FDUQA on TREC2005 QA Track

Lide Wu, Xuanjing Huang, Yaqian Zhou, Zhushuo Zhang, Fen Lin
Fudan University, Shanghai, China, 200433
{ldwu,xjhuang,zhouyaqian,zs_zhang,fenglin}@fudan.edu.cn

1. Introduction

In this year's QA Track, we participant in the main and document ranking task and do not take part in the relation task. We put the most effort in factoid and definition questions, and very little on list questions and document ranking task.

For factoid questions, we use three QA systems: system 1, system 2 and system 3. System 1 is very similar to our last year's system [Wu et al, 2004] except two main modifications. One is adding an answer validation-feedback scheme. The other is an improved answer projection module. System 2 is a classic QA system that does not use Web. System 3 is a pattern-based system that we used in TREC 2002 evaluation. The main contribution for factoid question is two improvements for Web-based QA system and the system combination.

For definition question, we attempt to utilize both the existing definitions in the Web knowledge bases and the automatically generated structured patterns. Effective methods are adopted to make full use of these resources, and they promise high quality response to definition questions.

For list questions, we use a pattern-based method to find more answers other than those found in the processing of factoid question.

For document ranking task, we only collect the outputs from document searching or answer projection module.

In the following, Section 2, 3, 4 will describe our algorithms to factoid, list and definition questions separately. Section 5 will present our results in TREC 2005.

2. Factoid Question

In order to answer factoid questions, we use three QA systems: system 1, system 2 and system 3. System 1 is very similar to our last year's system [Wu et al, 2004] except two main modifications. One is adding an answer validation-feedback scheme. The other is an improved answer projection module. System 2 is a classic QA system that does not use Web. System 3 is a pattern-based system that we used in TREC 2002 evaluation.

Table 1 illustrates the experimental results for these systems that test on TREC 2004's question set. FDUQA13 is the old system for TREC 2004, which is the baseline for system 1. The Combination represents the combined system of system 1, 2, 3. Obviously, the combined system achieves the best performance. System1 is best among the single systems and also

better than it baseline, FDUQA13.

In the following sections, we will describe our best single system, the Web-base QA system and the combined methods.

	Correct #	Precision (%)
FDUQA13	59	25.7
system 1	90	39.1
system 2	36	15.6
system 3	42	18.3
Combination	100	43.5

Table 1 factoid QA results on TREC 2004 question set

2.1 Web-based system

Figure 1 describes the process of our Web-based factoid question answering. This system bases on our last year's system. The main improvements include: adding an answer validation-feedback scheme, and improving the answer projection module. The next sections will describe the two improvements, and the other modules can be found in [Wu et al, 2004].

