similar, but not identical way to the TREC question-answering track. An “answer” in this system corresponds to the 50 byte responses in TREC and passages are approximately equivalent to the 250 byte TREC responses.

For 33 of 51 queries, the system suggested answers. 24 of the 33 were correct. For these 24, the “reciprocal rank” is 1, since only the top ranked passage is used to extract answers. Restricting the answer extraction to the top ranked passage also means that the other 27 queries have reciprocal rank values of 0. In TREC, the reciprocal ranks are calculated using the highest rank of the correct answer (up to 5). In our case, using only the top passage means that the mean reciprocal rank of 0.47 is a lower bound for the result of the 50 byte task.

As an example, the question “哪个城市是中国最大的城市” (Which city is the biggest city in China?), the answer returned is 上海 (Shanghai). In the top ranked passage, “China” and “Shanghai” are the two answer candidates that have the smallest distances. “Shanghai” is chosen as the final answer since “China” appears in the original question.

As an example of an incorrect response, the question “谢军在哪一年战胜了前苏联选手第一次获得国际象棋世界冠军” (In which year did Jun Xie defeat a Russian player and win the world chess championship for the first time?) produced an answer of 今天 (today). There were two candidate answers in the top passage, “October 18” and “today”. Both were marked as DATE by Identifinder, but “today” was closer to the matching words. This indicates the need for more date normalization and better entity classification in the system.

For 44 queries, the correct answer was found in the top-ranked passage. Even if the other queries are given a

reciprocal rank of 0, this gives a mean reciprocal rank of 0.86 for a task similar to the 250 byte TREC task. In fact, the correct answer for 4 other queries was found in the top 5 passages, so the mean reciprocal rank would be somewhat higher. For 2 of the remaining 3 queries, Hanquery did not retrieve a document in the top 10 that contained an answer, so answer extraction could not work.

4. Further Improvements

These results, although preliminary, are promising. We have made a number of improvements in the new version (v2) of the system. Some of these are described in this section.

One of the changes is designed to improve the system's ability to extract answers for the questions that ask for a number. A number recognizer was developed to recognize numbers in Chinese documents. The numbers here are numbers other than DATE, MONEY and PERCENTAGE that are recognized by IdentiFinder. The version of IdentiFinder used in our system can only mark up seven types of name entities and this limits the system's ability to answer other types of questions. The number recognizer is the first example of the type of refinement to named entity recognition that must be done for better performance.

An example of a question requiring a numeric answer is:

"克林顿是第几任美国总统? (What is the number of Clinton's presidency?)". This question could be answered in Marsha v2 by extracting the marked up number from the best passage in the answer extraction part, while Marsha v1 could only return the top 5 passages that were likely to have the answer to this question.

Another improvement relates to the best matching window of a passage. The size of the matching window in each passage is an important part of calculating the belief score for the passage. Locating the best matching window is also important in the answer-extraction processing because the final answer picked is the candidate that has the smallest average distance from the matching window. The best matching window of a passage here is the window that has the most query words in it and has the smallest window size. In the previous version of our system, we only consider the first occurrence of each query word in a passage and index the position accordingly. The matching window is thus from the word of the smallest index to the word of the largest index in the passage. It is only a rough approximation of the best matching window though it works well for many of the passages. In the second version of Marsha, we developed a more accurate algorithm to locate the best matching window of each passage. This change helped Marsha v2 find correct answers for some questions that previously failed. The following is an example of such a question.

For the question "美国贫困线以下的人口总数是多少? (How many people in the United States are below the poverty line?)"

The best passage is as follows:

"本报华盛顿９月２８日电记者张启昕报道：由于经济衰退，人民收入下降，*美国穷*人去年一年增加２００多*万*，使*美国*生活在政府规定的*贫困线以下*的*人口总数*达３３５８．５万，比１９８９年增加６．７％，这个数字还不包括大批流落街头的无家可归者"

This passage has two occurrences of query word "美国". In v1, the first occurrence of "美国" is treated as the start of the matching window, whereas the second occurrence is actually the start of the best matching window. There are two numbers "２００多万" (more than 2 million) and "３３５８．５万" (33.585 million) in the passage. The right answer "３３５８．５万" (33.585 million) is nearer to the best matching window and "２００多万" (more than 2 million) is nearer to the estimated matching window. Therefore, the right answer can be extracted after correctly locating the best matching window.

The third improvement is with the scoring strategies of passages. Based on the observation that the size of the best matching window of a passage plays a more important role than the order of the query words in a passage, we adjusted the score bonus for same order satisfaction from 0.5 to 0.05. This adjustment makes a passage with a smaller matching window get a higher belief score than a passage that satisfies the same order of query words but has a bigger matching window. As an example, consider the question:

"谁是第一个美国总统? (Who was the first president in the United States?)".

Passage 1 is the passage that has the right answer "乔治华盛顿".

Passage 1.

"１９９２年１２月２６日 星期六#pn:第七版#pm:国际副刊#xh:5#lm:世界一角#ti:美国总统的就职典礼#au:允文#rw:美国第一任总统乔治华盛顿#rw:比尔克林顿#rw:托马斯杰弗逊"

Passage 2.

“德国总理科尔二十五日下午离开波恩前往华盛顿进行为期一天的访问, 这将是他第一次会见美国总统克林顿”

Passage 1 and Passage 2 both have all query words. The size of the best matching window in Passage 1 is smaller than that in Passage 2 while query words in Passage 2 have the same order as that in the question. The scoring strategy in Marsha v2 selects Passage 1 and extracts the correct answer while Marsha v1 selected Passage 2.