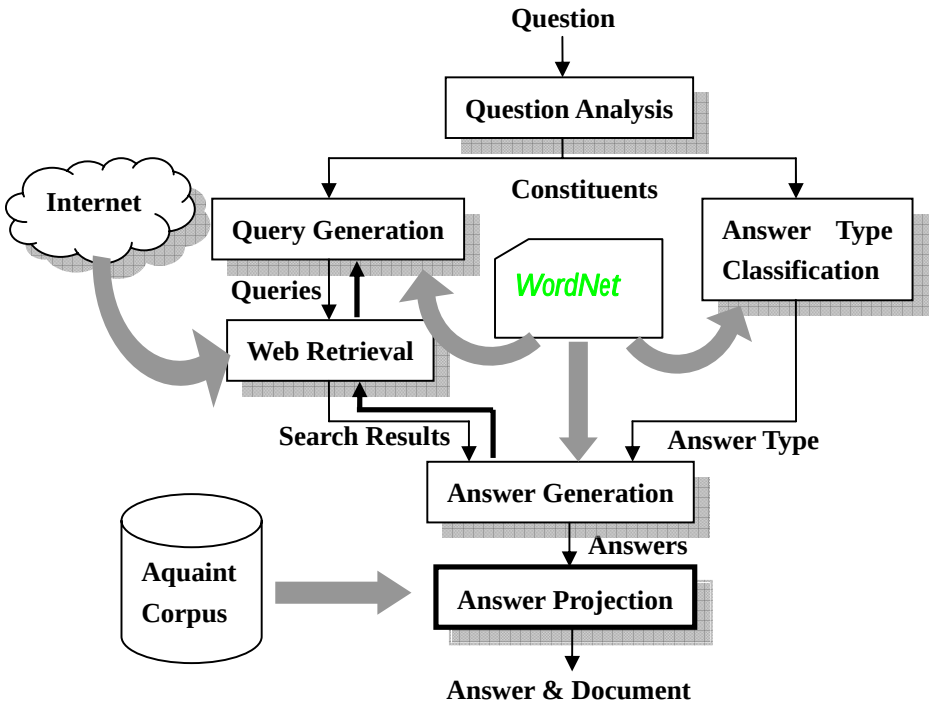

Figure 1 System Architecture for Web-base QA system

2.1.1Answer Validation-Feedback scheme

In this year, we attempt to use the answer validation-Feedback scheme to improve the performance. The main idea is that the answer generation procedure terminates until the top ranked answer are very confident.

The Query Generation module generates a list of queries from strict to loose according to the question and target, and provides them to the Web Retrieval module one by one.

The Web Retrieval module uses the query to search the Web and return snippets to the answer generation module five by five, but no more than 200 snippets.

The Answer Generation module extracts the answer from the snippets and ranking the answers. The procedure is stop until the top 1 answer is confident.

Currently, an answer is considered as confident if the occurrence number of the answer string is not less than five.

2.1.2Answer Projection

Answer projection is to find the support document(s) in certain corpus (AQUAINT for TREC task). It is a document retrieval problem and can also fit the demand of document ranking task of this year. Therefore, we do not specially develop a system for document ranking task.

For answer projection, the answer string is used as an additional key phrase, besides the queries of the question. In our system, the very strict query generation strategy is undertaken. There are four kinds of queries generated from question and target, as illustrated in following from strict to loose.

1. key phrases of question and target
2. key phrases of question and target except the verb phrases
3. key words (noun, adj, verb) of question and target
4. key words of question and target except the verb phrases

There are two kinds of allowed distribution of the key phrases/words, as illustrated in following from strict to loose.

1. within three sentences
2. within a document

There are two kinds of methods to use the answer string, as illustrated in following from strict to loose.

1. add the answer phrase to the queries, and search the documents
2. only use queries generated from question and target, and use answer phrase to filter the documents

The ranking strategy is: the distribution of the key phrases/words is the first key; the query strictness of query generation is the second key; and the methods of using the answer string is the third.

The procedure of answer projection uses the combination of the three strategies by the order according to the ranking strategy, and is terminated when any support document is found.

2.2 Answer Combination

The Web-based system is still underdeveloped, in order to capture the strongpoint of our currently developing and previous systems, six heuristic rules are applied to combine the results, as illustrated in following.

1. If the answer strings of system 1 and 2 are same, select the result of system2
2. else if the answer string of system 1 is NIL and system 2's is not NIL, and if the score of system 2 is larger than 0.8, select the result of system2
3. else if the answer string of system 1 is not NIL and system 2's is NIL, and if the score of system 1 is less than 0.5 or the Answer type belongs to the number class (except DAT), select the result of system2
4. if the answer string of system 1 is not NIL, the score of system 1 is less than that of system 2 and the answer type belongs to PRN , LCN or DAT, select the result of system2
5. if system 3 uses the strict rules to find the answer and the occurrence of answer string is not less than 5, and if the answer type is ABBR or the question is short and its interrogative word is when/where and the score of system 1 is less than 1.0, select the result of system3
6. if none of the above rules can be applied, select the result of system 1

These rules are validated by the experimental results on the data of TREC 2004. Rule 1 - 4 means when system 1 is not very confident about the answer of a question, system 2's result is prefer. Rule 5 means when system 3 is very confident about certain types of questions, system 3's result is prefer. Rule 6 means if none of rule 1 - 5 is work then use the result of system 1.

3. List Question

We use patterns to extract answers for list question. The flow chart of FDUQA on list

question is illustrated in figure 2. First, the question is approached by factoid question processing module (we use the Web-based factoid QA system as described in section 2), and top 10 candidate answers are used as seeds for list question processing module. Second, the seeds are put into the sentence corpus and new candidate answers are extracted by patterns. At last, candidate answers are filtered by answer filtering module and final answers are put out.

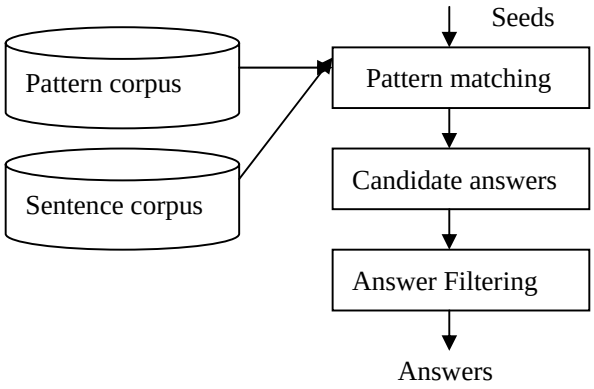

Figure 2 Flow Chart of FDUQA on List Question

3.1 Pattern matching

The answers of a list question always appear in a paratactic structure. We obtained these structures from the past TREC list questions, and built a pattern corpus consisting of these structures. Here are some examples of our patterns:

P1. *(?:including|include|included|includes|involve|involving|involves|involved) (NP like)? <A> , <A> and | or | as well as <A>*

P2. *such as <A> ((,<A>)* and | or <A>)?*

P3. *like <A>(,<A>)* (and | or | as well as <A>)?*

P4. *(<A>,)* <A> (and|or|as well as) <A>*

These patterns are expressed in regular expression. Tag <A> here means the answer. Top 10 answers found by factoid question processing module are used as seeds in pattern matching.

Last year, we use target and each seed as query to search in AQUAINT corpus, and the results are collected as sentence corpus. In this year, all the answer sentences supporting the seeds in factoid question processing module are collected to build a temporary sentence corpus.

One example for pattern matching is as follows:

For question "*What is OPEC countries*", we first treat it as a factoid question and answer "*Iraq*" was found. Then we use "*Iraq*" as a seed and put it into sentence corpus. There is a sentence which supports "*Iraq*" in factoid question processing module: "*OPEC members are*

Saudi Arabia , Iran , Kuwait , Qatar , United Arab Emirates , Algeria , Nigeria , Iraq , Libya , Ecuador , Gabon , Indonesia and Venezuela." Through pattern P4, "*Saudi Arabia*", "*Iran*" ,"*Kuwait*", "*Qatar*" , "*United Arab Emirates*" , "*Algeria*" , " *Nigeria*" , "*Libya*" , "*Ecuador*" , "*Gabon*" , "*Indonesia*" and "*Venezuela*" are extracted out as candidate answers. And all the candidate answers are put into answer validation modules.

3.2 Answer Filtering

We find right answers from candidate answers after answer filtering module. Here are some rules for answer filtering:

- Proper noun phrase filtering: The candidate answer which is not proper noun phrase will be filtered.
- Stop words filtering: A stop words list is used to filter the candidate answers.
- The candidate answer which is the same as other seeds will be thrown off.

At last, candidate answers are ranked by their frequencies and top ranked candidate answers will be put out as right answers.

The qualities of patterns and seed answers affect the performance of list question very much. Wrong seed answers will bring even worse results by patterns.

4. Definition Question

In order to automatically identify definition sentences from a large collection of documents, we utilize both the existing definitions in the Web knowledge bases and the automatically generated structured patterns. Effective methods are adopted to make full use of these resources, and they promise high quality response to definition questions.

4.1 System Overview

We adopt a general architecture for definition QA. The system consists of five modules: target classification, document processing, Web knowledge acquisition, structured pattern generation and definition extraction. The flow chart for definition question of FDUQA is in figure3.

First, a question target is input, and the target classification module identifies the target type based on a few heuristic rules. The question targets are classified into several types through this module, such as person, organization and other thing. This target type is used in the Web knowledge acquisition module [Zhang et al, 2005] to determine which kind of knowledge bases will be searched, and it is also used to determine which set of pattern will be adopted.

Second, the document processing module generates the candidate sentence set

according to the target term. This module has three steps, document retrieval, relevant sentence extraction and redundancy removal. After these steps, we get the candidate sentence set of this target, which can be expressed as $S_A\{A_1, A_2, \ldots A_k, \ldots, A_m\}$, where A_k (k=1..m) is a candidate answer in the set and m is the total number of the candidate answers.