Special processing of ordinals has also been considered in Marsha v2. Ordinals in Chinese usually start with the Chinese character "第" and are followed by a cardinal. It is better to retain ordinals as single words during the query generation in order to retrieve better relevant documents. However, the cardinals (part of the ordinals in Chinese) in a passage are marked up by the number recognizer for they might be answer candidates for questions asking for a number. Thus ordinals in Chinese need special care in a QA system. In Marsha v2, ordinals appearing in a question are first retained as single words for the purpose of generating a good query and then separated in the post processing after relevant documents are retrieved to avoid answer candidates being ignored.

5. Comparison with English Question Answering Systems

Some techniques used in Marsha are similar to the techniques in English question answering systems developed by other researchers. The template matching in Marsha for deciding the type of expected answer for a question is basically the same as the one used in the GuruQA (Prager et al., 2000) except that the templates consist of Chinese word patterns instead of English word patterns. Marsha has the ability of providing answers to eight types of questions: PERSON, LOCATION, ORGANIZATION, DATE, TIME, MONEY, PERCENTAGE, and NUMBER. The first seven types correspond to the named entities from IdentiFinder developed by BBN. We developed a Chinese number-recognizer ourselves which marks up numbers in the passages as answer candidates for questions asking for a number. The number could be represented as a digit number or Chinese characters. David A. Hull used a proper name tagger ThingFinder developed at Xerox in his question answering system. Five of the answer types correspond to the types of proper names from ThingFinder (Hull, 1999). The scoring strategy in Marsha is similar to the computation of score for an answer window in the LASSO QA system (Moldovan et al., 1999) in terms of the factors considered in the computation. Factors such as the number of matching words in the passage, whether all matching words in the same sentence, and whether the matching words in the passage have the same order as they are in the question are common to LASSO and Marsha.

We have also implemented an English language version of Marsha. The system implements the answer classes PERSON, ORGANIZATION, LOCATION, and DATE. Queries are generated in the same fashion as Marsha. If there are any phrases in the input query (named entities from IdentiFinder, quoted strings) these are added to an Inquery query in a #N operator all inside a #sum operator. For example:

> Question: "Who is the author of "Bad Bad Leroy Brown"
>
> Inquery query: #sum(#uw8(author Bad Bad Leroy Brown) #6(Bad Bad Leroy Brown))

Where N is number of terms + 1 for named entities, and number of terms + 2 for quoted phrases. If a query retrieves no documents, a “back off” query uses #sum over the query terms, with phrases dropped. The above would become #sum(author Bad Bad Leroy Brown).

The system was tested against the TREC9 question answering evaluation questions. The mean reciprocal rank over 682/693 questions was 0.300 with 396 questions going unanswered. The U.Mass. TREC9 (250 byte) run had a score of 0.367. Considering only the document retrieval, we find a document containing an answer for 471 of the questions, compared to 477 for the official TREC9 run which used expanded queries. This indicates that the Marsha heuristics have applicability to the English question answering task and are not limited to the Chinese question answering task.

6. Summary and Future Work

The evaluations on Marsha, although preliminary, indicate that techniques developed for question answering in English are also effective in Chinese. In future research, we plan to continue to improve these techniques and carry out more careful evaluations to establish whether there are any significant differences in the question-answering task between these two languages.

The evaluation of the English version of Marsha indicates that the Marsha heuristics work well in English as well as in Chinese. We now plan to incorporate these techniques in a cross-lingual question-answering system for English and Chinese. By using two systems with similar question processing strategies, we hope to exploit the query templates to produce accurate question translations.

We have also started to develop a probabilistic model of question answering using the language model approach (Ponte and Croft, 1998). This type of model will be essential for extending the capability of QA systems beyond a few common query forms.

Acknowledgements

This material is based on work supported in part by the Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623 and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912.

Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

We also want to express out thanks to people at CIIR for their help. Special thanks to David Fisher who implemented the English language version of Marsha, and Fangfang Feng for his valuable discussions on Chinese related research issues.

7. References

Broglio, J., Callan, J.P. and Croft, W.B. "Technical Issues in Building an Information Retrieval System for Chinese," CIIR Technical Report IR-86, Computer Science Department, University of Massachusetts, Amherst, (1996).

H. Fujii and W.B. Croft, "A Comparison of Indexing Techniques for Japanese Text Retrieval," Proceedings of SIGIR 93, 237-246, (1993).

H. Fujii and W.B. Croft, "Comparing the performance of English and Japanese text databases", in S. Armstrong et al (eds.), *Natural Language Processing using Very Large Corpora*, 269-282, Kluwer, (1999). (This paper first appeared in a 1994 workshop)

G. Salton, Automatic Information Organization and Retrieval, McGraw-Hill, (1968).

E. Voorhees and D. Harman (eds.), *The 7th Text Retrieval Conference (TREC-7)*, NIST Special Publication 500-242, (1999).

Ponte, J. and Croft, W.B. "A Language Modeling Approach to Information Retrieval," in the Proceedings of SIGIR 98, pp. 275-281(1998).

Moldovan, Dan et al, "LASSO: A Tool for Surfing the Answer Net," in the proceedings of TREC-8, pp 175-183. (1999).

Hull, David A., "Xerox TREC-8 Questio Answering Track Report," in the proceedings of TREC-8, pp743.

Prager, John, Brown, Eric, and Coden, Anni, "Question_Answering by Predictive Annotation," in the proceedings of SIGIR 2000.