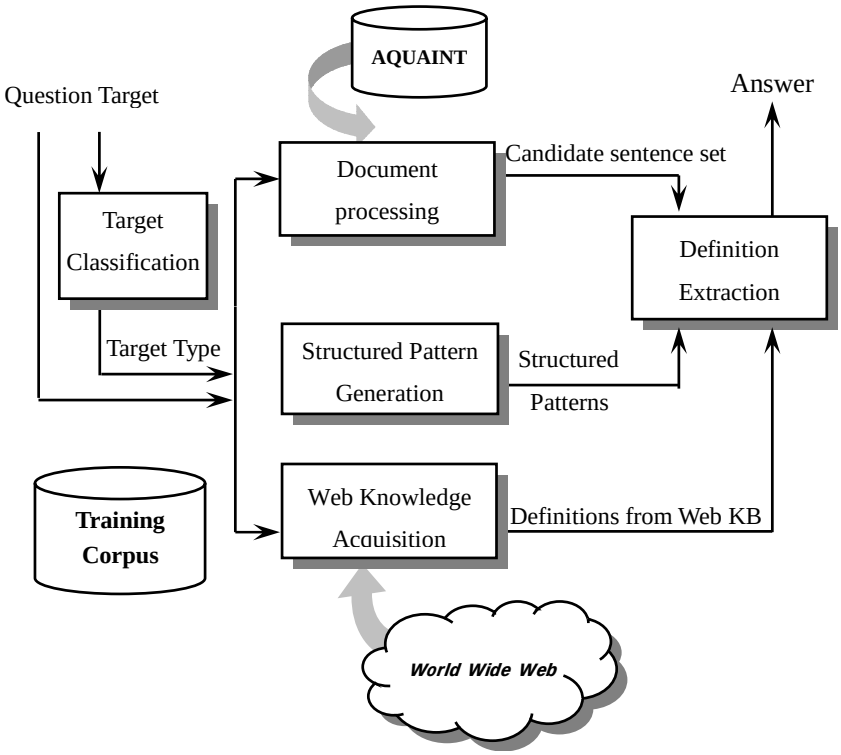

Figure 3 Flow Chart for Definition Question of FDUQA

Third, the Web knowledge acquisition module acquires the definitions of the target term from the Web knowledge base (KB). We search the definitions about the target from a number of online knowledge bases. These knowledge bases are the WordNet glosses and other online dictionaries such as the biography dictionary at www.encyclopedia.com. The definitions from them often supply knowledge that can be exploited directly and they are quite helpful to answering definition questions. We choose several authoritative KBs that cover different kinds of concept to achieve our goal. If we can find the definitions of question target from these sources, we use them to score the candidate sentences. (More detail can be found in [Wu et al, 2004])

In very few situations, no definitions can be found from the Web KBs, we form a centroid (i.e. a vector of words with their frequency) of the candidate sentence set to score the candidate sentences. The assumption is that words co-occurring frequently with the target in the corpus are more important ones for answering the question.

Fourth, we automatically generate several sets of structured patterns based on the training set, and then we score the candidate sentences using these patterns. (The generation of the structured patterns will be described in Section 4.2)

At last, the definition extraction module extracts the definition from the candidate sentence set based on the knowledge got from the Web knowledge base and the structured patterns. We will describe the detail of the definition extraction module in Section 4.3.

4.2 Structured Pattern Generation

Since definitional patterns can filter out those statistically highly-ranked sentences that are not definitional, and bring those definition sentences that are written in certain styles for definitions but are not statistically significant into the answer set [Cui et al 2004], we employ some automatically generated structured pattern to reinforce our Web knowledge based method.

We accumulate definition question-answer pairs from the all the submitted runs of the TREC12 and TREC13 QA tasks for use as training data. We generate a set of patterns for each kind of target (i.e. person, organization, and other thing) respectively.

For each kind of target, we firstly form a training set, which consists of the answer sentences to all the questions whose target term belong to this kind in the training corpus.

Secondly, in order to form general patterns, we substitute a general tag "<TARGET>" for those question targets in the answer sentences. We consider the context around the "<TARGET>", and the context is modeled as a window centered on "<TARGET>" according to the pre-defined size *w*. Thus we get fragments with size 2*w*+1 (*w* is set to 2 in our experiments) including the target term.

At last, we calculate the recurrence of these fragments in the training set, and the fragments with high occurrence are considered as the structured patterns. The frequency of the fragment in the training set is used as the weight of this pattern. A few top ranked patterns of the type "Person" are listed in Table 2.

Structured Pattern	**Weight**
<TARGET> , the	0.094
<TARGET> , a	0.042
<TARGET> , who	0.030
<TARGET> was a	0.012
known as <TARGET> , is	0.012

Table 2 Top ranked patterns of the type "Person"

4.3 Definition Extraction

For each candidate sentence A_i in the set S_A, we calculate its importance $Score_i$ as follows:

1. Calculate its similarity with the definition from Web KBs [Wu et al, 2004], which is expressed as $SimScore_i$;
2. Substitute a general tag "<TARGET>" for those question targets in A_i;

3. Apply hard matching between A_i and each structured patterns of this target type, and the pattern match score of A_i, $PatScore_i$ is set to the weight of the pattern matched.

4. $score_i = \alpha \bullet SimScore_i + \beta \bullet PatScore_i \quad (\alpha + \beta = 1)$. The weights α and β are fixed based on experiment, and they are set to 0.7 and 0.3 respectively in our experiments.

5. Rank the sentences of set S_A based on $Score_i$ (i=1..m), and the top ones are chosen as the definition of the target term.

The results reveal that the Web knowledge bases and the structured patterns are effective resources to definition question answering, and the method presented gives an appropriate framework for answering definition question.

5. Results

We submitted two runs for the main task and document ranking task of TREC14 QA Track: FDUQA14A and FDUQA14B. In the two runs, the algorithms used to answer factoid questions are different. FDUQA14A is system 1, while FDUQA14B is the combination system, as described in section 2. The results of list questions in the runs are just the same. As to definition questions, difference between the two runs is FDUQA14B combines the structured patterns while FDUQA14A not.

		FDUQA14A	FDUQA14B
Final Score		0.192	0.205
Factoid Question	#correct	86	94
	#unsupported	21	17
	#inexact	14	14
	#wrong	241	237
	Accuracy	0.238	0.260
List Question	Average F score	0.056	0.055
Definition Question	Average F score	0.231	0.232
Document Ranking	Average precision	0.1602	0.1435
	R-Precision	0.1813	0.1595

Table 3 Performance of FDUQA Runs in TREC 2005

From this table, we can see that system combination can achieve better performance than the single Web-based system. The algorithm we use to answer definition questions is quite promising. The list questions are answered not so well.

Although FDUQA14B has a better performance than FDUQA14A in main task, FDUQA14B is not as good as FDUQA14A in document ranking task. The reason is that we do

not try to find as many relevant documents as possible but only try to find the best supporting documents. This may affect the performance.

Acknowledgements

This research was supported by the National Natural Science Foundation of China under Grant No. 60435020. We are very thankful to Ningyu Chen, Feng Ji, Xiaofeng Yuan and Jian Huang for their contributions in our work.

Reference

Lide Wu, Xuanjing Huang, Lan You, Zhushuo Zhang, Xin Li, Yaqian Zhou. FDUQA on TREC2004 QA Track. In: Proceedings of the Thirteenth Text REtreival Conference, Gaithersburg, Maryland, 2004

Lide Wu, Xuanjing Huang, Yaqian Zhou, Yongping Du, Lan You.2003. FDUQA on TREC2003 QA Task. Proceedings of the TREC-12, Gaithersburg, Maryland, 2003

Lide Wu, Xuanjing Huang, Junyu Niu, Yingju Xia, Zhe Feng, Yaqian Zhou.2002. FDU at TREC2002: Filtering, QA, Web and Video Tasks. Proceedings of the TREC-11, Gaithersburg, Maryland, 2002.

Zhushuo Zhang, Yaqian Zhou, Xuanjing Huang, Lide Wu. Answering Definition Questions using web knowledge base. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing, 2005

Hang Cui, Min-Yen Kan, Tat-Seng Chua, Jing Xiao: A comparative Study o Sentence Retrieval for Definitional Question Answering. In Proceedings of the 27th Annual International ACM SIGIR Conference, 2004